

ELECTRIC SHEPHERD

Inventory Central

User Manual

Version 0.1.17a (alpha) · Built 2026-10-21

Inventory Central
v0.1.17a

1 Summary

2 Stock

3 Inventory

4 Connections

5 Decom

6 History

7 Config

8 Settings

ITEMS
20

CONNECTIONS
20

SUPPLY UNITS
193

LOW STOCK
4

Tue Oct 20 2026 2:07 PM

! 5 alerts

Site All HQ Annex

Type All types Access Point Firewall Laptop Monitor Patch Panel

Inventory > All sites > All types

17 items · sorted by Component Type

Type	Model	Serial	Asset Tag	Status	Room	Maker	Site	Loc.
Access Point	U6-Pro	F492BF5...	AP-101	Deployed	Floor 1...	Ubiquiti	HQ	
Access Point	U6-Pro	F492BF5...	AP-102	Deployed	Floor 2...	Ubiquiti	HQ	
Firewall	FortiGa...	FG100FT...	FW-01	Deployed	Server ...	Fortinet	HQ	A1
Laptop	ThinkPa...	LT-0045		Deployed	IT clos...	Lenovo	HQ	
Monitor	U2723QE	CN0K2PH6	MON-0012	Deployed		Dell		
Patch Panel	48-port...		PP-A1	Deployed	Server ...	Leviton	HQ	A1
PDU	AP8941	ZA22010...	PDU-A1	Deployed	Server ...	APC	HQ	A1
Router	EdgeRou...	F09FC2A...	RTR-01	Deployed	Server ...	Ubiquiti	HQ	A1
Server	PowerEd...	7XK2P43	SRV-001	Deployed	Server ...	Dell	HQ	A2
Server	PowerEd...	7XK2P47	SRV-002	Deployed	Server ...	Dell	HQ	A2
Server	ProLian...	CZ22310...	SRV-003	Deployed	Server ...	HPE	HQ	A2
Storage	RS3621x...	2130TWR...	NAS-01	Deployed	Server ...	Synology	HQ	A2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-01	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-02	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	Catalys...	FOC2145...	SW-CORE-01	Deployed	Server ...	Cisco	HQ	A1
Switch	Catalys...	FOC2145...	SW-CORE-02	Deployed	Server ...	Cisco	HQ	A1
UPS	SMX3000...	AS22102...	UPS-01	Deployed	Server ...	APC	HQ	A1

17 items in service - 11 types - 5 places - 3 on Stock
DEMO DATA: changes are not saved (run without --demo for your own data)
a add e edit E cell d del m status {} theme b sidebar w save ? help q quit

Electric Shepherd

Equipment, connections and supplies in your terminal · Linux and macOS

Contents

1. Welcome to Inventory Central	6
1.1 What is in version 0.1.17a	6
1.2 What you need	8
1.3 Installing and starting	8
1.4 The splash screen	8
1.5 About the version number	9
1.6 Where your data lives	9
1.7 Terms of use	10
1.8 Conventions in this manual	10
2. A Tour of the Screen	11
2.1 The sidebar layout	11
2.2 Top menu, page, type	12
2.3 The top-bar layout	14
2.4 Windows that are too small	15
2.5 Text-only mode	16
2.6 The top menus in one table	16
3. Keys, Forms and Menus	17
3.1 Moving around	17
3.2 The same keys on every list	19
3.3 Forms	20
3.4 Dates	23
3.5 Confirmations	24
3.6 Mouse	24
4. Quick Start: Ten Minutes with Your Own Data	25
4.1 Start it	25
4.2 Say where things are	25
4.3 Add two devices	25
4.4 Connect them	26
4.5 Stock the shelf	26
4.6 See what the program thinks	26
4.7 Sort, search and filter	27
4.8 Write a report	27
4.9 Quit	27
5. The Inventory	28
5.1 What is recorded	28
5.2 Adding, editing and cloning	29
5.3 Status: Deployed, Reserved, Stock, Decommissioned	30
5.4 Site pages and types	31
5.5 Working with many items	32
5.6 Viewing one item	34
5.7 Decom: taking items out of service	35
5.8 Renaming a device	38
5.9 Sorting, searching and filtering	38
6. Ports	39
6.1 What is recorded	39
6.2 Ports made with the item	40
6.3 Adding ports later	41
6.4 Ports from existing cables	41
6.5 What follows the item	41
7. Connections	42
7.1 What is recorded	42

7.2 Devices and ports	43
7.3 What the program refuses	44
7.4 Power	44
7.5 Site pages and types	45
7.6 Viewing, cloning, editing and deleting	45
7.7 Sorting, searching and filtering	46
8. Stock	47
8.1 Items on the shelf	48
8.2 Supply lines	48
8.3 Counting	49
8.4 Low stock	50
8.5 Optics and the Connections page	50
8.6 Cloning, viewing, deleting	51
8.7 Sorting, searching and filtering	51
9. Config	52
9.1 Sites, floors, rooms and locations	53
9.2 Models and the other lists	56
9.3 Importing places with items	56
9.4 The files	57
10. Sorting, Searching and Filtering	58
10.1 Sorting	58
10.2 Searching	59
10.3 Filtering on one column	60
10.4 What stays and what goes	61
10.5 Finding things: some recipes	62
11. Summary	63
11.1 What it shows	63
11.2 The totals in the margin	64
12. Checks	65
12.1 Errors, alerts and notes	65
12.2 Values the forms would refuse	67
12.3 Going to the record	68
13. History	69
13.1 The columns	69
13.2 Reading one change	70
13.3 One action, one change set	70
13.4 The file	70
13.5 When something is wrong with it	71
13.6 Backups and size	72
14. The Settings Page	73
14.1 Theme	74
14.2 Clock beside the date	79
14.3 Splash screen	79
14.4 Start on page	81
14.5 Sidebar menu	81
14.6 Mouse	81
14.7 Web page (server)	81
14.8 The settings file	82
15. Exporting Reports	83
15.1 The three questions	83
15.2 What is in a CSV file	85
15.3 What is in a Markdown file	85
15.4 Exporting is read-only	86
16. Importing CSV Files	87

16.1 Step one: choose the file	87
16.2 What counts as "already here"	87
16.3 What the program reads	88
16.4 Step two: read the preview	89
16.5 After the import	90
16.6 Moving data in from other sources	91
17. Your Data: Files, Saving and Backups	92
17.1 The data files	92
17.2 What is inside the files	93
17.3 When it is saved	94
17.4 One copy at a time: the lock	95
17.5 Who can read the files	96
17.6 Demo mode	96
17.7 The crash log	96
18. Sharing the Data: the Server	97
18.1 What changes when you use it	97
18.2 Starting a server by hand	97
18.3 Working with the server	98
18.4 When two people change the same row	98
18.5 When the server cannot be reached	99
18.6 When the server cannot record changes	99
18.7 Going back to files	99
18.8 Backing up the server	99
18.9 People on other computers: SSH	100
18.10 Who may do what: roles	100
18.11 Directory groups (LDAP, Active Directory) through sssd	101
18.12 Idle sessions	102
18.13 The server checks the data too	102
18.14 Setting it up as a service	102
18.15 Turning it back into files: teardown	103
18.16 Certificates for the web page	104
18.17 The server's options	104
18.18 What the server does not do yet	105
19. The Web Page	106
19.1 How people get in	106
19.2 Reading the page	107
19.3 Changing data	110
19.4 Moving equipment in a rack	112
19.5 The port map	113
19.6 Printing	113
19.7 Closing the page	114
19.8 Looks	115
19.9 Setting it up	116
19.10 What the page does not do yet	117
20. The Command Line	119
20.1 Options	119
20.2 Printing a page as text: `--dump`	120
20.3 Environment	121
20.4 Exit status	122
20.5 The helper copy	122
21. When Something Goes Wrong	123
21.1 Starting	123
21.2 The screen	123
21.3 Keys	124
21.4 Records	125
21.5 Saving and files	125
21.6 Other	126
21.7 Reporting a problem	126

22. Limits and Things to Know	127
22.1 Size	127
22.2 What it displays	127
22.3 Keys	128
22.4 Files and systems	128
22.5 What counts as a duplicate	129
22.6 What the Checks cannot know	129
Appendix A. Key Reference	130
A.1 On every page	130
A.2 Every list	130
A.3 Summary, Checks and Settings	131
A.4 In a form	132
A.5 In a menu or suggestion list	133
A.6 In other dialogs	134
Appendix B. The Fields	135
B.1 Choices and suggestions	140
Appendix C. CSV Columns and Alternative Headings	142
C.1 Inventory	142
C.2 Connections	143
C.3 Supplies	143
C.4 Ports	144
C.5 Sites	144
C.6 Rooms	144
C.7 Locations	145
C.8 Manufacturers	145
C.9 Models	145
C.10 Types	145
C.11 CPUs	146
C.12 Memory	146
C.13 Port types	146
C.14 Floors	146
Appendix D. If You Used an Earlier Version	147
D.1 Keys that changed	147
D.2 What else is different	148
Appendix E. Release Notes	149
E.1 Version 0.1.17a, October 2026	149
E.2 Version 0.1.16a, October 23, 2026	149
E.3 Version 0.1.15a	150
E.4 Version 0.1.14a	150
E.5 Version 0.1.13a	150
E.6 Version 0.1.12a	151
E.7 Version 0.1.11a	151
E.8 Version 0.1.10a	151
E.9 Version 0.1.9a	152
E.10 Version 0.1.8a	152
E.11 Version 0.1.7a	152
E.12 Version 0.1.6a	152
E.13 Version 0.1.5a	153
E.14 Version 0.1.4a	153
E.15 Version 0.1.3a	153
E.16 Version 0.1.2a	153
E.17 Version 0.1.1a	153
E.18 Version 0.1a	154

1. Welcome to Inventory Central

Inventory Central is a tracker for the equipment, the network and power connections and the supplies of a small or medium IT environment, and it lives in your terminal. You record what you own (servers, switches, PDUs, access points, laptops) and where it is (which site, room and rack, at which rack unit), which ports each device has, how it is cabled (which port goes to which port, on what cable, carrying what), and what is on the shelf (spare equipment, patch cables, optics, ties and labels, and how many of each). It then keeps the records tidy for you: it refuses duplicates, offers values you have used before, keeps two devices out of the same rack units and two cables off the same port, warns when a supply is running low, and writes any view of your data to a CSV or Markdown report.

It is one small program. It has no dependencies, needs no network connection, and keeps everything in plain CSV files that any spreadsheet can open and that you can back up, search and edit by hand.

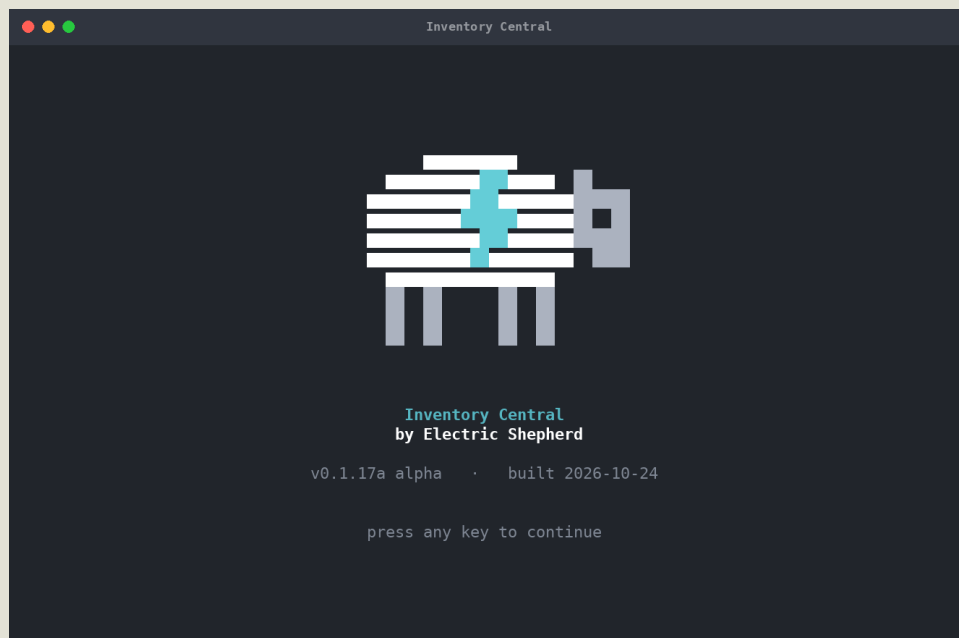


Figure 1. The start-up splash: the Electric Shepherd sheep, the name of the program, the version and the build date. Any key continues.

1.1 What is in version 0.1.17a

Version 0.1.17a is an alpha of the Rust edition of Inventory Central. It has eight top menus, each with the number key that opens it:

- **1 Summary:** counts and totals across the tables (Chapter 11), and on its second page the **Checks:** duplicated tags, double-booked ports and rack units, cables to devices or ports that are not on record, supplies at or below their minimum and gaps in the records, found for you (Chapter 12).
- **2 Stock:** what is on the shelf and not in service: equipment in stock or reserved, and the supply lines (supplies, parts and cables) with counts and minimum levels (Chapter 8).

- **3 Inventory:** the equipment in service, with asset tag, serial number, type, status, manufacturer, model, site, room, location (a rack, a shelf, a desk, a bin or a closet) and rack position, CPU and memory, hostname, project, purchase date, purchase order and notes (Chapter 5).
- **4 Connections:** the cabling between ports, with both ends, the connection type and speed, VLAN, cable grade, color and length, the optic in the port, and a status. Power cords run from a PDU or UPS outlet to a power input. Its last type is the **Ports** of every item (Chapters 7 and 6).
- **5 Decom:** equipment and connections taken out of service for good, kept on record until you delete them, and restored as they were with one key (section 5.7).
- **6 History:** every change ever made, by whom and when, from the change history kept beside your data (Chapter 13).
- **7 Config:** the lists that fill the menus, on two pages: **Sites** (sites, floors, rooms and locations) and **Types** (types, manufacturers, models, CPUs, memory and port types) (Chapter 9).
- **8 Settings:** how the program looks and where it starts (Chapter 14).

Under a top menu, **v** steps through its **pages**, one for each site, and **j** through its **types**; the page heading says where you are, for example `Inventory > HQ > Switch` (Chapter 2).

On top of that:

- **Sorting, searching and filtering** that work the same on every list (Chapter 10).
- **Working with many rows at once:** mark rows with **Shift-Down** (or **J**) and delete, move or change them together; add items by scanning serial numbers or asset tags with a barcode scanner; clone an item in one key (section 5.5).
- **Menus and suggestions** in every form. Types, statuses, speeds and colors come from lists; sites, rooms, locations, models and ports come from your Config lists and your records, and each list narrows to what fits: the rooms of the chosen site, the ports of the chosen device.
- **CSV import and export**, and **Markdown export**, of the whole table or only the rows you are looking at (Chapters 15 and 16).
- **Seventeen color themes**, among them the Electric Shepherd brand colors, muted dark themes from cool Winter to warm Samhain, and one light theme, Paper (section 14.1), stepped through with **{** and **}**, and a sidebar you can switch off.
- **The mouse:** click a page, a view, a row, a column heading, a key hint, a form field or a menu entry; the wheel scrolls (section 3.6).
- **Careful saving:** every change is saved at once, atomically, with a backup, and a lock stops two copies of the program from writing the same files (Chapter 17). Every change is also recorded in the change history (Chapter 13).
- **A server for sharing**, `inventoryd`, which you may start if several people should work on the same data at once, on the server's computer or from their own through SSH, as viewers, editors or admins (Chapter 18). Until you start it, or an administrator runs `inventoryd setup`, nothing of it runs; `inventoryd teardown` turns it back into files. The server can also show the data on a **web page**, behind nginx, to people with a certificate from your own authority (Chapter 19).

Note. Inventory Central keeps records; it does not discover your network and it does not talk to it. Everything in it is what you typed or imported. This is an alpha release: keep your own backup of the data files (section 1.6) until you are comfortable with it.

1.2 What you need

- **Linux or macOS** and any terminal that understands ANSI escape codes (practically every modern terminal does). Native Windows is not supported; use WSL.
- A terminal at least **50 columns by 12 rows**. An ordinary **80 by 24** window is fully supported. At **100 columns by 24 rows** or more you get the sidebar layout, and about **120 columns by 34 rows** shows most of the columns of every table without trimming.
- A **Rust toolchain** (1.97 or newer) to build it from source. Inventory Central uses only the Rust standard library, so building works offline.
- The standard POSIX commands `sh`, `stty` and `date`, which every Linux and macOS system has. The program uses `stty` to control the terminal and `date` to read your local time (it falls back to UTC without it).

1.3 Installing and starting

Inventory Central ships as a source archive, and as a ready-built program for Linux on x86-64. To build it from the source, unpack it, build it once, and run it:

```
tar -xzf inventory-central-0.1.17a.tar.gz
cd inventory-central-0.1.17a
cargo build --release           # no network access needed (no dependencies)
./target/release/inventory --demo # look around with sample data (nothing is saved)
./target/release/inventory      # your own data
```

The program is called `inventory`. To run it from anywhere, copy it to a folder on your `PATH` (for example `~/local/bin`). The first time you run it without `--demo` it creates its data folder and starts empty, ready for your own records (Chapter 4 walks through that).

Tip. Run `inventory --demo` first. It loads a small office: two sites, three floors, seven rooms and seven locations (four racks, two shelves and a bin); twenty items, three of them on the shelf; 244 ports; twenty connections, five of them power cords; and eighteen supply lines, with a few things for the Checks page to point out. Changes made in demo mode are never saved, and a reminder is shown on the message line.

1.4 The splash screen

Every time Inventory Central starts it shows a splash screen: the Electric Shepherd sheep above the name of the program, the company, the version and the build date. It waits for a key, and that key is used up by the splash: pressing `q` on the splash does not quit, it only continues. If the window is too short for the sheep (it needs 20 rows), a framed title box is shown instead, and in a terminal that is not set to UTF-8 the title box is always used.

To start without the splash, use the `-n` (or `--no-splash`) option, set `INVENTORY_NO_SPLASH=1` in your environment, or choose **Off** under Settings (section 14.3). The splash only appears when the interactive program starts; `--dump`, `--help`, `--version` and `--licence` never show it. `inventory --version` prints one line:

```
$ inventory --version
Inventory Central v0.1.17a (alpha, built 2026-10-23)
```

1.5 About the version number

The short version reads `v0.1.17a`: major version 0, minor version 1, build 17, and the letter **a** for alpha (**b** will mean beta). Every build raises the number. The version is set in one place and everything you see is derived from it. The build date is stamped in when the program is compiled, in UTC.

1.6 Where your data lives

Your records are kept in plain CSV files, one per table and one per Config list, saved after **every** change; there is no save command to remember. By default they are in:

<code>~/.local/share/inventory-central/inventory.csv</code>	the inventory
<code>~/.local/share/inventory-central/inventory-connections.csv</code>	the connections
<code>~/.local/share/inventory-central/inventory-supplies.csv</code>	the supplies
<code>~/.local/share/inventory-central/inventory-ports.csv</code>	the ports
<code>~/.local/share/inventory-central/inventory-sites.csv</code>	the Config lists, one file each
<code>~/.local/share/inventory-central/inventory.journal</code>	the change history

If `XDG_DATA_HOME` is set, the folder is `$XDG_DATA_HOME/inventory-central/` instead. You can use a different place with `-f PATH` (`--file`), or with the `INVENTORY_FILE` environment variable; the other files are always kept beside the one you name and named after it. Your settings are kept apart, in `~/.config/inventory-central/settings`. Chapter 17 has the details, including the backups and the lock.

- **Crash-safe saves.** Each file is written to a temporary file, flushed to disk, and then renamed over the old one, so a crash or power cut can never leave half a file.
- **An automatic backup.** The first save of each session keeps the previous file as `inventory.csv.bak` beside it (and likewise for the others).
- **A damaged file is never overwritten.** If a file cannot be read, the program refuses to start with a message and leaves it untouched.

Tip. To back up your records, copy the `.csv` files in the folder. To move to another computer, copy them there and point `-f` at the first one, or put them in the default folder.

1.7 Terms of use

Inventory Central is released by Electric Shepherd under the terms in the file `LICENCE`: free for personal or internal business use; no decompiling, reverse engineering, selling, renting, leasing or sublicensing; provided as is. The same text is built into the program: `inventory --licence` (or `-L`, or `--license`) prints it.

1.8 Conventions in this manual

- Keys to press are shown like `a`, `Enter`, `Esc` or `Ctrl-S`. Capital letters mean Shift, so `O` is different from `o`.
- `Text in this typeface` is something you type, a file name, or a command.
- Screenshots are real screens of the program. Most are the sidebar layout in a window of 120 columns by 34 rows; yours may show a slightly narrower or wider layout (Chapter 2). The sample data in the screenshots is the one that `--demo` loads, and its date is Tuesday, October 20, 2026, at 2:07 PM.
- Tip, Note and Careful boxes flag shortcuts, background, and things that can lose or change data.

2. A Tour of the Screen

Every screen of Inventory Central is built the same way, and it adapts to the size of your terminal. This chapter shows what is where. The pages themselves are covered one by one from Chapter 5 on.

2.1 The sidebar layout

In a window of **100 columns by 24 rows or more** you get the sidebar layout (Figure 2).

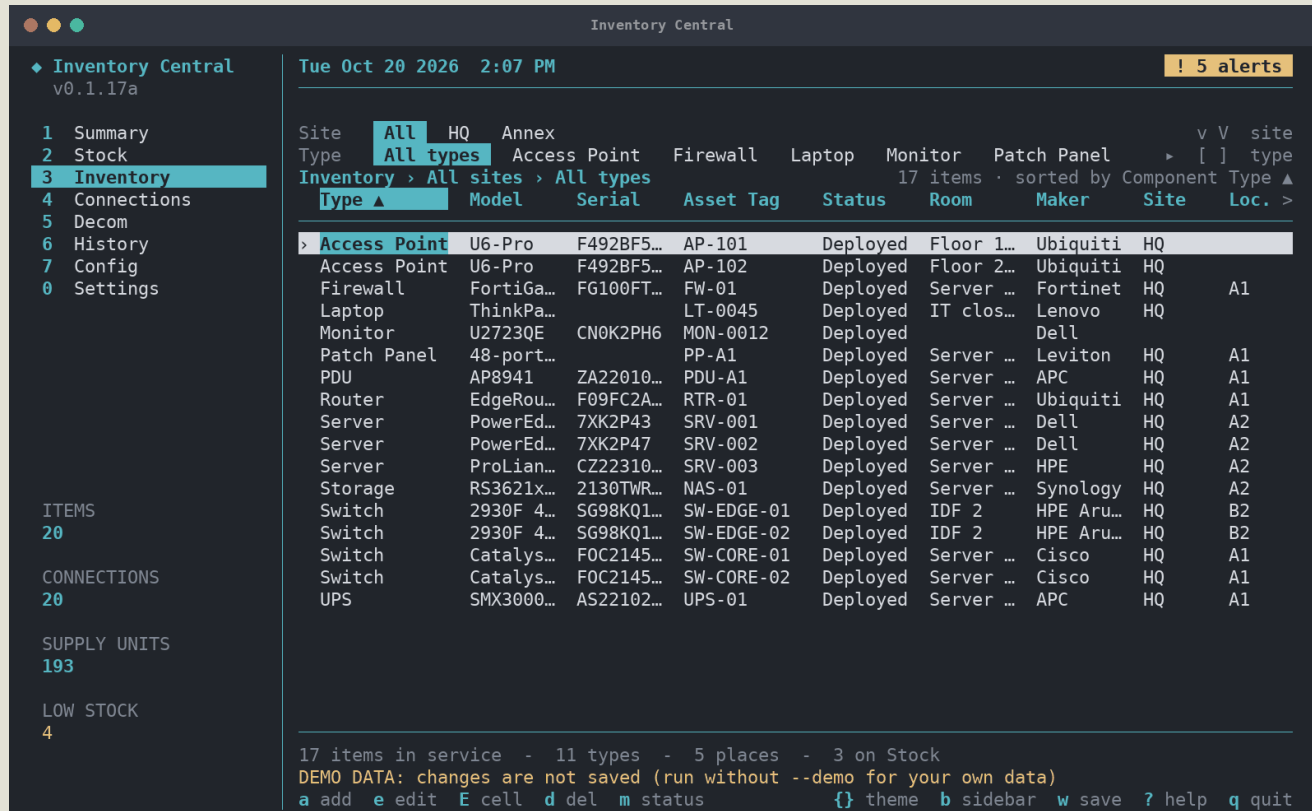


Figure 2. The Inventory in the sidebar layout, showing the demo data. The sidebar is on the left; on the right, the site bar, the type bar, the page heading and the table.

- **The sidebar** (left, 24 columns wide). At the top, the name of the program and its version. Below that, the eight **top menus**, each with the number key that opens it: **1** Summary, **2** Stock, **3** Inventory, **4** Connections, **5** Decom, **6** History, **7** Config and **0** Settings. The menu you are in is highlighted. At the bottom, four totals that are always in view: **ITEMS** (inventory records), **CONNECTIONS**, **SUPPLY UNITS** (how many individual cables, optics and so on are on hand, all lines added together) and **LOW STOCK** (supply lines at or below their minimum, shown in yellow when it is not zero).
- **The header** (top right). The date and the time, in 12-hour or 24-hour form (section 14.2), and on the right the **alert badge**, for example **! 5 alerts**. It counts the errors and alerts found by the Checks page (Chapter 12) and is yellow, or red when there is at least one error. When there is nothing to report the badge is not shown.
- **The page** (right). Two bars, the page heading, and a table or a report, described below.

- **The bottom lines.** Just above the key hints, one line says something useful about the page (a count, or what you are searching for) and gives way to a **message** when the program has something to tell you: a save that failed, an export that was written, a stock level that changed. The message goes away at your next key. The last line lists the keys for this page; **b**, **?** and **q** are always at the right.

The sidebar can be switched off: **b** on any page turns it off and on again, and the Settings page has the same switch (section 14.5). With it off, a wide window uses the top-bar layout described below, and the page gets the whole width (Figure 3).

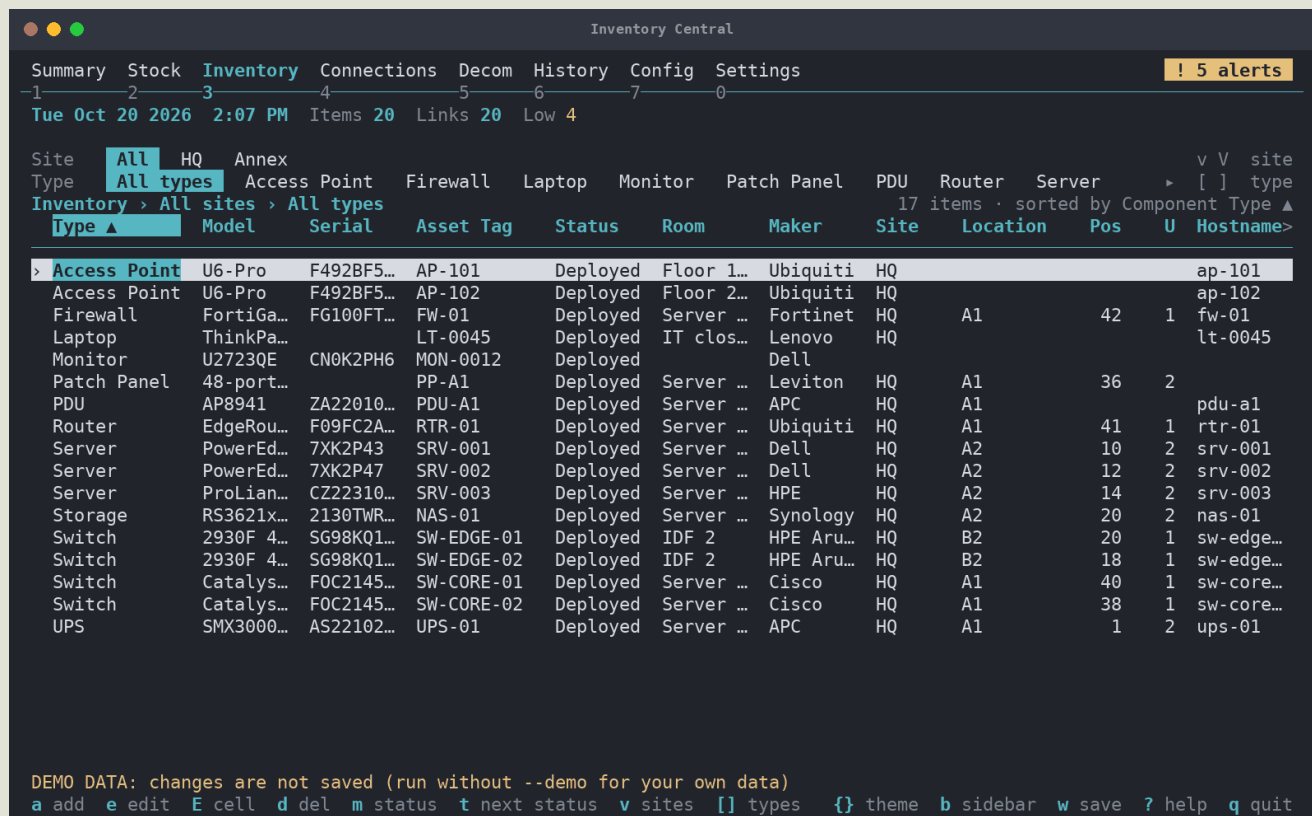


Figure 3. A 120-column window with the sidebar switched off: the top menus are named along the top, in full.

2.2 Top menu, page, type

Every list is reached in three steps: a **top menu**, then a **page** of it, then a **type**. The top menu is the number key (or **Tab** and **Shift-Tab**, **Right** and **Left**, or a click). Under it, two bars at the top of the page show the rest (Figure 2):

- **The site bar** (Site All HQ Annex): the pages of the menu, **All**, then one for each site on your Sites list. **v** moves to the next site and **V** (Shift-v) back; both wrap around. A site page orders its rows by place, room by room and rack by rack from the top unit down, with a heading over each group (section 7.5). The site you choose stays chosen on the menu's other types.
- **The type bar** (Type All types Access Point Firewall ...): **All types**, then each type of the rows on the page, and then the menu's other lists. **]** moves to the next type and **[** back. On Stock the other lists are **Supplies**, **Parts** and **Cables**; on Connections the **Ports**; on Decom the decommissioned **Connections**. A new row added on a type's page starts as that type.

Top menu	Pages (v)	Types (j)
Decom	All, each site	All types, each item type, Connections
Config	Sites, Types	the lists of the page

Reading a table

On a list page the page heading ends with the number of rows, for example `17 items`, or `2 of 17 items` while a search or filter is on, and the sort. Under it are the column headings, and then the rows.

- The **highlighted row** is the one the keys act on. The `>` at its left edge marks it too, so it is still visible in the mono themes.
- **Marked rows** (section 5.5) show in the accent color, with a mark at the left edge; the line under the table counts them.
- The **current column** has its heading highlighted. `o` sorts by it, `f` filters on it, and `h` and `l` move the highlight to the column on the left or right (Chapter 10).
- The column you sort by shows ▲ (ascending) or ▼ (descending) in its heading.
- A ... at the end of a value (~ in a terminal without Unicode) means it was too long for its column and has been trimmed on screen. The **current column** (the one with the highlighted heading) is always shown in full, and the other columns make room for it, so moving onto a column with `h` and `l` shows its values whole. The record keeps the full text: `Space` shows every field in full (section 5.6).
- A heading that does not fit uses a short form before it is cut: **Maker** or **Mfr.** for Manufacturer, **Serial** or **S/N** for Serial Number, **Purchased** for Purchase Date, **Host** for Hostname, **Loc.** for Location.
- A `>` at the right end of the headings means there are more columns to the right, and `<` at the left that some are scrolled away. Move the current column with `l` and the table scrolls with it.
- Colors carry meaning, and always together with the words. In the Full color look a connection's **Status** is green for Active, blue for Planned, yellow for Disabled and dim for Spare and Decommissioned; an item's **Status** is blue for Stock and Reserved; a supply **Qty** is yellow at or below its minimum and red at zero (or when it is not a number at all).

2.3 The top-bar layout

Smaller windows, from **50 columns by 12 rows** up to just under the sidebar size, use the top-bar layout, so an ordinary 80 by 24 terminal still gets a full-width page (Figure 5).

The top menus are named across the top (the current one highlighted) with the alert badge at the right. Eight names do not fit in 80 columns, so short ones are used there: `Sum Stk Inv Conn Dcm Hist Cfg Set`. The date and time and the totals (`Items 20 Links 20 Low 4`) are on the next line. The program's name and version move to the right end of the line just above the key hints. Everything else works the same, and the table simply shows fewer columns at a time; the current column is always kept in view.

The key hints at the bottom are trimmed to fit: the keys that do not fit are left out of the hint line, not out of the program. `?` always lists all of them.

Type	Model	Serial	Asset Tag	Status	Room	Maker
Access Point	U6-Pro	F492BF5...	AP-101	Deployed	Floor 1...	Ubiquiti
Access Point	U6-Pro	F492BF5...	AP-102	Deployed	Floor 2...	Ubiquiti
Firewall	FortiGa...	FG100FT...	FW-01	Deployed	Server ...	Fortinet
Laptop	ThinkPa...		LT-0045	Deployed	IT clos...	Lenovo
Monitor	U2723QE	CN0K2PH6	MON-0012	Deployed		Dell
Patch Panel	48-port...		PP-A1	Deployed	Server ...	Leviton
PDU	AP8941	ZA22010...	PDU-A1	Deployed	Server ...	APC
Router	EdgeRou...	F09FC2A...	RTR-01	Deployed	Server ...	Ubiquiti
Server	PowerEd...	7XK2P43	SRV-001	Deployed	Server ...	Dell
Server	PowerEd...	7XK2P47	SRV-002	Deployed	Server ...	Dell
Server	ProLian...	CZ22310...	SRV-003	Deployed	Server ...	HPE
Storage	RS3621X...	2130TWR...	NAS-01	Deployed	Server ...	Synology
Switch	2930F 4...	SG98KQ1...	SW-EDGE-01	Deployed	IDF 2	HPE Aru...
Switch	2930F 4...	SG98KQ1...	SW-EDGE-02	Deployed	IDF 2	HPE Aru...

Figure 5. The same page in an 80 by 24 window. The top menus are listed across the top, with their numbers on the rule below; the totals sit on the line under that.

2.4 Windows that are too small

Below 50 columns or 12 rows the program cannot draw a usable page and says so instead (Figure 6). Make the window larger and it carries on; nothing is lost. While the window is too small, keys wait; only **q**, **Ctrl-C** and **Ctrl-L** act (since 0.1.16a), so nothing can change behind a screen that shows nothing.

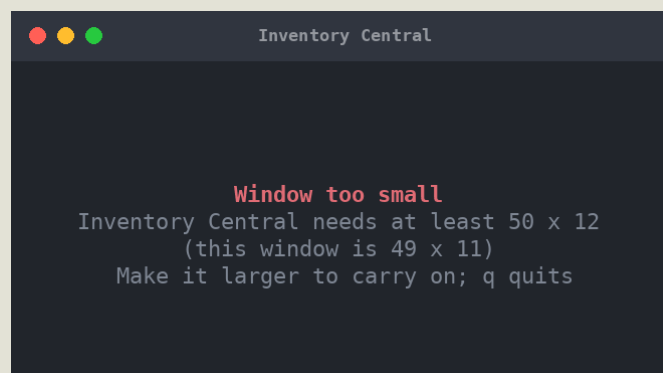


Figure 6. Too small a window. The message says what the program needs and what it has.

The program looks at the window size about once a second, so resizing works while it is running, and the clock moves on every minute without a key press. If your terminal does not report its size, set **COLUMNS** and **LINES** before starting the program (Chapter 20).

2.5 Text-only mode

Inventory Central draws its frames, arrows and the sheep with Unicode characters. If your locale is not UTF-8 (LC_ALL, LC_CTYPE and LANG are looked at, in that order), or if INVENTORY_ASCII is set, it uses plain ASCII instead: |, -, ^ and v for the frames and arrows, and the framed title box for the splash (Figure 7). To go the other way and force Unicode in a terminal that is configured for something else, set INVENTORY_UNICODE.

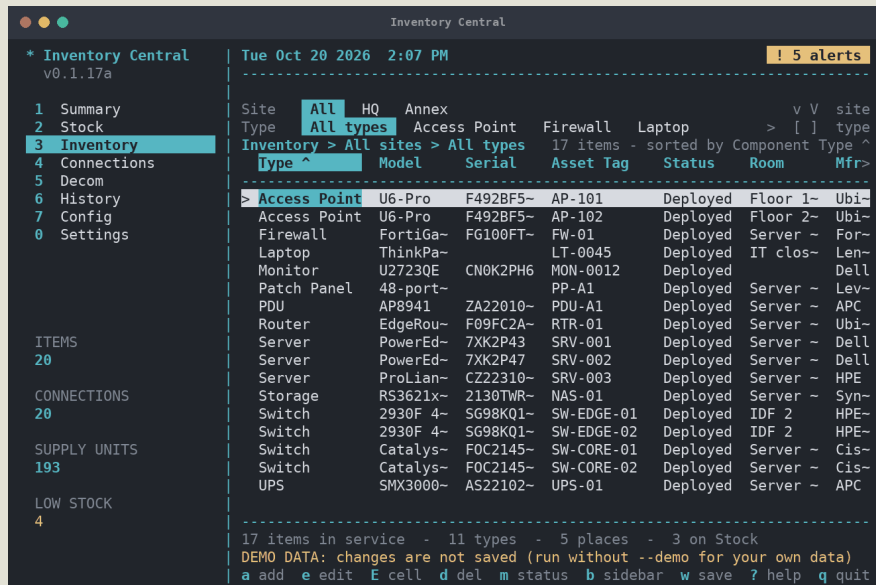


Figure 7. The same page with only ASCII characters, as shown when the locale is not UTF-8.

2.6 The top menus in one table

Top menu	Key	What it is for
Summary	1	Counts and totals across the tables; its second page is Checks, the problems and gaps found in the records (Chapters 11 and 12).
Stock	2	What is on the shelf: equipment in stock or reserved, supplies, parts and cables, with counts (Chapter 8).
Inventory	3	The equipment in service, by site and by type (Chapter 5).
Connections	4	The cabling between ports, network and power, and the ports (Chapters 7 and 6).
Decom	5	Equipment and connections out of service for good, until restored or deleted (section 5.7).
History	6	Every change, by whom and when (Chapter 13).
Config	7	The lists behind the menus: sites, floors, rooms, locations, types, models and more (Chapter 9).
Settings	0	Theme, clock, splash, start page, sidebar and mouse (Chapter 14).

You can also move between the top menus with **Tab** and **Shift-Tab**, or **Right** and **Left**; they wrap around at the ends.

3. Keys, Forms and Menus

Inventory Central is driven entirely from the keyboard, and the same handful of habits work everywhere. The keys follow vi where that makes sense: **j** and **k** move down and up, **h** and **l** move between columns, **g** and **G** go to the first and last row. Learn them once here and the rest of the manual is only about *what* each page does. Press **?** (or **F1**) at any time to see the key reference in the program itself (Figure 8; the full list is also in Appendix A).

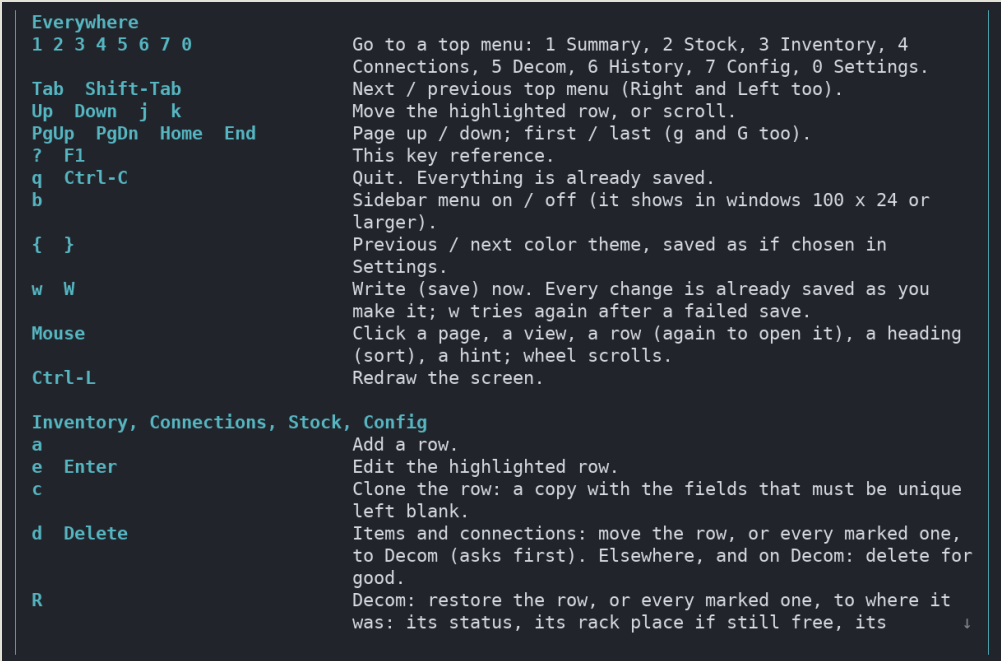


Figure 8. The built-in key reference, opened with **?**. Scroll it with **Up** and **Down**; **Esc** closes it.

3.1 Moving around

Keys	What they do
1 to 7 , 0	Open a top menu: 1 Summary, 2 Stock, 3 Inventory, 4 Connections, 5 Decom, 6 History, 7 Config, 0 Settings.
Tab / Shift-Tab , Right / Left	Next / previous top menu. They wrap around at the ends. (In a form or a dialog, Tab moves between fields instead.)
v / V	Next / previous page of the menu: All, then each site (the site bar). On Summary: Summary and Checks; on Config: its Sites and Types pages (section 2.2).
] / [	Next / previous type (the type bar): All types, each type on the page, then the menu's other lists: Supplies, Parts and Cables on Stock, Ports on Connections, Connections on Decom. On Config: the lists of the page.
Up / Down or k / j	Move the highlighted row in a table, or scroll a report.
PgUp / PgDn	Move or scroll a page at a time.

Keys	What they do
Home / End or g / G	Jump to the first or last row.
? or F1	Open the key reference.
b	Switch the sidebar off or on (section 14.5).
{ / }	Previous / next color theme, from cool to warm, saved as if chosen on the Settings page (section 14.1).
w / W	Save now. Every change is already saved as you make it; this is for peace of mind, and to retry after a failed save.
Ctrl-L	Redraw the screen, for example after another program wrote over it.
q or Ctrl-C	Quit. Everything has already been saved.

Every key that steps forward has one that steps back: **V** for **v**, **[** for **]**, **Shift-Tab** for **Tab**, **Up** for **Down**, and **r** reverses a sort.

3.2 The same keys on every list

Inventory, Stock, Connections (with its Ports), Decom and the lists of Config are all tables, and they all answer to the same keys:

Key	Action
a	Add a new row. A form opens.
e or Enter	Edit the highlighted row.
c	Clone the highlighted row: a form opens with the same values, and the fields that must be unique (asset tag and serial number, cable ID and ports, quantity, a name) left empty.
d or Delete	Delete. An item or a connection is moved to Decom (it can be restored there); on Decom, and on every other list, the row is deleted for good. You are always asked first. With rows marked, it acts on every marked row.
R	Decom only. Restore the row, or every marked one, as it was (section 5.7).
E	Change one cell: the highlighted cell's field opens, already being changed; Enter saves it, Esc leaves it.
Shift-Down / Shift-Up , J / K	Mark the row and the next or previous one, for a group action. Insert or a click with Shift marks one row; Ctrl-A marks every row shown; Esc unmarks them all (section 5.5).
M	Change fields of every marked row at once (section 5.5).
Space	View every field of the row in full, with the connections that use a device (section 5.6).
m	Items only. The status form: Deployed, Reserved, Stock or Decommissioned, with a note (section 5.3).
t	Items only. The next status at once: Deployed, Reserved, Stock, and round.
C	Items only. A quick clone, with no form: Serial Number -, no Asset Tag, Hostname or rack place.
B	Items only. Add items by scanning serial numbers or asset tags (section 5.5).
p	Items only. Make the item's ports (Chapter 6).
P	Ports and Connections. Make the ports that cables name but that are not on record (section 6.4).
+ / -	Supply lines only. One more, or one fewer, on hand. Saved at once (section 8.3).
o	Sort by the current column . Press it again to reverse. (s does the same.)
O	Sort dialog: choose a column and a direction (Chapter 10).
r	Flip the sort direction.
Alt-1 to Alt-9	Sort by the first to the ninth column from the left. Press again to reverse.
h / l	Move the current column (the highlighted heading) one column to the left or right. The columns themselves stay where they are.
/	Search every column for some text. Esc clears it.
f	Filter on the current column only. Esc clears it.
x	Export the rows to a CSV or Markdown file (Chapter 15).
i	Import a CSV file into this page (Chapter 16).

Actions always apply to the row you can see highlighted, however the table is sorted or filtered. When you add a row it becomes the highlighted one.

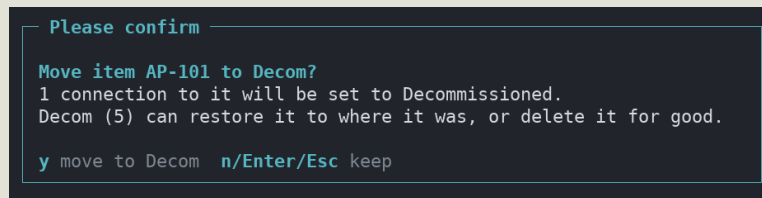


Figure 9. **d** asks first. An item in service moves to Decom: the box says which rack it leaves and how many of its connections are decommissioned with it. Only **y** goes ahead; **n**, **Esc** or **Enter** keeps the record.

The Summary, Checks and Settings pages have their own, shorter list of keys: **Enter** (or **e**) goes to the row a check is about, or changes the highlighted setting, and **Up** and **Down** move.

3.3 Forms

Adding or editing anything opens a form: a small dialog with one line per field, and under the fields two buttons, [**Save**] and [**Cancel**]. The field marked with > at its left edge is the one you are editing, and a short explanation of it appears near the bottom of the dialog. The last line of every form lists its keys.

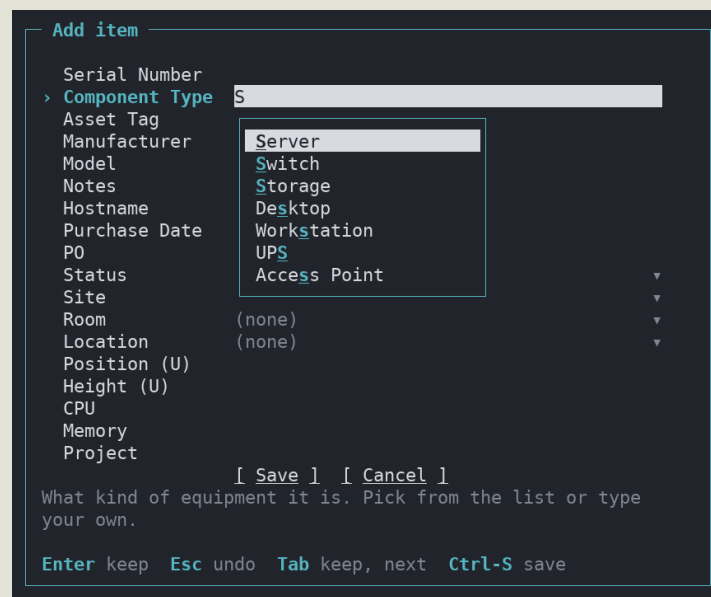


Figure 10. A form (adding an inventory item) with the suggestions of a field open. The list narrows as you type.

Adding or editing a record (an item, a connection, a supply line, a row of a Config list, the ports of a device) works in two steps since 0.1.13a, as in vi: you **move** between the fields, and you **change** one field at a time.

- **Moving.** **j** and **k** (or **Down** and **Up**, **Tab** and **Shift-Tab**) go to the next and previous field; **g** and **G** to the first and last. The field you are on is highlighted. Typing a letter here changes nothing: the form says Press Enter to change this field (j and k move between fields).
- **Changing.** **Enter** (or **e** or **Space**) starts changing the field. Its whole value is selected, so typing replaces it; the arrow keys, **Home** and **End** let you change part of it instead. A menu opens

its list, and a field with suggestions shows them. **Enter** again keeps the change and goes back to moving; **Esc** puts the field back as it was. **Tab**, **Down** and **Up** keep the change and move on in one step.

- **Saving.** **Ctrl-S** (or **F2**) saves from anywhere, keeping a change in progress. **Esc** while moving cancels the form.

The footer of the form says which of the two you are doing: j k move · Enter change · Esc cancel while moving, Enter keep · Esc undo · Tab keep, next while changing a field.

The other forms (sorting, searching, filtering, exporting, importing) are short and keep the simpler keys below: you type straight into the field, and **Enter** moves to the next field and runs the form from the last one.

Key	What it does in a form
Enter	In a record form: change the field; again, keep the change. In other forms, on a text field: move to the next field, and run the form from the last one. On a menu field: open the menu; a second Enter takes the highlighted choice. On a button: press it.
Tab / Down	Next field. After the last field come the Save and Cancel buttons, then the first field again.
Shift-Tab / Up	Previous field. From the first field, the Cancel button.
Left / Right	On a menu field: cycle through the choices without opening the menu. On a text field: move the cursor. On the buttons: Save or Cancel.
Ctrl-S or F2	Save from anywhere in the form. If something is wrong, the form stays open and tells you what.
Esc	Close an open menu first. In a record form, while changing a field: put it back as it was. Otherwise cancel the form without saving.
Home / End (or Ctrl-A / Ctrl-E)	Go to the start or the end of the text.
Delete / Backspace	Delete the character after or before the cursor.
Ctrl-U / Ctrl-K	Delete everything before or after the cursor.
Ctrl-N / Ctrl-P	On a field with suggestions: show the list, or step to the next and previous suggestion.

A field that already has a value is shown **selected** when you start changing it: just start typing to replace it, or use the arrow keys, **Home**, **End** and **Delete** to change part of it.

Pasting (since 0.1.16a). Text pasted from the clipboard goes into the field being changed, with line breaks and tabs made spaces. Anywhere else (on a list, on a menu, in a record form while moving between fields) a paste is ignored, and the message line says so (Pasted text is ignored: press Enter to change a field, then paste). Pasted text never runs as keys: a pasted dy on a list deletes nothing.

When a form cannot be saved, it stays open, puts the cursor in the field that needs attention, and says what is wrong in plain words: "Enter an Asset Tag or a Serial Number (or both)", "That Cable ID is already in use by another connection", "Quantity must be a whole number, like 12 (use 0 if you have none)". Fix it and save again. Nothing is changed until a form saves.

Figure 11. A form that could not be saved. The message names the problem and the cursor is in the field to fix.

Menus and suggestions

Fields with a ▼ at the right are menus: you must pick one of the choices (or leave the field empty, where it allows that). Open one with **Enter** or **Space**, or simply start typing. The list narrows to the choices that contain what you type, upper and lower case alike; the choices that *start* with it come first, the matching letters are underlined, and the letters you typed are shown on the top of the list's frame. **Up** and **Down** move through it and **Enter** chooses. **Backspace** or **Delete** on a closed menu empties the field again.

Figure 12. A menu field. Status can only be one of the listed choices.

Add supply line

Item: LC Patch

> **Category**: fib

Part / Model No.: Fiber patch cable

Manufacturer: Bulk fiber

Type / Spec:

Connector:

Color:

Length:

Speed:

Quantity:

Minimum Qty:

Location:

Notes:

Class: Supply

[Save] [Cancel]

The kind of supply. Pick from the list or type your own.

Enter keep Esc undo Tab keep, next Ctrl-S save

Figure 13. Typing fib in the Category menu: two choices are left, and the letters that match are underlined.

Many other fields are **text with suggestions**: you may type anything, and the program offers values to save typing and keep spellings the same. The list starts with the standard choices (cable grades, colors, lengths, speeds, connection types), adds what is on your Config lists (manufacturers, models, types, CPUs, memory, port types), and then the values you have already used in that column. Some lists depend on other fields of the same form:

- **Room** offers the rooms of the chosen **Site**, and **Location** the locations of the chosen room, each with its kind (rack, shelf, desk, bin, closet). Change the site, and a room that is not at the new site is emptied (Figure 14). A room's **Floor** offers the floors of its site.
- **Model** offers the models of the chosen **Manufacturer**; choosing a model from the Models list fills in an empty Manufacturer, Type and Height.
- **Device** fields on a connection offer your inventory's asset tags, each with its type, manufacturer and model beside it. **Port** fields offer the ports of the device at that end, and say which are already taken (Figure 28).
- **Rack/Location** fields of connections, and **Location** of supply lines, offer every location used on either. (Items have a **Hostname** instead since 0.1.13a, which offers nothing: hostnames are unique.)
- **Project** offers the projects already used on items.
- **Optic / SFP** offers the part numbers of the transceiver modules in your supplies.

The list opens by itself as soon as you type in such a field, and **Enter** opens it on an empty one. While a suggestion list is open, **Up** and **Down** highlight a suggestion, and **Enter** or **Tab** takes it. As you type, the first suggestion that **starts with** what you typed is highlighted, so **Enter** takes it; if no suggestion starts that way, nothing is highlighted and **Enter** keeps what you typed, so a list never gets in the way of typing something new. **Esc** closes the list.

3.4 Dates

The one date field, Purchase Date, takes the year first: 2025-03-09. Slashes and dots are accepted (2025/3/9, 2025.03.09), and so is the word **today**. It is stored as 2025-03-09 whatever you typed, and a date that does not exist (the 30th of February) is refused.

3.5 Confirmations

A confirmation box (deleting, moving to Decom, changing a status, replacing a file) is answered with **y** or **Enter** for yes, and **n**, **Esc** or **q** for no. Any other key is ignored, so a stray key never confirms anything by accident. A delete, and a move to Decom, are the exception: they take **y** only, and **Enter** means no. Pasted text never answers a confirmation.

3.6 Mouse

Since 0.1.6a the program answers the mouse. Everything it does can still be done with keys; the mouse is a shortcut.

Click on	What happens
A top menu (sidebar or top bar)	That menu opens, as its number key does.
An entry of the site bar or the type bar	That page or type opens, as v or j would reach it.
A column heading	The table is sorted by that column; a second click reverses the order, as o does.
A row	The row is highlighted, and the column you clicked becomes the current one. A second click on the same row opens it , as Enter does. A click with Shift held marks or unmarks the row.
A key hint at the bottom (a add, e edit ...)	That key is pressed.
A line on the Checks or Settings page	It is highlighted; a second click opens it.
A form field	It takes the focus. A menu field opens its list; clicking it again closes the list. Clicking a suggest field that already has the focus shows all its suggestions.
An entry in an open list	It is taken as the field's value.
[Save] or [Cancel]	Saves or cancels the form.
A choice in a settings menu	It is highlighted (a theme is previewed); a second click chooses it.

The **wheel** moves three rows at a time in a table, scrolls a report, the help and menus, and moves through an open list in a form. A click outside an open dialog does nothing, so a stray click never cancels your typing. The right and middle buttons are not used.

While the program has the mouse, most terminals still select text if you hold **Shift** as you drag. To give the mouse back to the terminal entirely, set **Mouse** to **Off** on the Settings page (section 14.6).

4. Quick Start: Ten Minutes with Your Own Data

This chapter takes you from an empty program to a small, working inventory: a site with a floor, a room and a rack, two devices in it with their ports, one cable between them, a shelf of patch cables, and a first look at the Checks, the summary and a report. It is deliberately short; every step points to the chapter that explains it fully.

Tip. Use a practice file so nothing of yours is touched: `inventory -f ~/ic-practice.csv`. The other tables and lists go in files named after it beside it (`ic-practice-connections.csv`, `ic-practice-sites.csv` and so on). Delete them when you are done, or keep them and carry on.

4.1 Start it

Run `inventory` (or the practice command above). The splash appears; press any key. You land on the **Inventory**, which is empty and says so: *No items in service yet. Press a to add one; items in stock or reserved are on the Stock page (2).*

4.2 Say where things are

Places come first, because items are placed in them: site, floor, room, location. Press **7** for **Config**, which opens on its **Sites** page, on the **Sites** list. **1** steps through the lists of the page.

1. Press **a**. In a form you **move** between fields with **j** and **k** and **change** one with **Enter** (section 3.3): press **Enter**, type HQ as the **Name**, **Enter** again to keep it, and **Ctrl-S** to save.
2. Press **1** for the **Floors** list, **a**, choose HQ as the **Site** (**Enter** opens the menu, **Enter** takes it), type Ground as the name, and save. Floors are optional; skip this step if your sites are on one level.
3. Press **1** for **Rooms**, **a**, choose HQ, type Server room as the name, choose Ground as the **Floor**, and save.
4. Press **1** for **Locations**, **a**, choose HQ and Server room, name it A1; **Units** is already 42, and a location with no **Kind** is a rack. Save. (A shelf, a desk, a bin or a closet is added the same way, with its kind and no units; any number of items can share one.)

Chapter 9 covers the lists, including Models (on Config's **Types** page, **v**), which can fill in a new item's details and port counts for you.

4.3 Add two devices

1. Press **3** for the **Inventory** and **a**. A form opens. It starts on **Serial Number**; press **Tab** to move to **Component Type** and type sw. A list of suggestions opens and narrows to **Switch**; press **Down** to highlight it and **Enter** to take it.
2. Press **Tab** to **Asset Tag** and type SW-01. Fill in whichever of **Manufacturer** (Cisco) and **Model** (Catalyst 9300-48P) you like.

3. Move down to **Site**, **Room** and **Location** and choose HQ, Server room and A1; type 40 in **Position (U)** and 1 in **Height (U)**.
4. Press **Ctrl-S** to save (or **Tab** to the Save button and press **Enter**). The row appears, and is already on disk.
5. Because a switch has ports, a **Ports** form follows. Type 52 in **Network ports**, Gi1/0/{n} in **Port names** and 1 in **Power inputs**, and press **Ctrl-S**: the switch now has ports Gi1/0/1 to Gi1/0/52 and PSU1 (Chapter 6).
6. Add the second device the same way: type **Server**, tag SRV-001, rack A1 at position 10, height 2. In its Ports form give it 4 network ports named eth{n}, starting at 0, and 2 power inputs.

You need an **Asset Tag** or a **Serial Number**, and the program refuses a tag that is already in use, or a rack unit that another item already fills. Chapter 5 covers every field.

4.4 Connect them

1. Press **4** to go to **Connections**, then **a**.
2. **Cable ID**: C-0001. **Status** is already **Active**, which is what a new connection starts as. **Type**: DAC. **Speed**: 10G.
3. **A Device**: type SW and the list offers SW-01 with its type, maker and model beside it. Press **Down** and **Enter** to take it. The **A Rack/Location** and **A Serial** fields fill in from the inventory, wherever they are still empty.
4. **A Port**: press **Ctrl-N** for the switch's ports, or type gi1/0/3; the port is written as the switch has it. Then the **B** end: device SRV-001, port eth2.
5. Press **Ctrl-S**.

A port must be one of the device's ports, and a port that another active cable already uses is refused with the name of that cable. Chapter 7 explains the rules.

4.5 Stock the shelf

1. Press **2** for **Stock**, then **I** for the **Cables** (the last type on Stock's type bar), then **a**.
2. **Item**: LC/LC patch cable. **Category**: type fib, then **Enter** to choose **Fiber patch cable**. **Type / Spec**: OM4 duplex. **Connector**: LC/LC. **Color**: Aqua. **Length**: 3 m.
3. **Quantity**: 18. **Minimum Qty**: 10. Save with **Ctrl-S**. A line added on the Cables starts with the **Class** Cable, which is what lists it there.
4. Press **-** a few times as you take cables off the shelf. At 10 the quantity turns yellow and the message line tells you the line is at its minimum: LC/LC patch cable (Aqua, 3 m): 10 on hand (at or below the minimum of 10).

4.6 See what the program thinks

Press **1** for the **Summary**, and **v** for its second page, **Checks**. If you took the quantity down to the minimum, an alert says so; if you left out a serial number, a note says that. **Enter** on a check jumps to the row it is about. Chapters 11 and 12 explain them.

4.7 Sort, search and filter

Go back to the Inventory (**3**). Press **v** to see the HQ site, room by room and rack by rack, and **j** to see one type at a time. Press **l** to move to another column and **o** to sort by it, **/** to search every column for some text, or **f** to filter on the current column only. **Esc** clears a search or filter. Chapter 10 has the details.

4.8 Write a report

Press **x** on any list. Choose **CSV** or **Markdown**, whether to write only the rows on screen or every row, and a file name, then **Ctrl-S**. The message line tells you where the file went. Chapter 15 covers it, and Chapter 16 covers reading a CSV file back in.

4.9 Quit

Press **q**. There is nothing to save; every change was saved when you made it. Start the program again and everything is there.

Note. Try the same steps without a terminal: `inventory --dump -f ~/ic-practice.csv --page connections` prints a page as plain text and exits, and `--keys` presses keys first (Chapter 20).

5. The Inventory

The Inventory (press **3**) is the list of the equipment **in service**: servers, switches, routers, firewalls, access points, laptops, monitors, storage, power gear, patch panels. One row is one physical item. An item on the shelf is on the Stock menu instead, and one taken out of service for good is on Decom (section 5.3): each item is listed in exactly one place, so nothing is counted twice.

Figure 2 in Chapter 2 shows the Inventory with the demo data, sorted by type.

5.1 What is recorded

Each item has eighteen fields. Appendix B lists them with their limits; in short:

Field	What it holds
Component Type	What kind of equipment it is. Your Types list, the standard ones (Server, Switch, Router, Firewall, Laptop, Storage, UPS, PDU, Access Point, Patch Panel and more) or your own word.
Model, Manufacturer	What it is called and who made it, from your Models and Manufacturers lists or typed. The Model list offers the models of the chosen manufacturer.
Serial Number	The manufacturer's serial number. Together with the Manufacturer it must be unique.
Asset Tag	Your own label on the equipment, such as SW-CORE-01. It must be unique.
Status	Deployed (in service), Reserved (on the shelf, held for someone), Stock (on the shelf) or Decommissioned (out of service for good). A new item is Deployed unless you add it on Stock (section 5.3).
Site, Room, Location	Where it is, chosen from the Config lists (Chapter 9): the rooms of the site, and the locations of the room. A location is a rack , a shelf , a desk , a bin or a closet ; any number of items can share a shelf, a desk, a bin or a closet, and in a rack each item has its own units.
Position (U), Height (U)	In a rack: the lowest rack unit it fills, counted from the bottom, and how many units tall it is (empty counts as 1).
CPU, Memory	What is inside it, from your CPUs and Memory lists or typed: 2 x Xeon Silver 4410Y, 128 GB DDR5.
Hostname	The device's network name, such as sw-core-01 or sw-core-01.hq.example.org. Letters, digits and hyphens in dot-separated parts; kept in lower case; no two items may share one.
Project	The project the item belongs to, if any. Projects already used are offered.
Purchase Date, PO	When you bought it and under which purchase order.
Notes	Anything worth remembering, up to 1,000 characters.

The table shows the columns in the order you look for them: **Type, Model, Serial Number, Asset Tag, Status, Room**, then the rest, and starts sorted by type, then model, serial number and asset tag.

An item needs an **Asset Tag** or a **Serial Number** (or both); nothing else is required. Everything else can be filled in later, and the Checks page will remind you of what is missing (Chapter 12).

Tip. Some equipment has no serial number, and some is never tagged. Type - or n/a (also -- or na) to say so on purpose: a placeholder is kept as written, is never "already in use", and stops the Checks page from asking for the number.

5.2 Adding, editing and cloning

Press **a** to add an item. The form lists the fields in the order of the data file, starting with the serial number. Save with **Ctrl-S** or the Save button. Chapter 3 explains forms, menus and suggestions.

Figure 14. Adding an item at HQ: the Room menu lists the rooms of that site only.

The program checks the new item before it is kept:

- an Asset Tag or a Serial Number is required;
- the Asset Tag must not be in use by another item (upper and lower case do not count as different: sw-01 is SW-01);
- the Serial Number must not be in use by another item *of the same manufacturer*;
- the Asset Tag must not be another item's Serial Number, and the Serial Number must not be another item's Asset Tag (an item is found by either);
- the Purchase Date, if there is one, must be a real date;
- the Site must be on the Sites list, the Room a room of that site and the Location a location of that room; their spelling is taken from the list;
- a Position needs a rack as the Location, and the units from Position up to its Height must be inside the rack (an item 2 units tall at position 42 of a 42-unit rack does not fit) and not filled by another item in service.

If something is wrong the form stays open, names the problem and puts the cursor in the field to fix, for example U14-15 is already filled by SRV-003 (U14-15) or 'Lab' is not in the Sites list. Add it under Config (key 7) first.

When you choose a **Model** that is on your Models list, the program fills in the Manufacturer, Type and Height if they are still empty. When the item is saved and it has ports to make (by its model, or because it is a kind of item that has them), the **Ports** form follows (Chapter 6). A new item added on a site page starts at that site, and one added on a type's page as that type.

Press **e** or **Enter** to edit the highlighted item, or **E** to change only the **cell** under the highlight: its field opens already being changed, **Enter** saves it and **Esc** leaves it. Press **c** to **clone** it: the form opens with all the values copied except the Asset Tag and the Serial Number, which are left empty because they must be different (and the clone offers its own ports when saved). **C** clones at once, with no form: the copy has the Serial Number - and no Asset Tag, Hostname or rack place, ready for you to fill in later.

Tip. For many similar items, add the first one carefully, then scan the rest in (section 5.5), or clone it once for each and type only the new tag, serial number and rack position.

5.3 Status: Deployed, Reserved, Stock, Decommissioned

Every item has one of four statuses, and the status decides where it is listed:

Status	Listed on	Means
Deployed	Inventory (3)	In service. A blank or unknown status counts as Deployed.
Reserved	Stock (2)	On the shelf, held for someone or something; the note says who.
Stock	Stock (2)	On the shelf, free to use.
Decommissioned	Decom (5)	Out of service for good, kept on record until you delete it (section 5.7).

Press **m** for the **status form**: the status, and a note (saved as the item's Notes; a reserved item's note says who it is for). With rows marked (section 5.5), it sets every marked item at once. **t** steps the status at once, with no form: Deployed, Reserved, Stock, and round again.

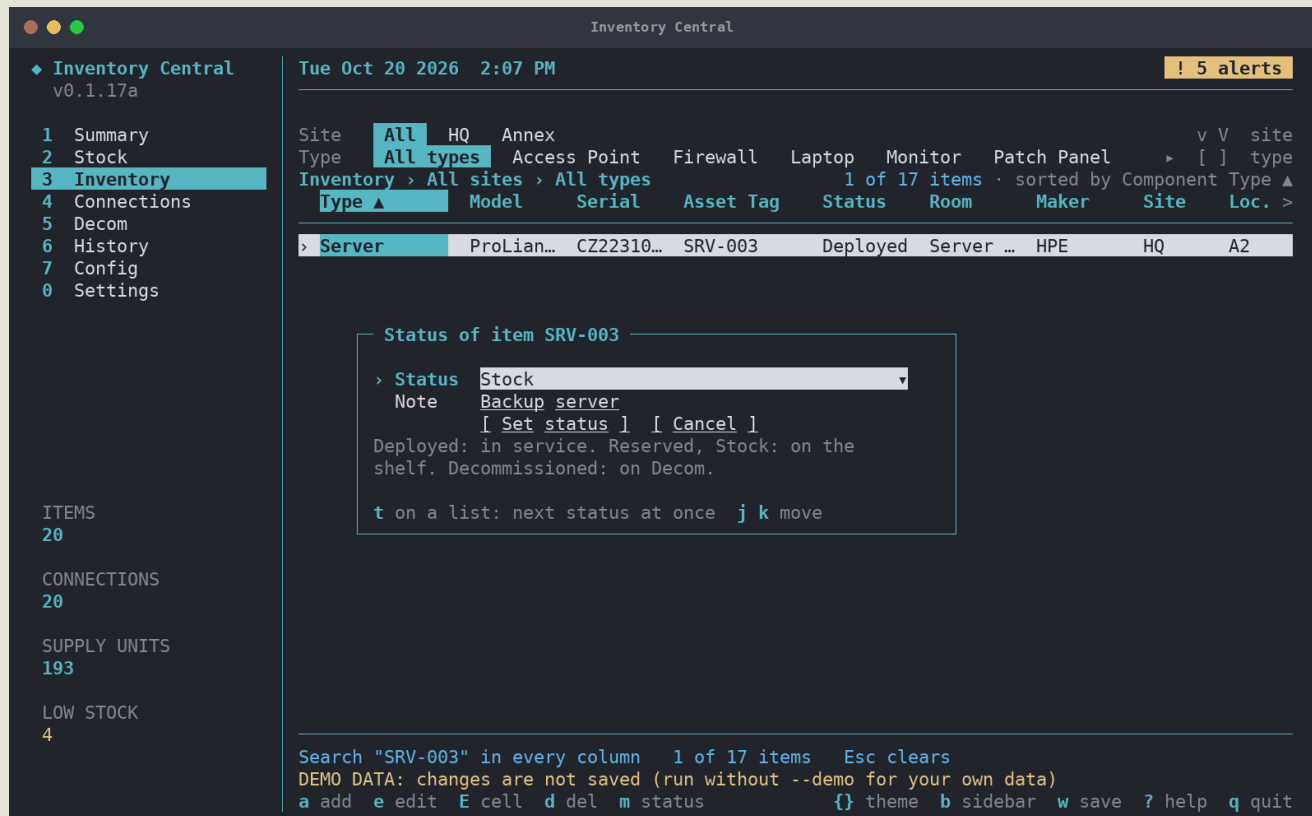


Figure 15. The status form, with Stock chosen for a racked server. Saving asks first, and says what happens to its rack place and its cables.

Taking an item **out of service** (from Deployed to Reserved, Stock or Decommissioned) asks first, and:

- takes it out of its **rack**: the Location and Position are emptied, while the Site and Room stay. A shelf, desk, bin or closet it is on is kept: that is where it is stored;
- sets its **Active** connections to **Decommissioned**, so that its ports are free; planned connections are left alone;
- and moves it to the Stock list or to Decom. The message says where it is now listed.

Putting it back to **Deployed** lists it on the Inventory again. Give it its rack position again with **e**.

5.4 Site pages and types

v steps from **All** to one page for each site on your Sites list (Figure 16). A site page lists the items at that site in place order: room by room, location by location, and in a rack from the top unit down, with a heading over each room and location. Items with no location come after the others of their room. The sort you choose orders the items within a group.

Inventory Central v0.1.17a

Tue Oct 20 2026 2:07 PM ! 5 alerts

Site All HQ Annex v V site
Type All types Access Point Firewall Laptop Patch Panel PDU Router [] type
Inventory > HQ > All types 16 items · by room, rack · then sorted by Component Type ▲

Type ▲	Model	Serial	Asset Tag	Status	Room	Maker	Site	Loc. >
▶ Floor 1 office								
Access Point	U6-Pro	F492BF5...	AP-101	Deployed	Floor 1...	Ubiquiti	HQ	
▶ Floor 2 office								
Access Point	U6-Pro	F492BF5...	AP-102	Deployed	Floor 2...	Ubiquiti	HQ	
▶ IDF 2 / B2								
Switch	2930F 4...	SG98KQ1...	SW-EDGE-01	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-02	Deployed	IDF 2	HPE Aru...	HQ	B2
▶ IT closet								
Laptop	ThinkPa...		LT-0045	Deployed	IT clos...	Lenovo	HQ	
▶ Server room / A1								
Firewall	FortiGa...	FG100FT...	FW-01	Deployed	Server ...	Fortinet	HQ	A1
Router	EdgeRou...	F09FC2A...	RTR-01	Deployed	Server ...	Ubiquiti	HQ	A1
Switch	Catalys...	F0C2145...	SW-CORE-01	Deployed	Server ...	Cisco	HQ	A1
Switch	Catalys...	F0C2145...	SW-CORE-02	Deployed	Server ...	Cisco	HQ	A1
Patch Panel	48-port...		PP-A1	Deployed	Server ...	Leviton	HQ	A1
UPS	SMX3000...	AS22102...	UPS-01	Deployed	Server ...	APC	HQ	A1
PDU	AP8941	ZA22010...	PDU-A1	Deployed	Server ...	APC	HQ	A1
▶ Server room / A2								
Storage	RS3621x...	2130TWR...	NAS-01	Deployed	Server ...	Synology	HQ	A2
Server	ProLian...	CZ22310...	SRV-003	Deployed	Server ...	HPE	HQ	A2
Server	PowerEd...	7XK2P47	SRV-002	Deployed	Server ...	Dell	HQ	A2
Server	PowerEd...	7XK2P43	SRV-001	Deployed	Server ...	Dell	HQ	A2

17 items in service - 11 types - 5 places - 3 on Stock
Inventory > HQ > All types
a add e edit E cell d del m status {} theme b sidebar w save ? help q quit

Figure 16. The HQ page: the items of each room and rack under a heading, the top of each rack first.

**** steps through the **types** of the items on the page: All types, then Access Point, Firewall, Laptop and so on (Figure 4). The site you chose stays chosen, so Inventory > HQ > Switch is the switches at HQ.

5.5 Working with many items

Marking rows. **Shift-Down** (or **J**) marks the highlighted row and the next one, and moves down; **Shift-Up** (or **K**) does the same upwards. **Insert**, or a click with **Shift** held, marks or unmarks one row; **Ctrl-A** marks every row shown, and again unmarks them. Marked rows show in the accent color and the line under the table counts them; **Esc** unmarks them all. Only rows you can see can be marked: a search, a filter, or another page or type that hides a row unmarks it.

Inventory Central v0.1.17a

Tue Oct 20 2026 2:07 PM ! 5 alerts

Site: All HQ Annex v V site
Type: All types Access Point Firewall Laptop Monitor Patch Panel [] type
Inventory > All sites > All types 17 items · sorted by Component Type ▲

Type ▲	Model	Serial	Asset Tag	Status	Room	Maker	Site	Loc. >
✓Access Point	U6-Pro	F492BF5...	AP-101	Deployed	Floor 1...	Ubiquiti	HQ	
✓Access Point	U6-Pro	F492BF5...	AP-102	Deployed	Floor 2...	Ubiquiti	HQ	
✓Firewall	FortiGa...	FG100FT...	FW-01	Deployed	Server ...	Fortinet	HQ	A1
✓Laptop	ThinkPa...		LT-0045	Deployed	IT clos...	Lenovo	HQ	
Monitor	U2723QE	CN0K2PH6	MON-0012	Deployed		Dell		
Patch Panel	48-port...		PP-A1	Deployed	Server ...	Leviton	HQ	A1
PDU	AP8941	ZA22010...	PDU-A1	Deployed	Server ...	APC	HQ	A1
Router	EdgeRou...	F09FC2A...	RTR-01	Deployed	Server ...	Ubiquiti	HQ	A1
Server	PowerEd...	7XK2P43	SRV-001	Deployed	Server ...	Dell	HQ	A2
Server	PowerEd...	7XK2P47	SRV-002	Deployed	Server ...	Dell	HQ	A2
Server	ProLian...	CZ22310...	SRV-003	Deployed	Server ...	HPE	HQ	A2
Storage	RS3621x...	2130TWR...	NAS-01	Deployed	Server ...	Synology	HQ	A2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-01	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-02	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	Catalys...	FOC2145...	SW-CORE-01	Deployed	Server ...	Cisco	HQ	A1
Switch	Catalys...	FOC2145...	SW-CORE-02	Deployed	Server ...	Cisco	HQ	A1
UPS	SMX3000...	AS22102...	UPS-01	Deployed	Server ...	APC	HQ	A1

ITEMS 20
CONNECTIONS 20
SUPPLY UNITS 193
LOW STOCK 4

4 marked (d, M, m, t) - 17 items in service - 11 types - 5 places - 3 on Stock
4 items marked: d deletes them, M changes them, m or t sets their status; Esc unmarks
a add e edit E cell d del m status {} theme b sidebar w save ? help q quit

Figure 17. **J** pressed three times: the row and the next are marked each time, four in all. The line under the table counts them.

With rows marked:

- **d** moves every marked item (or connection) to Decom, after one question;
- **m** sets their status, and **t** steps it;
- **M** changes **fields of every marked row** at once (Figure 18). The form shows a value where the rows agree and a blank where they differ, and only the fields you change are set. The Asset Tag, Serial Number, Hostname and rack position stay each row's own. It is all or nothing: if one row would break a rule, nothing changes, and the form says which row and why.

Change 3 items

Serial Number (each its own)

> **Component Type**

Asset Tag (each its own)

Manufacturer

Model

Notes

Hostname (each its own)

Purchase Date

P0

Status **Deployed**

Site **HQ**

Room (none)

Location (none)

Position (U) (each its own)

Height (U)

CPU

Memory

Project

[Change 3] [Cancel]

The rows differ here: blank keeps each one's own; a value is set on all.

Note only the fields you change are set on all **j k** move

atch Panel ▶ [] type
sorted by Component Type ▲

Maker	Site	Loc. >
...	Ubiquiti	HQ
...	Ubiquiti	HQ
...	Fortinet	HQ A1
...	Lenovo	HQ
...	Dell	
...	Leviton	HQ A1
...	APC	HQ A1
...	Ubiquiti	HQ A1
...	Dell	HQ A2
...	Dell	HQ A2
...	HPE	HQ A2
...	Synology	HQ A2
...	HPE Aru...	HQ B2
...	HPE Aru...	HQ B2
...	Cisco	HQ A1
...	Cisco	HQ A1
...	APC	HQ A1

Figure 18. Changing three marked items at once. A blank keeps each row's own value; a value you set goes on all of them.

Adding by scanning. **B** adds items by **scanning** their serial numbers or asset tags, typed or read by a barcode scanner (Figure 19). Choose what you scan for each item (the serial number, the asset tag, or both, in either order), how many items (blank: until you finish), and where the other values come from: **a form, filled in once**, or **a copy of the highlighted item** (a bulk clone, which skips the form). Then scan. The scan field takes characters as they come, so a scanner's code and its own **Enter** add the item at once and the field is ready for the next one; nothing else needs a key. A scan that is refused, a serial number already on record for instance, stays on screen with the reason, to scan again. **f** then **Enter**, **Enter** on an empty scan, or **Esc** finishes, and the message says how many items were added.

Add items by scanning

> **Scan** Serial number

How many

Other values **A form, filled in once**

[Next] [Cancel]

What you scan for each item. A scanner's own Enter goes on to the next.

Enter next **Ctrl-S** next **Esc** cancel

Figure 19. Adding items by scanning: what to scan, how many, and where the other values come from.

5.6 Viewing one item

A table has to trim long values to fit. Press **Space** to see every field of the highlighted item in full (Figure 20). Under the fields the view lists **the connections that use this device**: for each one, the cable ID, the port on this device, and the device and port at the other end, with the cable's status. **Up** and **Down** scroll when the box is taller than the window; **e** edits the item and **Esc** closes the view.

```
Item AP-101
Serial Number F492BF5A1101
Component Type Access Point
Asset Tag AP-101
Manufacturer Ubiquiti
Model U6-Pro
Notes PoE from SW-EDGE-01
Hostname ap-101
Purchase Date 2024-05-06
PO P0-24-0201
Status Deployed
Site HQ
Room Floor 1 office
Location -
Position (U) -
Height (U) -
CPU -
Memory -
Project Wi-Fi refresh 2024

Connections C-0010 eth0 to SW-EDGE-01 Gi1/0/1 [Active]

e edit Up/Down scroll Esc close
```

Figure 20. Viewing an item. The last lines list the connections that use it.

5.7 Decom: taking items out of service

d (or **DeLete**) on an item does not delete it: it **moves it to Decom** (press **5**), after a question that says what goes with it (Figure 21). The item's status becomes Decommissioned, it leaves its rack, and its active connections are decommissioned with it, exactly as a status change does. Connections go to Decom the same way. Decom lists them like any other menu, with a page per site and the decommissioned **Connections** as its last type.

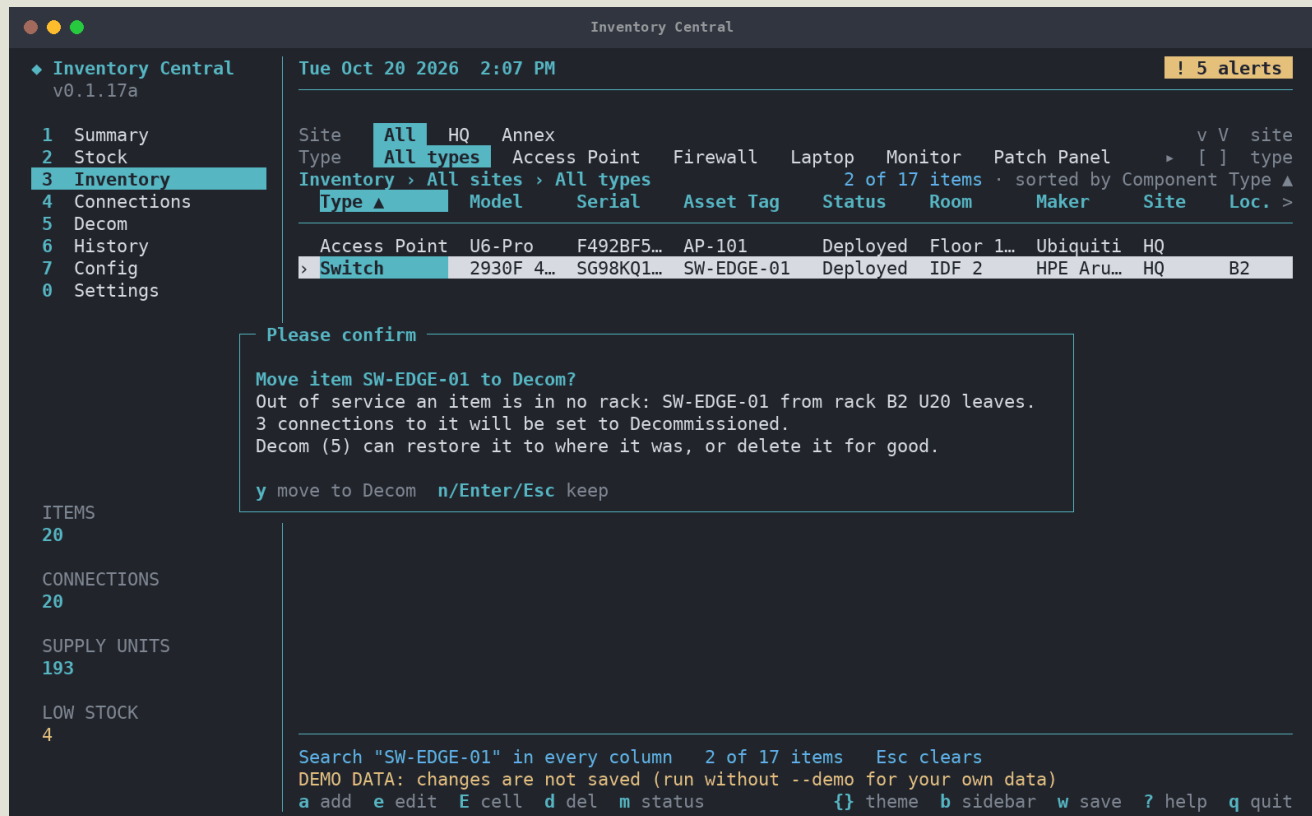


Figure 21. Moving a racked switch to Decom: it leaves rack B2, and its three connections are decommissioned with it.

On Decom:

- **R** restores the item, or every marked one, as it was before: its status, its rack place (if that is still free) and the connections that were decommissioned with it. If its rack place has been taken in the meantime, it comes back without one, and the message says so.
- **m** gives it any status, as on the other lists.
- **d** deletes it for good, after a **y**. An item's **ports are deleted with it**, and the question says how many. Connections that still name the device are not changed; the Checks page will report that the device is no longer in the inventory.

The screenshot shows the 'Inventory Central' application window. The left sidebar contains a menu with options: Summary, Stock, Inventory, Connections, **Decom**, History, Config, and Settings. Below the menu, statistics are shown: ITEMS 20, CONNECTIONS 20, SUPPLY UNITS 193, and LOW STOCK 4. The main panel displays the 'Decom' section for 'Tue Oct 20 2026 2:07 PM'. It shows a table with columns: Type, Model, Serial, Asset Tag, Status, Room, Maker, Site, and Loc. Two items are listed: 'Access Point' (U6-Pro, F492BF5..., AP-101, Decomm..., Floor 1..., Ubiquiti, HQ) and 'Firewall' (FortiGa..., FG100FT..., FW-01, Decomm..., Server ..., Fortinet, HQ). The bottom status bar indicates '2 items in Decom - 2 types' and provides a demo warning: 'DEMO DATA: changes are not saved (run without --demo for your own data)'. It also includes keyboard shortcuts: R restore, d delete for good, v sites, [] types, {} theme, b sidebar, w save, ? help, q quit.

Figure 22. Decom with two items taken out of service. **R** restores, **d** deletes for good.

The screenshot shows the 'Inventory Central' application window after a restoration. The left sidebar and statistics remain the same. The main panel displays the 'Decom' section for 'Tue Oct 20 2026 2:07 PM'. The table is now empty, and a message states: 'No decommissioned items. d on another page sends one here; d here deletes it for good.' The bottom status bar indicates '0 items in Decom - 0 types' and 'Restored 1 item and 1 connection'. It also includes the same keyboard shortcuts as Figure 22.

Figure 23. After **R**: the access point is back in service on the Inventory, with the connection that went to Decom with it.

Note. Restoring uses the change history of this session, which knows the rack place and the connections an item had. For an item decommissioned in an earlier session, **R** puts it in Stock (a connection back to Active); set its status and its rack place with **m** and **e**.

5.8 Renaming a device

An item's ports are tied to its Asset Tag (or, with no tag, its serial number). Change the tag and its ports follow: the message says how many were moved.

A connection refers to a device by its Asset Tag (or its serial number when it has no tag): a name that matches an item is stored that way when the connection is saved. Connections do **not** follow a renamed device. Checks reports each one ("device 'SW-01' is not in the Inventory"), and you can fix them by editing the connections or by importing a corrected connections file (Chapter 16).

Careful. Decide your tags before you cable. Changing a tag later means correcting every connection that names it.

5.9 Sorting, searching and filtering

The list sorts in natural order, so AP-2 comes before AP-10. **o** sorts by the current column, **0** chooses a column and a direction, **/** searches every column, **f** filters on the current column, and **x** and **i** export and import. They are the same on every list and are explained in Chapters 10, 15 and 16.

The line under the table counts what the page lists: 17 items in service - 11 types - 5 places - 3 on Stock.

6. Ports

Every item can have **ports**: its network ports, its management and console ports, its power inputs and, for a PDU or a UPS, its power outlets. The ports are what connections are made on (Chapter 7): a cable runs from one port to another, and a port can carry only one cable in service.

The ports are kept in a table of their own and listed as the **Ports**, the last type of the Connections menu: press **4**, then **[** (Figure 24).

Inventory Central v0.1.17a

Tue Oct 20 2026 2:07 PM ! 5 alerts

Site **All** HQ Annex Type All types Copper DAC Fiber MM Fiber SM Power Stack **Ports** v V site [] type

Connections > All sites > Ports 244 ports · sorted by Device ▲

Device ▲	Port	Kind	Type	Speed	Notes
AP-101	eth0	Network	RJ45 1G	1G	
AP-102	eth0	Network	RJ45 1G	1G	
FW-01	port1	Network	RJ45 1G	1G	
FW-01	port2	Network	RJ45 1G	1G	
FW-01	port3	Network	RJ45 1G	1G	
FW-01	port4	Network	RJ45 1G	1G	
FW-01	port5	Network	RJ45 1G	1G	
FW-01	port6	Network	RJ45 1G	1G	
FW-01	port7	Network	RJ45 1G	1G	
FW-01	port8	Network	RJ45 1G	1G	
FW-01	PSU1	Power in	C14 inlet		
FW-01	PSU2	Power in	C14 inlet		
NAS-01	10G-1	Network	SFP+ 10G	10G	
NAS-01	10G-2	Network	SFP+ 10G	10G	
NAS-01	PSU1	Power in	C14 inlet		
NAS-01	PSU2	Power in	C14 inlet		
PDU-A1	Inlet1	Power in	C14 inlet		
PDU-A1	Outlet 1	Power out	C13 outlet		
PDU-A1	Outlet 2	Power out	C13 outlet		
PDU-A1	Outlet 3	Power out	C13 outlet		
PDU-A1	Outlet 4	Power out	C13 outlet		
PDU-A1	Outlet 5	Power out	C13 outlet		

244 ports on 14 devices - 194 network, 18 power in, 32 power out

Connections > All sites > Ports

a add e edit d del v sites [] types o sort {} theme b sidebar w save ? help q quit

Figure 24. The Ports: every port of every item, by device and then by name.

6.1 What is recorded

Field	What it holds
Device	The item the port is on. You may type its Asset Tag or its serial number; it is saved as the Asset Tag (or the serial number when it has no tag). It must be an item in the Inventory.
Port	The port's name as printed on the device: Gi1/0/1, eth0, PSU1, Outlet 3. Unique on its item.
Kind	Network , Power in (an inlet or power supply), Power out (an outlet), Console , Management or Other .

Field	What it holds
Port Type	The connector or module, from your Port types list or your own words: RJ45 1G, SFP+ 10G, C14 inlet.
Speed	The port's speed, such as 1G or 10G; empty for power ports.
Notes	Anything worth remembering.

6.2 Ports made with the item

When you add an item whose model is on your Models list, or a kind of item that has ports (a server, switch, router, firewall, storage, UPS, PDU, patch panel or access point), a **Ports** form follows the item form (Figure 25). Its counts come from the item's model.

The screenshot shows the 'Ports of SW-NEW-01' form in the Inventory Central application. The form is titled 'Ports of SW-NEW-01' and contains several input fields: 'Network ports' (52), 'Port names' (Gi1/0/{n}), 'Port type', 'Speed', 'Power inputs' (2), 'Input names' (PSU{n}), 'Outlets', 'Outlet names' (Outlet {n}), and 'Start at' (1). Below these fields, there is a message: '{n} is the number: Gi1/0/{n} makes Gi1/0/1, Gi1/0/2 ... eth{n} with Start at 0 makes eth0 ...'. At the bottom of the form, there are buttons for 'Make ports 1', 'Cancel 1', and 'Enter keep'. On the right side of the form, there is a list of components with columns for 'Maker', 'Site', and 'Loc.'. The list includes Ubiquiti, Fortinet, Lenovo, Dell, Leviton, APC, and Cisco. The bottom of the screen shows a table of items with columns for 'Type', 'Model', 'Serial', 'Asset Tag', 'Status', 'Room', 'Maker', 'Site', and 'Loc.'. The table contains one row for a UPS item with model SMX3000...

Figure 25. The Ports form for a new switch: 52 network ports and 2 power inputs, from its model, and a pattern for the names.

Field	What it asks
Network ports	How many network ports to make.
Port names	A pattern for their names. {n} is replaced by the number: Gi1/0/{n} makes Gi1/0/1, Gi1/0/2 and so on. A pattern without {n} gets the number at the end (port makes port1; a longer word gets a space: Outlet 1).
Port type, Speed	What the network ports are, from the Port types list, and their speed.
Power inputs, Input names	How many power supplies or inlets, and their pattern (PSU{n}).
Outlets, Outlet names	Power outlets, for a PDU or a UPS, and their pattern (Outlet {n}).
Start at	The first number of the network ports: 1, or 0 for names like eth0. Power inputs and outlets always count from 1.

Enter walks down the form and makes the ports from the last field; **Ctrl-S** makes them from anywhere. The message says how many were made: Made 54 ports for SW-NEW-01; see Connections: Ports (4, then []). **Esc** makes none; the item is kept either way.

A device that needs ports of several kinds, a switch with 48 copper ports and 4 fiber uplinks for instance, gets them in two goes: make the copper ports in the form that follows the new item, then press **p** on the item for the uplinks.

6.3 Adding ports later

Press **p** on an item, on the Inventory, to open the same form at any time. Names the item already has are left out, so pressing **p** twice never makes a port twice; the message says how many were left as they were. Single ports are added, edited and deleted on the Ports like any other row (**a**, **e**, **d**; a port is deleted for good, not moved to Decom).

6.4 Ports from existing cables

Connections recorded before ports existed (or imported from elsewhere) name their ports as text. Press **P** (Shift-p) on the Ports or on Connections to make, after a question, every port that a cable names but that its item does not have yet. A port on a power cable becomes an outlet at the PDU or UPS end and an input at the other; every other port becomes a Network port with the cable's speed. Cables to devices that are not in the inventory are left alone. When there is nothing to make, the message says `Every port the connections name is on record.`

6.5 What follows the item

- **Renaming** an item's Asset Tag moves its ports to the new tag.
- **Deleting** an item for good, on Decom, deletes its ports; the question says how many. Moving it to Decom keeps them.
- An item's ports are shown under its heading on the site pages of Connections, and the ports of the device at each end of a connection are what the connection form offers.

The Checks page reports a port whose device is not in the Inventory (an alert), two ports with the same name on one item (an error) and a Kind outside the list (a note).

7. Connections

Connections (press **4**) records the cabling: which port on which device is joined to which port on which other device, by what kind of cable, at what speed. One row is one cable, or one link such as a stack or a console lead, or a **power cord** from a PDU or UPS outlet to a device's power input. A row has two ends, called **A** and **B**. Decommissioned connections are listed on Decom (press **5**, then **[]**) rather than here.

The screenshot shows the 'Connections' screen in Inventory Central v0.1.17a. The left sidebar lists navigation options: Summary, Stock, Inventory, Connections (selected), Decom, History, Config, and Settings. The main area displays a table of connections. The top bar shows the date and time (Tue Oct 20 2026 2:07 PM) and a notification for 5 alerts. The table has columns: Cable ID, Status, A Device, A Port, B Device, B Port, Type, Speed, and A Loc. The status column is color-coded: green for Active, blue for Planned, and dim for Spare. The table lists 20 connections, including network cables, power cords, and console leads. At the bottom, a summary line states: '20 connections: 17 active, 1 decommissioned, 1 planned, 1 spare'. Below this is a demo data notice and a keyboard shortcut legend.

Cable ID	Status	A Device	A Port	B Device	B Port	Type	Speed	A Loc.
C-0001	Active	SW-CORE...	Tel1/1/1	SW-EDGE...	Tel1/1/1	Fiber MM	10G	Rack A1
C-0002	Active	SW-CORE...	Tel1/1/1	SW-EDGE...	Tel1/1/1	Fiber MM	10G	Rack A1
C-0003	Active	FW-01	port1	SW-CORE...	Tel1/1/2	Fiber SM	10G	Rack A1
C-0004	Active	SW-CORE...	Tel1/1/3	SRV-001	eth2	DAC	10G	Rack A1
C-0005	Active	SW-CORE...	Tel1/1/3	SRV-001	eth3	DAC	10G	Rack A1
C-0006	Active	SW-CORE...	Tel1/1/4	SRV-002	eth2	DAC	10G	Rack A1
C-0007	Active	SW-CORE...	Tel1/1/4	SRV-002	eth3	DAC	10G	Rack A1
C-0008	Active	SW-CORE...	Tel1/1/5	NAS-01	10G-1	Fiber MM	10G	Rack A1
C-0009	Active	SW-CORE...	Stack1	SW-CORE...	Stack1	Stack	40G	Rack A1
C-0010	Active	SW-EDGE...	Gi1/0/1	AP-101	eth0	Copper	1G	Rack B2
C-0011	Active	SW-EDGE...	Gi1/0/1	AP-102	eth0	Copper	1G	Rack B2
C-0012	Active	SW-EDGE...	Gi1/0/48	sw-lab-...	Gi0/1	Copper	1G	Rack B2
C-0013	Planned	SW-CORE...	Tel1/1/5	SRV-003	eth2	Fiber MM	10G	Rack A1
C-0014	Spare	SW-EDGE...	Gi1/0/24	PP-A1	24	Copper	1G	Rack B2
P-0001	Active	UPS-01	Outlet 1	PDU-A1	Inlet1	Power		HQ / Se...
P-0002	Active	PDU-A1	Outlet 1	SW-CORE...	PSU1	Power		HQ / Se...
P-0003	Active	PDU-A1	Outlet 2	SW-CORE...	PSU1	Power		HQ / Se...
P-0004	Active	PDU-A1	Outlet 3	FW-01	PSU1	Power		HQ / Se...
P-0005	Active	PDU-A1	Outlet 4	SRV-001	PSU1	Power		HQ / Se...

20 connections: 17 active, 1 decommissioned, 1 planned, 1 spare
 DEMO DATA: changes are not saved (run without --demo for your own data)
 a add e edit d del v sites [] types o sort { } theme b sidebar w save ? help q quit

Figure 26. Connections with the demo data. Status is colored: green Active, blue Planned, dim Spare.

7.1 What is recorded

Each connection has eighteen fields, in four groups (Appendix B has the full list with limits):

Group	Fields
The cable	Cable ID (your label on the cable; unique), Status, Type, Speed
The A end	A Device, A Port, A Rack/Location, A Serial
The B end	B Device, B Port, B Rack/Location, B Serial
The details	Cable (the grade: Cat6a, OM4, OS2...), Color, Length, VLAN, Optic / SFP, Notes

- **Status** is a menu: **Active** (in use), **Planned**, **Spare**, **Disabled** or **Decommissioned**. A new connection starts as Active.

- **Type** is the kind of link: Copper, Fiber MM, Fiber SM, DAC, AOC, Console, Stack, **Power** or Other (or your own word).
- **Speed** offers 10M up to 400G. **Cable**, **Color** and **Length** offer the usual values (3 m, 10 ft, Aqua, OM4) and the ones you have used.
- **VLAN** holds the VLAN or network the link carries, or **trunk**.
- **Optic / SFP** is the transceiver in the port, for example SFP-10G-SR. It offers the part numbers of the modules in your supplies.

You need a **Cable ID**, or a **device at one end or both**. A cable with only one end is fine: it is how you record a cable that is plugged in at one side and runs to something you have not entered yet.

7.2 Devices and ports

Type in a **Device** field and the program offers the items in your Inventory by Asset Tag (or Serial Number for an item without one), each with the type, manufacturer and model beside it (Figure 27), followed by device names used on other connections that are not in the inventory. When you take a device that is in the inventory, the program fills in the **Rack/Location** and **Serial** of that end for you, wherever they are still empty: the item's Location, or its site, room, rack and units (HQ / Server room / A2 / U14-15). You can overtype them.

Add connection

Cable ID
Status Active
Type
Speed
> **A Device**
A Port
A Rack/Location
A Serial
B Device
B Port
B Rack/Location
B Serial
Cable
Color
Length
VLAN
Optic / SFP
Notes

SW-CORE-01	Switch Cisco Catalyst 9300-48P
SW-CORE-02	Switch Cisco Catalyst 9300-48P
SW-EDGE-01	Switch HPE Aruba 2930F 48G PoE+
SW-EDGE-02	Switch HPE Aruba 2930F 48G PoE+
FW-01	Firewall Fortinet FortiGate 100F
RTR-01	Router Ubiquiti EdgeRouter 12
SRV-001	Server Dell PowerEdge R740
SRV-002	Server Dell PowerEdge R740

[Save] [Cancel]

The A-end device. Picking from the inventory fills in location and serial.

Enter keep **Esc** undo **Tab** keep, next **Ctrl-S** save

Figure 27. Choosing a device. The list shows the inventory.

The **Port** field then offers **the ports of that device** (Chapter 6), each with its kind and type, and says which ones a cable in service already uses (Figure 28). On a power cable only the power ports are offered, and on any other cable only the others. Typing works as well: the port is matched to the device's ports ignoring capitals and written as the device has it.

A device name is matched to the inventory by Asset Tag first, then by Serial Number, ignoring case. A name that matches neither is allowed (a switch at another site, a device you have not entered yet), and so is any port name on such a device or on an item that has no ports on record. The Checks page lists those so that you can come back to them. A name that matches an item is saved as the item's tag (or serial number), so one port cannot be taken twice under two names.

Add connection

Cable ID
 Status Active
 Type
 Speed
 A Device SRV-003
 > A Port eth2
 A Rack/Location
 A Serial eth0 Network SFP+ 10G
 B Device eth1 Network SFP+ 10G
 B Port eth2 Network SFP+ 10G (used by C-0013)
 B Rack/Location eth3 Network SFP+ 10G
 B Serial PSU1 Power in C14 inlet
 Cable PSU2 Power in C14 inlet
 Color
 Length
 VLAN
 Optic / SFP
 Notes

[Save] [Cancel]

The A end's port, one of the device's ports (Ctrl-N lists them): Gi1/0/24.

Enter keep Esc undo Tab keep, next Ctrl-S save

Figure 28. The ports of SRV-003. Port eth2 is taken by the planned cable C-0013.

7.3 What the program refuses

A connection is checked when you save it:

- it needs a Cable ID, or a device at one end or both;
- the Cable ID must not be in use by another connection;
- **both ends must not be the same port** of the same device;
- at an end whose device has ports on record, **the port must be one of them** (SW-CORE-01 has no port Te9/9/9 on record: pick one of its ports (Ctrl-N lists them), or add it on Connections: Ports);
- a **port can only be in use once**: if the same device and port are already used by another connection that is *in service*, the form names that connection (Port Te1/1/3 on SW-CORE-01 is already used by C-0004);
- **Status** must be one of the five choices.

In service means every status except **Spare**, **Disabled** and **Decommissioned**. A retired or spare cable therefore does not hold on to its ports: you can give a port to a new cable without deleting the old cable's record, which is worth keeping as history. The Cable ID, on the other hand, stays unique across all connections, retired or not.

7.4 Power

A power cord is a connection of Type **Power**, from a **Power out** port (an outlet of a PDU or UPS) to a **Power in** port (a device's power supply or inlet). The demo has five: the UPS feeds the PDU, and the PDU feeds the core switches, the firewall and a server.

- A cable whose ports are power ports and whose Type is empty becomes a Power cable by itself.
- A Power cable must join a Power in port to a Power out port: two inputs, or two outlets, are refused.
- Any other cable must not use a power port: PSU1 is a Power in port: set Type to Power.

7.5 Site pages and types

v steps from **All** to one page for each site on your Sites list. A connection is at a site when the device at either end is there. A site page groups the cables under a heading for each **room, rack and device**, in the order of the rooms, the racks and the units from the top of each rack down, and lists each device's cables by port (Figure 29). A cable is listed under its A end's device when that end is at the site, and otherwise under its B end's. Within a group the sort you choose applies.

Inventory Central

Tue Oct 20 2026 2:07 PM ! 5 alerts

Site **All** **HQ** Annex
 Type **All types** Copper DAC Fiber MM Fiber SM Power Stack Ports
 Connections > HQ > All types 19 connections · by room, rack · then sorted by Cable ID ▲

Cable ID ▲	Status	A Device	A Port	B Device	B Port	Type	Speed	A Loc.
IDF 2 / B2 / SW-EDGE-01								
C-0010	Active	SW-EDGE...	Gi1/0/1	AP-101	eth0	Copper	1G	Rack B2
C-0012	Active	SW-EDGE...	Gi1/0/48	sw-lab-...	Gi0/1	Copper	1G	Rack B2
IDF 2 / B2 / SW-EDGE-02								
C-0011	Active	SW-EDGE...	Gi1/0/1	AP-102	eth0	Copper	1G	Rack B2
C-0014	Spare	SW-EDGE...	Gi1/0/24	PP-A1	24	Copper	1G	Rack B2
Server room / A1 / FW-01								
C-0003	Active	FW-01	port1	SW-CORE...	Tel1/1/2	Fiber SM	10G	Rack A1
Server room / A1 / SW-CORE-01								
C-0009	Active	SW-CORE...	Stack1	SW-CORE...	Stack1	Stack	40G	Rack A1
C-0001	Active	SW-CORE...	Tel1/1/1	SW-EDGE...	Tel1/1/1	Fiber MM	10G	Rack A1
C-0004	Active	SW-CORE...	Tel1/1/3	SRV-001	eth2	DAC	10G	Rack A1
C-0006	Active	SW-CORE...	Tel1/1/4	SRV-002	eth2	DAC	10G	Rack A1
C-0008	Active	SW-CORE...	Tel1/1/5	NAS-01	10G-1	Fiber MM	10G	Rack A1
Server room / A1 / SW-CORE-02								
C-0002	Active	SW-CORE...	Tel1/1/1	SW-EDGE...	Tel1/1/1	Fiber MM	10G	Rack A1
C-0005	Active	SW-CORE...	Tel1/1/3	SRV-001	eth3	DAC	10G	Rack A1
C-0007	Active	SW-CORE...	Tel1/1/4	SRV-002	eth3	DAC	10G	Rack A1
C-0013	Planned	SW-CORE...	Tel1/1/5	SRV-003	eth2	Fiber MM	10G	Rack A1
Server room / A1 / UPS-01								
P-0001	Active	UPS-01	Outlet 1	PDU-A1	Inlet1	Power		HQ / Se...
Server room / A1 / PDU-A1								

20 connections: 17 active, 1 decommissioned, 1 planned, 1 spare

Connections > HQ > All types

a add e edit d del v sites [] types o sort {} theme b sidebar w save ? help q quit

Figure 29. The HQ page of the connections: a heading for each room, rack and device.

j steps through the connection **types** on the page (Copper, DAC, Fiber MM, Power and so on), and the last type is the **Ports** of every item (Chapter 6).

7.6 Viewing, cloning, editing and deleting

Press **Space** to see all fields of the highlighted connection in full (Figure 30). **c** clones a connection, leaving the **Cable ID** and both **ports** empty, which is the quick way to add the next cable of a bundle between the same two devices. **e** edits, and **d** moves the connection to Decom, asking first; there **R** restores it and **d** deletes it for good (section 5.7).



Figure 30. Viewing a connection: every field, in the order of the form.

7.7 Sorting, searching and filtering

The page is sorted by Cable ID, then by A device and A port. Searching **Rack A1** finds every cable with that text at either end; filtering the **Status** column for **spare** lists the cables on the shelf. To find everything plugged into one device, search for its tag (the search looks in the device, port, location and serial columns of both ends), look under its heading on the site page, or press **Space** on the device in the Inventory. Chapter 10 has the details, and the summary line under the table counts the connections by status (20 connections: 17 active, 1 decommissioned, 1 planned, 1 spare; the count is of every connection, Decom's included).

8. Stock

Stock (press **2**) is the shelf: everything you have that is **not in service**. Stock is a status, not a place: the shelf can be at any site, and the site bar (**v**) shows the stock at each one. The type bar (**]** and **[**) has the types of the equipment on the shelf first, and then the three kinds of **supply line**:

Type	What it lists
All types , then each item type	Equipment whose Status is Stock or Reserved : a spare switch, a boxed access point, a laptop held for a new starter.
Supplies	Supply lines of the class Supply: optics, connectors, rack hardware, labels and ties.
Parts	Supply lines of the class Part: spare power supplies, fans, disks, batteries.
Cables	Supply lines of the class Cable: patch cables, DAC and AOC cables, bulk fiber, power cables.

The screenshot shows the Inventory Central application interface. The sidebar on the left lists navigation options: Summary, Stock (highlighted), Inventory, Connections, Decom, History, Config, and Settings. The main display area shows the Stock view for 'Tue Oct 20 2026 2:07 PM'. It includes a site bar (All, HQ, Annex), a type bar (All types, Access Point, Laptop, Switch, Supplies, Parts, Cables), and a table of items. The table has columns: Type, Model, Serial, Asset Tag, Status, Room, Maker, Site, and Loc. The items listed are: Access Point (U6-Pro, F492BF5..., AP-103, Stock, Store r..., Ubiquiti, HQ, Bin B4), Laptop (ThinkPa..., PF3XK210, LT-0046, Reserved, IT clos..., Lenovo, HQ, Shelf...), and Switch (Catalys..., FOC2145..., SW-SPARE-01, Stock, Store r..., Cisco, HQ, Shelf...). The bottom status bar shows '3 items in stock - 3 types' and a demo data warning: 'DEMO DATA: changes are not saved (run without --demo for your own data)'. Keyboard shortcuts are listed: a add, e edit, E cell, m status, t next status, {} theme, b sidebar, w save, ? help, q quit.

Figure 31. Stock with the demo data: the three items on the shelf, two in stock and one reserved, each on its shelf or in its bin.

8.1 Items on the shelf

An item on the shelf is an ordinary inventory record whose Status is **Stock** or **Reserved**: it is one record, listed here while it is on the shelf and on the Inventory while it is in service. Nothing is entered twice. Its **Location** is where it is stored: a shelf, a bin or a closet (Chapter 9), which any number of items can share. **m** sets its status (Deployed puts it back in service), **t** steps it, and the status form's note says who a reserved item is for (section 5.3). An item added with **a** here starts as Stock.

8.2 Supply lines

On the Supplies, Parts and Cables a row is a **supply line**: one kind of product, with a quantity. A line is told apart from the others by what makes the product different: its **Item name**, **Part / Model No.**, **Type / Spec**, **Connector**, **Color** and **Length**. Ten aqua 3 m LC/LC patch cables and ten yellow 10 m ones are two lines; buying five more aqua 3 m cables does not make a third, it raises the quantity of the first. The program enforces this: adding a line that matches an existing one is refused with *A supply line like this already exists: change its quantity instead.* (Upper and lower case do not count as a difference.)

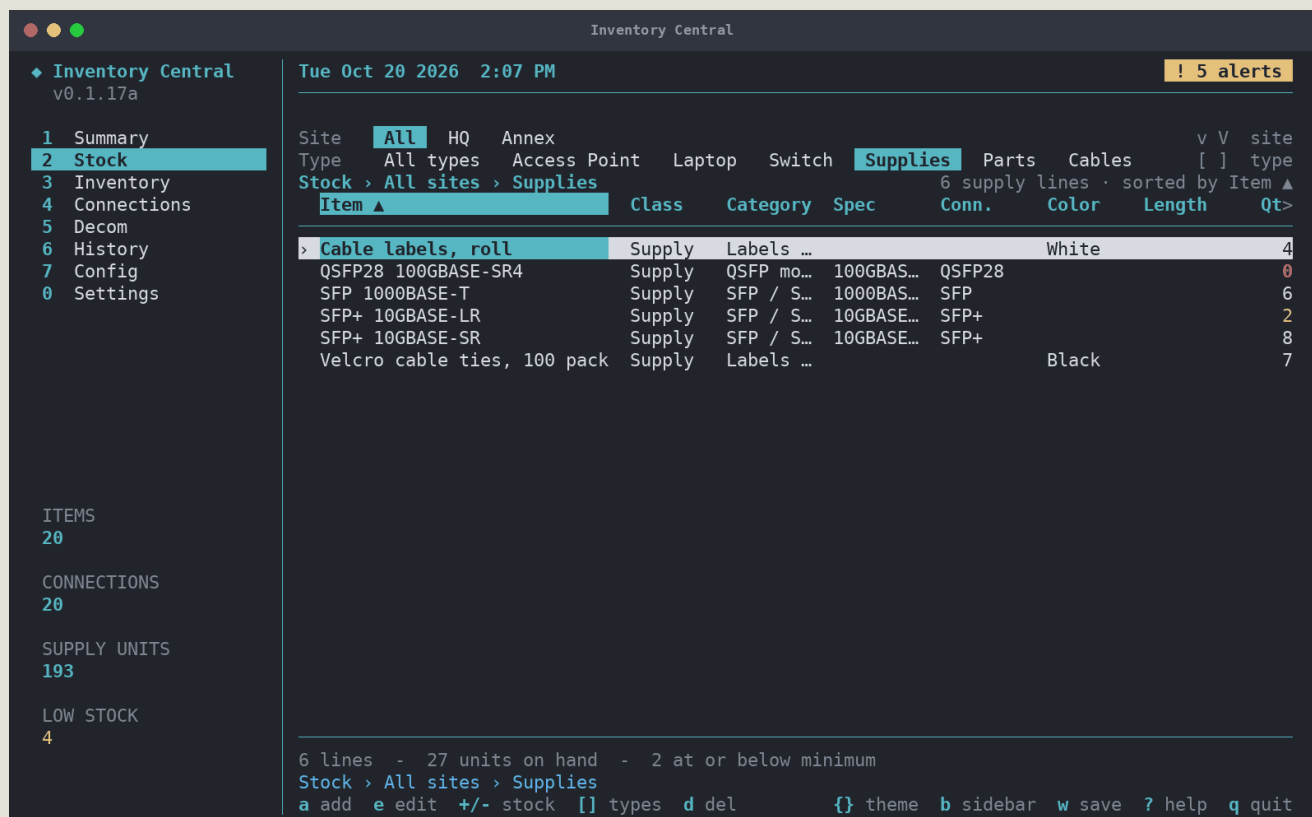


Figure 32. The Supplies. A quantity is yellow at or below its minimum and red at zero.

Field	What it holds
Item	What it is, in your words: LC/LC patch cable, SFP+ 10GBASE-SR. Required.
Class	Supply , Part or Cable : which of the three types it is listed under.
Category	The kind of supply: Fiber patch cable, Copper patch cable, SFP / SFP+ module, QSFP module, DAC / AOC cable, Bulk fiber, Connector / adapter, Power cable, Rack hardware, Labels and ties, Spare part, Tools, Other (or your own).

Field	What it holds
Part / Model No., Manufacturer	The manufacturer's part number and the maker, for reordering.
Type / Spec	The fiber grade, copper category or optic standard: OM4 duplex, Cat6a UTP, 10GBASE-SR.
Connector	The connectors (LC/LC, MPO/MTP, RJ45) or a module's form factor (SFP+, QSFP28).
Color, Length, Speed	The jacket color, the length with its unit (3 m, 10 ft) and the speed rating (10G).
Quantity	How many are on hand: a whole number, 0 or more.
Minimum Qty	Optional. The level at or below which you want a warning.
Location	Where it is stored: a room, shelf or bin.
Notes	Anything worth remembering.

A line's **Class** decides where it is listed; change it and the line moves. A line added on the Supplies, Parts or Cables starts with that class (and one added on the Cables in the Fiber patch cable category). When a line you save belongs under another type, the message says where it went ((*listed under Stock: Cables*)). A line with no class, from a file written before 0.1.17a, is a Cable when the words of its Category include cable, cord, fiber or fibre (as the built-in categories Fiber patch cable, Copper patch cable, DAC / AOC cable, Bulk fiber and Power cable all do), and a Supply otherwise.

Every text field offers suggestions: the standard lists (Appendix B) first, then the values you have used.

8.3 Counting

Quantity is a whole number of at most nine digits. The form refuses anything else, such as 12.5, -3 or ten (*Quantity must be a whole number, like 12 (use 0 if you have none)*). **Minimum Qty** follows the same rule, or may be left empty for no warning.

How to count is your choice, as long as you are consistent: patch cables by the cable, optics by the module, bulk fiber by the spool or the reel (put the length of the reel in **Length**), ties and labels by the pack or the roll.

Adjusting stock with the keys

You do not have to open a form to take a cable off the shelf. On the Supplies, Parts and Cables **+** adds one to the highlighted line and **-** removes one. (On many keyboards **=** and **_**, the same keys without Shift, work too.) The change is saved at once and the message line reports the result, for example LC/LC patch cable (Aqua, 10 m): 7 on hand (Figure 33). If the new count is at or below the minimum, the message says so, in yellow:

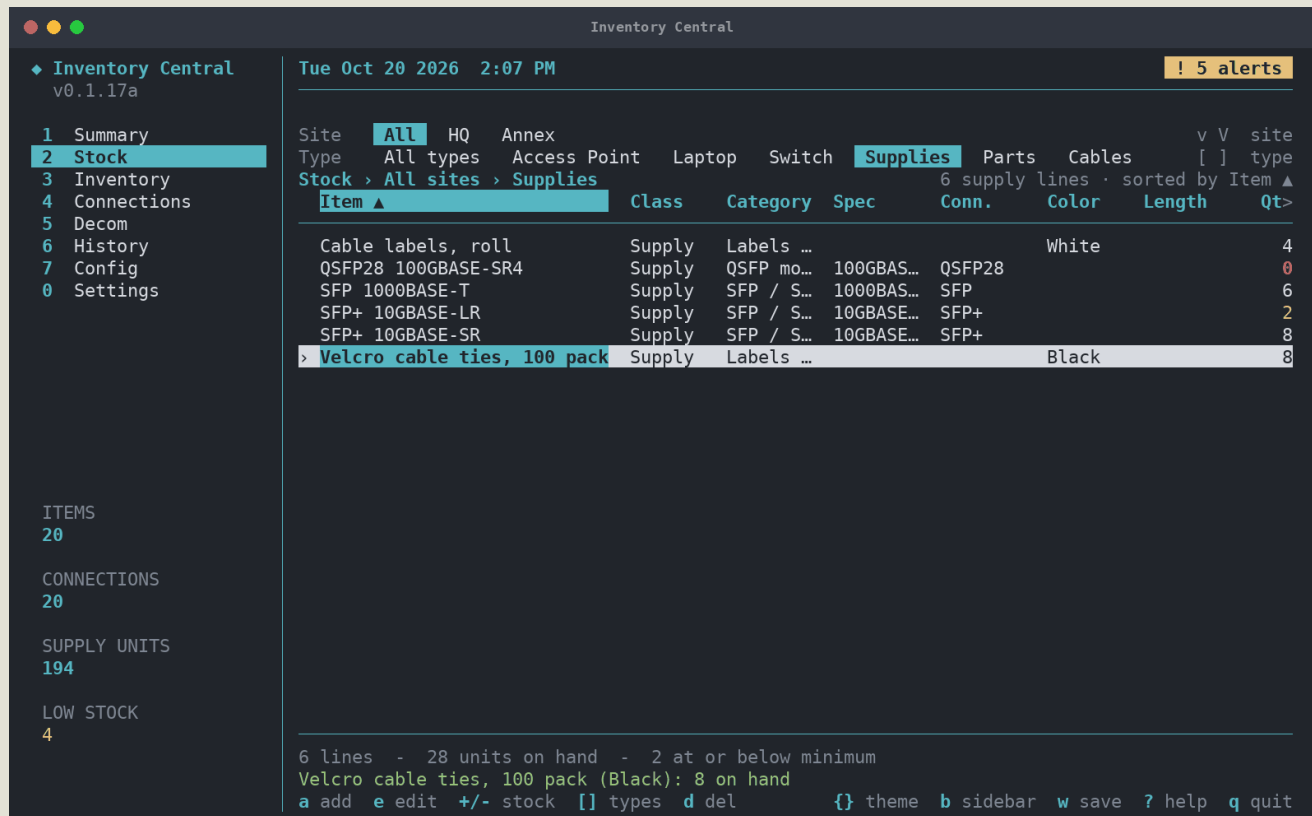


Figure 33. Pressing **+** on a line that is below its minimum: the count goes up by one and the message says the line is still at or below the minimum.

The count never goes below 0 or above 999,999,999, and a line whose quantity is not a number (possible only if the file was edited by hand) is left alone with a message asking you to correct it with **e**. For a bigger change, such as a delivery of forty cables, use **e** and type the new total.

8.4 Low stock

A line is **low** when it has a minimum greater than zero and its quantity is at or below it. Low stock shows up in four places:

- the yellow (or, at zero, red) **Qty** in the table;
- the **LOW STOCK** total in the sidebar, or **Low** in the top bar;
- the line under the table: 6 lines - 27 units on hand - 2 at or below minimum;
- an **alert** on the Checks page for each line (QSFP28 100GBASE-SR4: out of stock (minimum 2), SFP+ 10GBASE-LR: 2 on hand, at or below the minimum of 4), which also counts in the alert badge.

A line with no minimum, or a minimum of 0, is never low.

8.5 Optics and the Connections page

The part numbers of your transceiver modules (every line whose category contains SFP, QSFP or module) are offered when you fill in the **Optic / SFP** field of a connection. Keep the part number in **Part / Model No.** on the supply line and the two pages use the same spelling.

8.6 Cloning, viewing, deleting

c clones the highlighted line with the **Quantity** left empty, which is the way to add the same cable in another color or length: change what differs and type the count. **Space** shows every field in full. **d** deletes, asking first. (A supply line is deleted for good; only items and connections go to Decom.)

8.7 Sorting, searching and filtering

The supply lines are sorted by Item, then by Part No. Filter the **Category** column for *SFP* to see the modules, or search *OM4* to find every line with that grade in any column. Sorting by **Qty** puts the emptiest shelves where you can see them (press **r** to put the fullest first). See Chapter 10.

9. Config

Config (press **7**) holds the lists that the other menus draw on: the places where equipment can be, and the names that fill the menus and suggestion lists of the forms. Keeping them here means a site, a rack or a model is spelled one way everywhere, and a new item can be placed and described by choosing rather than typing.

Its lists are on two pages, moved between with **v** and **V**; **** and **[** step through the lists of a page (Figure 34):

- **Sites:** Site > Floor > Room > Location. **Sites**, **Floors**, **Rooms** and **Locations**: where equipment can be.
- **Types:** Type > Manufacturer > Model. **Types**, **Manufacturers** and **Models**, and with them **CPUs**, **Memory** and **Port types**: what equipment is.

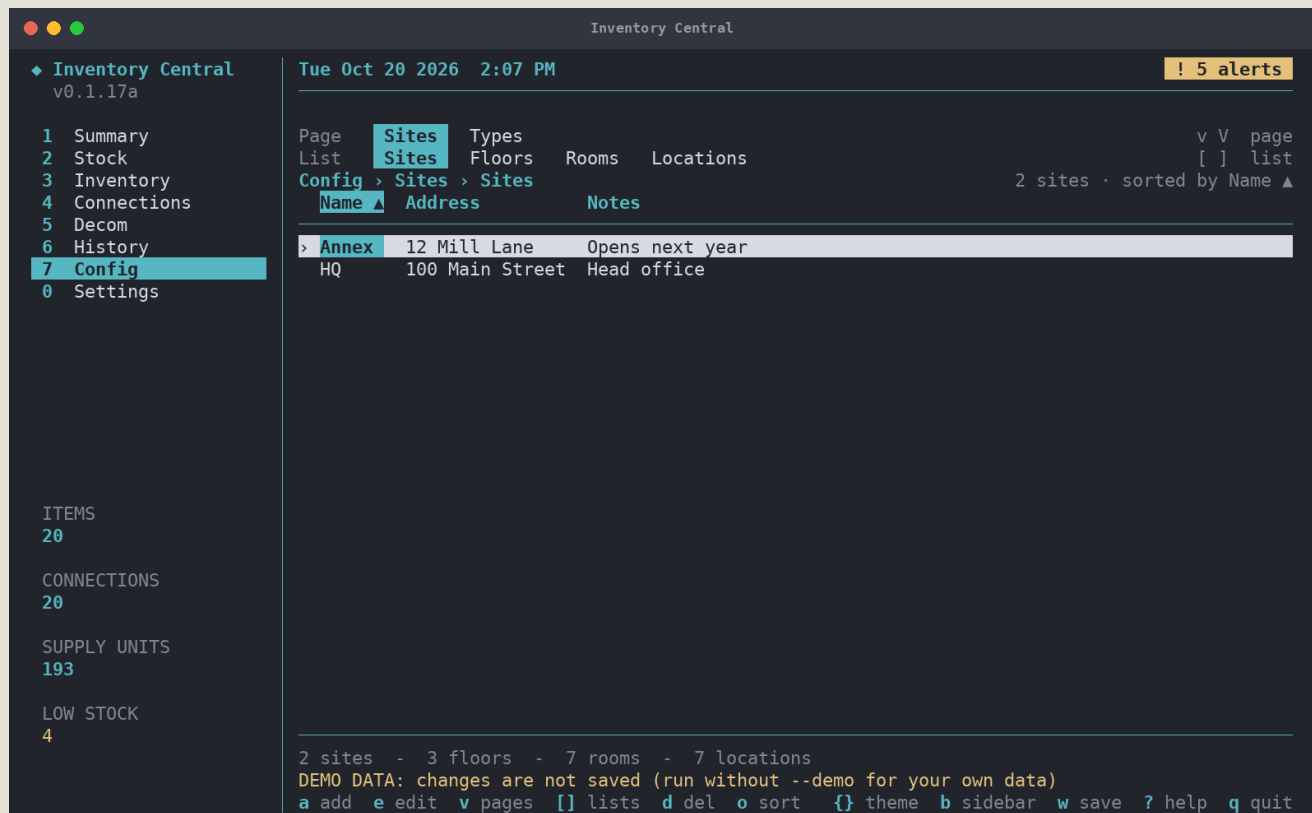


Figure 34. Config on its Sites page, on the first list, Sites. The page bar and the list bar are above the table.

List	What it holds	Used by
Sites	A building or a campus: name, address, notes.	The Site of an item; the site pages of every menu.
Floors	A floor of a site: site, name (Ground, 1, Basement), notes.	The Floor of a room, narrowed to its site.
Rooms	A room in a site: site, floor, name, notes.	The Room of an item, narrowed to its site.

List	What it holds	Used by
Locations	A place in a room: site, room, name, Kind (Rack, Shelf, Desk, Bin or Closet), units (racks only), notes.	The Location of an item, narrowed to its room; the rack-unit rules.
Types	A kind of equipment.	The Component Type of items.
Manufacturers	A maker's name and notes.	The Manufacturer of items.
Models	Manufacturer, model, type, height in units, network ports, power inputs, outlets, notes.	The Model of items, and what a new item of the model gets: its manufacturer, type, height and ports.
CPUs, Memory	Processors and memory sizes.	The CPU and Memory of items.
Port types	A connector or module (RJ45 1G, SFP+ 10G, C13 outlet) with its kind (Network, Power in, Power out, Console, Management, Other) and speed.	The Port Type of ports.

Every list answers to the usual keys: **a** adds, **e** edits, **c** clones, **d** deletes, **o** sorts, **/** searches, and **x** and **i** export and import, one list at a time.

9.1 Sites, floors, rooms and locations

The place lists nest: a floor and a room belong to a site, a room may name one of its site's floors, and a location belongs to a room. Add them in that order. A floor's and a room's name must be unique within the site, and a location's within its room, so two sites can each have a "Server room" and two rooms each an "A1". Floors are optional: a room with no floor is fine, and the Checks page notes a room whose floor is not on the Floors list.

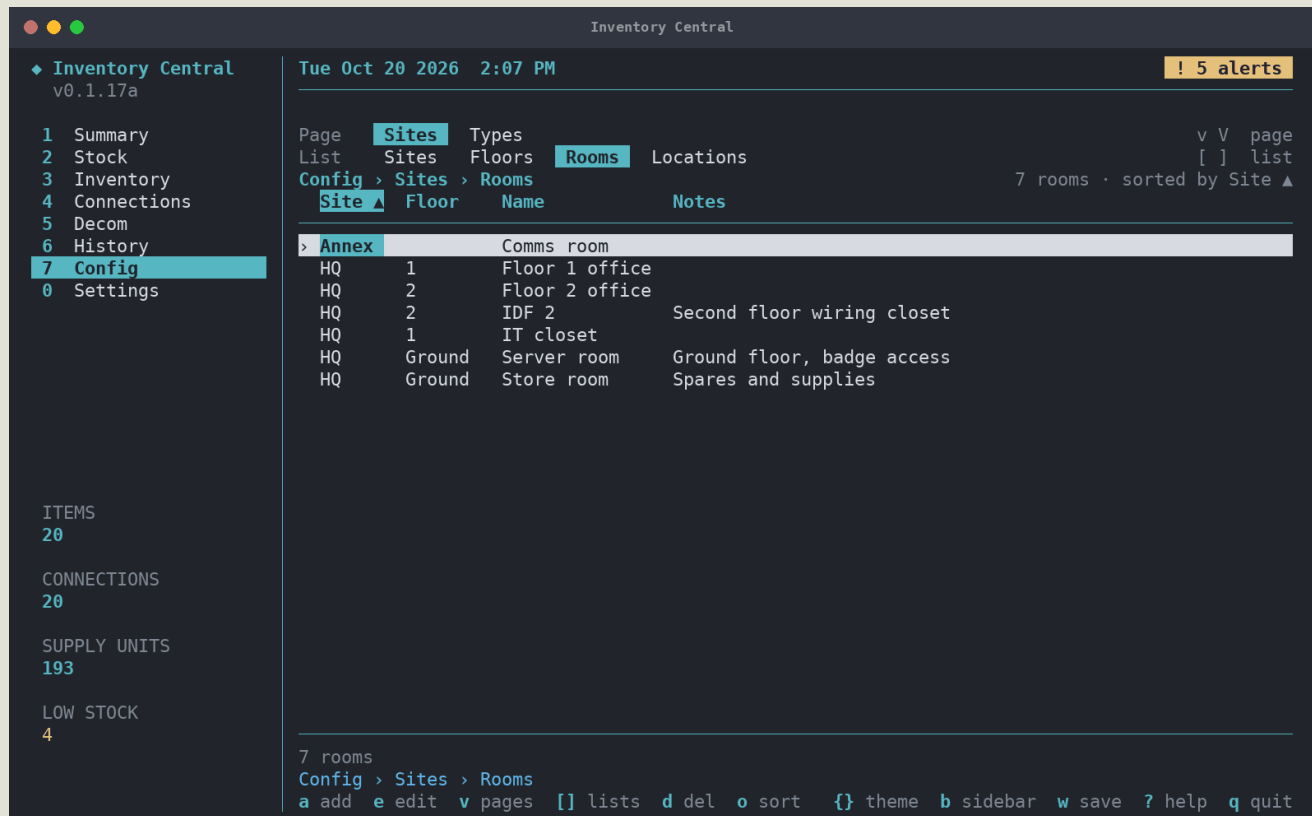


Figure 35. The Rooms list: each room with its site and its floor.

A **location** is where in the room an item is, and has a **Kind**:

Kind	What it is
Rack	Has a number of units (42 when you add one). An item in a rack has a Position and a Height, and two items cannot fill the same units (Chapter 5).
Shelf, Desk, Bin, Closet	Has no units. Any number of items share it: the laptops on a shelf, the access points in a bin. Name them as you label them: Shelf 2, Desk 12, Bin A-12.

A location with no Kind is a rack, unless its name starts with one of the other kinds (Shelf 2 is a shelf); that is how the racks of files written before 0.1.17a are read. A location that is not a rack may not have units (Only a rack has units: leave Units empty).

Inventory Central v0.1.17a

Tue Oct 20 2026 2:07 PM ! 5 alerts

Page **Sites** Types v V page
List **Sites** Floors Rooms **Locations** [] list
Config > Sites > Locations 7 locations · sorted by Site ▲

Site ▲	Room	Name	Kind	Units	Notes
> Annex	Comms room	C1	Rack	12	Wall-mounted
HQ	IDF 2	B2	Rack	24	Floor switches
HQ	IT closet	Shelf 1	Shelf		Laptops waiting to be issued
HQ	Server room	A1	Rack	42	Network and power
HQ	Server room	A2	Rack	42	Servers and storage
HQ	Store room	Bin B4	Bin		Small boxed items
HQ	Store room	Shelf 2	Shelf		Spare network gear

7 locations - 4 racks, 120 units, 18 filled
Config > Sites > Locations
a add e edit v pages [] lists d del o sort {} theme b sidebar w save ? help q quit

ITEMS 20
CONNECTIONS 20
SUPPLY UNITS 193
LOW STOCK 4

Figure 36. The Locations list: four racks, two shelves and a bin. The line under the table adds up the racks' units and the units filled.

When you add a location, the Room menu lists the rooms of the chosen site, and the form ends with its Save and Cancel buttons like every other (Figure 37).

Add location

> Site (none) ▼
Room (none) ▼
Name
Units 42
Notes
Kind (none) ▼
[Save] [Cancel]
The site the location is in.

j k move Enter change Esc cancel Ctrl-S save

Figure 37. Adding a location.

Renaming and moving

Rename a site, a floor, a room or a location (or move a room to another site, or a location to another room) by editing it, and everything that names it follows: its floors, its rooms, its locations and its items. The message says how many: Saved site Head Office; 3 floors, 6 rooms, 6 locations and 19 items updated to match.

Deleting

A site, floor, room or location that is still in use cannot be deleted. The program says what still uses it and changes nothing: Not deleted: site HQ still has 3 floors, 6 rooms, 6 locations and 19 items. Move or delete those first. Empty it first: move or delete its items, then its locations, then its rooms and floors.

9.2 Models and the other lists

The lists on the Types page are names that the forms offer. Deleting one does not change the items that use it: they keep the value as it is, and the question says how many (3 items use it and keep the value; it is no longer offered).

A **model** carries what every unit of it has in common. When you choose a model that is on the list for a new item, its Manufacturer, Type and Height are filled in where they are still empty, and its **Network ports**, **Power inputs** and **Outlets** become the counts of the Ports form that follows (Chapter 6).

Inventory Central

Tue Oct 20 2026 2:07 PM ! 5 alerts

Page Sites **Types** v V page
List Types Manufacturers **Models** [] list
Config > Types > Models
8 models · sorted by Manufacturer ▲

Manufacturer ▲	Model	Type	U	Ports	Power	Outlets	Notes
> APC	SMX3000RMLV2U	UPS	2	1	1		
Cisco	Catalyst 9300-48P	Switch	1	52	2		
Dell	PowerEdge R740	Server	2	4	2		
Fortinet	FortiGate 100F	Firewall	1	22	2		
HPE	ProLiant DL380 Gen10	Server	2	4	2		
HPE Aruba	2930F 48G PoE+	Switch	1	52	1		
Synology	RS3621xs+	Storage	2	4	2		
Ubiquiti	U6-Pro	Access Point		1	0		

ITEMS
20

CONNECTIONS
20

SUPPLY UNITS
193

LOW STOCK
4

8 models
Config > Types > Models
a add e edit v pages [] lists d del o sort {} theme b sidebar w save ? help q quit

Figure 38. The Models list of the demo, on Config's Types page, with the port counts that new items of each model get.

9.3 Importing places with items

Items name their site, room and location, and those must be on the lists. When you import items (Chapter 16), the sites, rooms and locations the file names that are not on the lists yet are added with them, for the rows that are imported; the preview says how many (Also adds to the Config lists the places the file names that are not there yet: 1 site, 2 rooms, 1 location). A location added this way gets its kind from its name. The Config lists can also be imported on their own, a list at a time.

9.4 The files

Each list is a CSV file of its own beside the inventory file: `inventory-sites.csv`, `inventory-floors.csv`, `inventory-rooms.csv`, `inventory-racks.csv` (the Locations; the file keeps the list's old name), `inventory-manufacturers.csv`, `inventory-models.csv`, `inventory-types.csv`, `inventory-cpus.csv`, `inventory-memory.csv` and `inventory-porttypes.csv`, written and backed up like the others (Chapter 17).

The Checks page reports a list entry listed twice (an error), a floor or a room whose site is not on the Sites list, a room whose floor is not on the Floors list, a location whose room is not on the Rooms list, and a rack size that is not a number (notes).

10. Sorting, Searching and Filtering

Every list (Inventory, Stock and its supply lines, Connections and its Ports, Decom, and the lists of Config) shares one set of tools for finding things, and they work as they did in the earlier version of the program. This chapter describes them once. What you see on screen is also what **x** exports (Chapter 15).

10.1 Sorting

The page opens sorted by its main column: Component Type (then Model, Serial Number and Asset Tag) on the Inventory, Stock and Decom, Cable ID on Connections, Item on the supply lines, the name on a Config list. The page heading says what the sort is (sorted by Component Type ▲), and the sorted column carries ▲ for ascending or ▼ for descending in its heading.

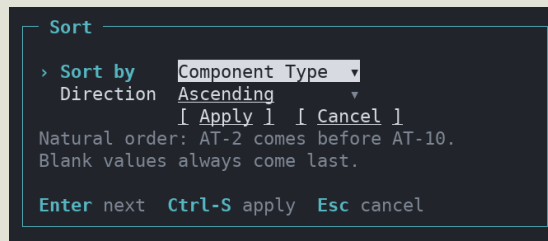


Figure 39. The Sort dialog: a column and a direction.

There are four ways to change it:

- **o** sorts by the **current column**, the one whose heading is highlighted (**h** and **l** move it). Press **o** again on the same column to reverse the order. **s** does the same.
- **O** opens the Sort dialog (Figure 39). Choose the column from a menu, choose Ascending or Descending, and save with **Enter** or **Ctrl-S**.
- **Alt-1** to **Alt-9** sort by the first to the ninth column from the left, in one key; the same key again reverses the order. (This is what the digit keys did in the earlier C version. Here the digits open a top menu. Your terminal must send Alt as Escape; in the macOS Terminal turn on "Use Option as Meta key".) Since 0.1.16a any other Alt key is taken as **Esc** followed by the key, so **Esc** pressed just before a key (which a terminal sends the same way) is not lost.
- **r** flips the direction of the current sort.

The sort is **natural**: text is compared the way people read it, so AP-2 comes before AP-10, and Gil/0/2 before Gil/0/10. Dates and numbers come out in order too (dates are always written year first, and quantities are compared as numbers). **Blank values always sort last**, whichever way the sort runs, so rows with no serial number do not crowd the top of the list.

When two rows are equal in the sorted column, the order is settled by the page's main columns (for example, Cable ID, then A Device, then A Port), and after that by the order the rows were added, so the list never jumps about.

10.2 Searching

/ opens the Search dialog (Figure 40). Type some text and press **Enter**. Only rows with that text in **any column** are shown. The match is a plain "contains", and upper and lower case do not matter: `cisco` finds `Cisco`, and `rack a1` finds `Rack A1 U40`.

The screenshot shows the Inventory Central application window. The title bar reads "Inventory Central". The top status bar shows "Tue Oct 20 2026 2:07 PM" and a yellow alert box with "! 5 alerts".

On the left is a sidebar menu with the following items:

- Inventory Central v0.1.17a
- 1 Summary
- 2 Stock
- 3 Inventory (highlighted)
- 4 Connections
- 5 Decom
- 6 History
- 7 Config
- 8 Settings

 Below the menu are statistics:

- ITEMS: 20
- CONNECTIONS: 20
- SUPPLY UNITS: 193
- LOW STOCK: 4

The main area displays a table of inventory items. Above the table, filters are set to "All" for Site and "All types" for Type. The table header shows "Inventory > All sites > All types" and "4 of 17 items · sorted by Component Type". The table columns are: Type, Model, Serial, Asset Tag, Status, Room, Maker, Site, and Loc. The first row is highlighted in blue.

Type	Model	Serial	Asset Tag	Status	Room	Maker	Site	Loc.
Switch	2930F 4...	SG98KQ1...	SW-EDGE-01	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-02	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	Catalys...	FOC2145...	SW-CORE-01	Deployed	Server ...	Cisco	HQ	A1
Switch	Catalys...	FOC2145...	SW-CORE-02	Deployed	Server ...	Cisco	HQ	A1

At the bottom of the main area, a search bar contains "switch". Below it, a message line reads: "Search 'switch' in every column 4 of 17 items Esc clears". Below that, a status line reads: "4 of 17 items match". At the very bottom, a keyboard shortcut bar shows: "a add e edit E cell d del m status {} theme b sidebar w save ? help q quit".

Figure 40. After a search for switch: the count at the right of the sort line says 4 of 20 items, and the line under the table says what is being searched for.

While a search is active the count in the page heading changes to, for example, 4 of 20 items (in the accent color), and the line under the table says what you are looking at: Search "switch" in every column 4 of 20 items Esc clears. The program also tells you how many rows matched, on the message line.

Press **Esc** on the list to clear the search. Or press **/** again: the dialog remembers the last text, and an empty text shows every row.

If nothing matches, the table says *No items match*. Press *Esc* to clear the search. (Figure 41). Nothing has been lost; the rows are only hidden.

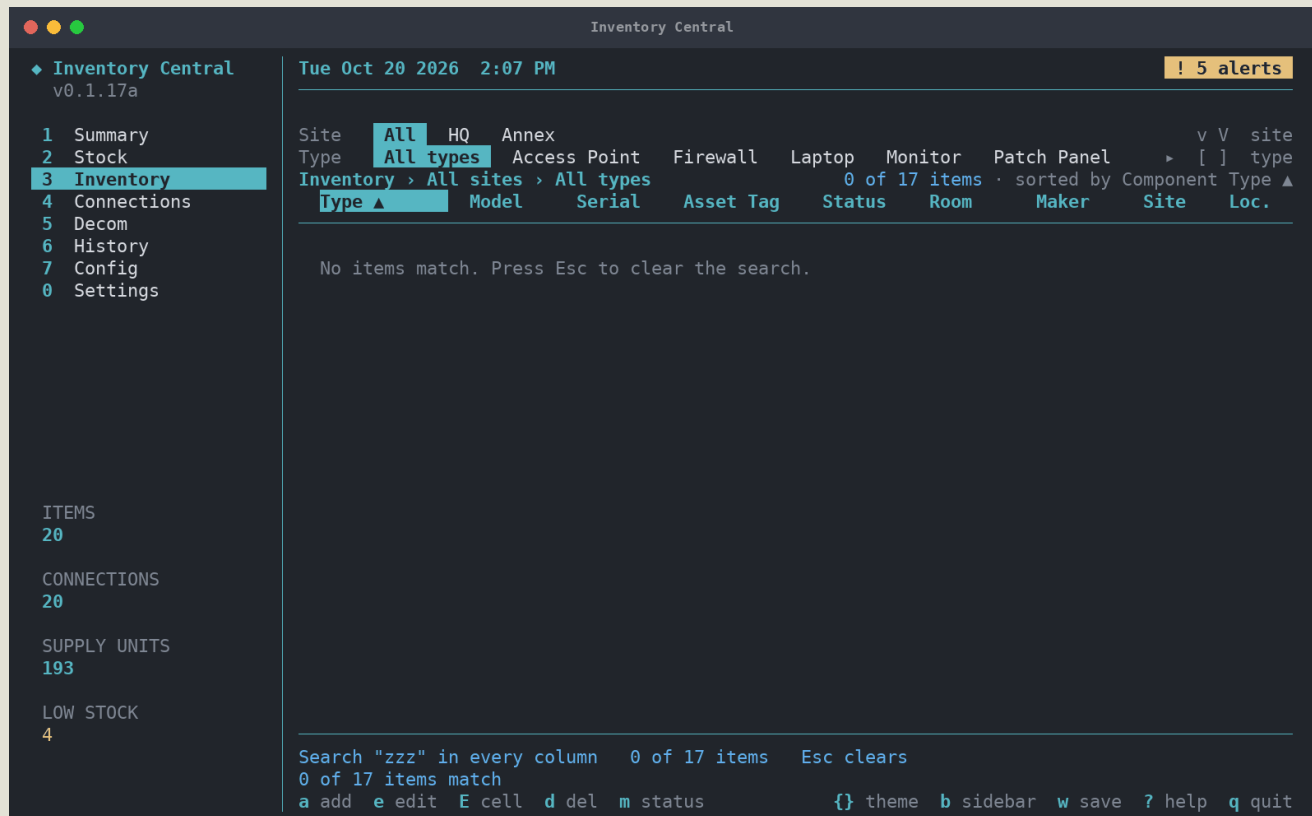


Figure 41. A search that matches nothing. **Esc** brings every row back.

10.3 Filtering on one column

f is the same idea restricted to the **current column**. Move the current column to the one you care about with **h** and **l**, press **f**, and type the text (Figure 42). Use it when a word is common but means something only in one place: *spare* in the Status column finds the spare cables, where a search for *spare* would also find every cable with "Spare" in its notes.

The screenshot shows the Inventory Central application window. On the left is a sidebar with a menu: Summary, Stock, Inventory (highlighted), Connections, Decom, History, Config, and Settings. Below the menu are statistics: ITEMS 20, CONNECTIONS 20, SUPPLY UNITS 193, and LOW STOCK 4. The main area displays a table of inventory items. At the top, it says 'Tue Oct 20 2026 2:07 PM' and '5 alerts'. The table has columns: Site, Type, Model, Serial, Asset Tag, Status, Room, Manufacturer, Site, and Loc. The 'Manufacturer' column is selected, and a filter dialog is open. The dialog shows 'Filter "cisco" in Manufacturer' and '2 of 17 items match'. Below the dialog, a table shows two items: a Switch with model Catalys... and serial FOC2145... at site HQ, location A1. The bottom of the window shows a command line with various shortcuts like 'a add', 'e edit', 'E cell', 'd del', 'm status', '{ } theme', 'b sidebar', 'w save', '? help', and 'q quit'.

Figure 42. A filter on the Manufacturer column for cisco. The line under the table names the column.

The dialog is titled with the column's name (Filter Manufacturer) and the line under the table reads Filter "cisco" in Manufacturer 3 of 17 items Esc clears. A page has one search or filter at a time; starting one replaces the other. The search and the sort are independent: you can change the sort while a filter is on, and the filter stays.

10.4 What stays and what goes

The sort, the search or filter and the current column are kept while the program runs, separately for each page and type, so you can filter the connections, look at the supplies, and come back to find the connections still filtered. On a site page the rows are grouped by place first (section 7.5), and the sort orders the rows within each group. They are **not** saved: the next time you start the program every page is sorted by its main column with nothing filtered. Adding, editing or importing rows works as usual while a filter is on, and the highlight stays on the row you just added if it is shown. If an edit makes a row stop matching the filter, it simply disappears from the page.

Note. Searching and filtering never change your data, only what is shown. An empty filter or search shows everything. If a list looks shorter than expected, check the count in the page heading for an of in it.

10.5 Finding things: some recipes

I want to see	Do this
Every cable that goes to one device	On Connections, / and type the device's tag, or look under its heading on the site page (v).
Every switch	On Inventory, move to the Type column with l , f and type switch.
Items with no serial number	Open Checks (1 , then v): each one has a note. (Sorting cannot bring blanks to the top, because blank values always sort last.)
The emptiest shelves	On Stock, on the Supplies, Parts or Cables, move to Qty and press o .
Everything in one rack	On the Inventory, v to the site: each rack has its own heading. Or filter the Location column.
Only cables that are spare or retired	On Connections, filter the Status column for spare, then for decomm.
Everything that was bought on one order	On Inventory, filter the PO column.

11. Summary

The Summary (press **1**) counts what you have. It is the first of the Summary menu's two pages; **v** goes to the second, Checks (Chapter 12). It is a report, not a table: nothing on it is edited, and **Up**, **Down**, **PgUp** and **PgDn** (or **j** and **k**) scroll it when it is taller than the window.

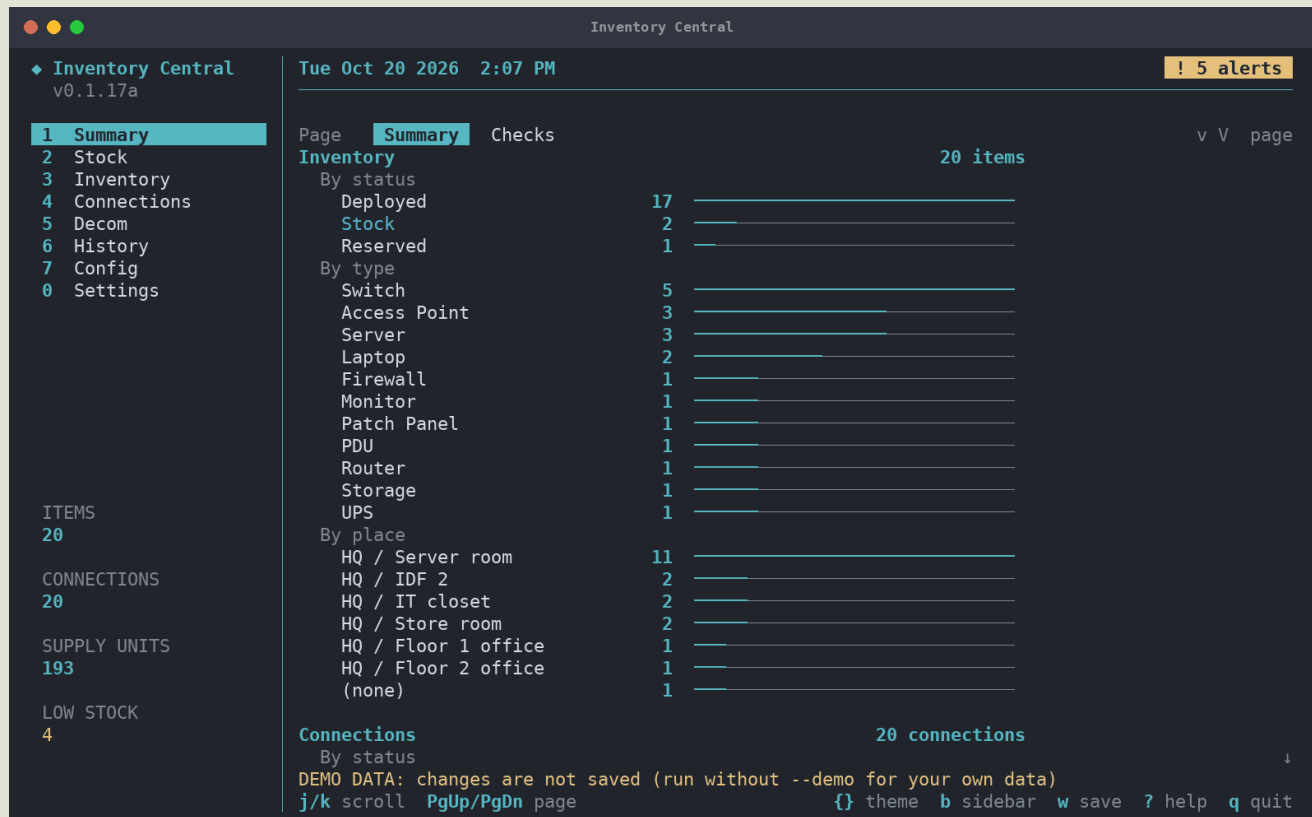


Figure 43. The Summary: three sections, each a few counts with a bar for scale.

11.1 What it shows

Inventory. The number of items, then a count **by status** (Deployed, Reserved, Stock and Decommissioned), **by type** and **by place**. An item's place is its site and room; items with no site are counted together. Types and places are compared without regard to case (Server and server are one), and items with none are counted under (none), which is always listed last.

Connections. The number of connections, and counts **by status**, **by kind** (the Type column), and **by speed**. Below those, the **busiest devices**: the devices named at either end of the most connections, with the top eight shown. It is the quick way to see which switch is nearly full.

Supplies. The number of lines and the total number of units on hand. If any lines are at or below their minimum, a yellow line says how many and points you to the Checks page. Then the **units on hand by category**.

Each group shows its ten largest values, biggest first (ties in natural order), and folds the rest into a single line such as 7 others, with their combined count. The bars are drawn to scale within each group, and drawn with # and . in text-only mode.

11.2 The totals in the margin

The four totals in the sidebar are on every page, and they are the same numbers as here:

Total	What it counts
ITEMS	Every item: in service, on the shelf and on Decom.
CONNECTIONS	Every connection, power cords and Decom's included.
SUPPLY UNITS	The quantities of every supply line, added up. A quantity that is not a whole number counts as 0 (and Checks reports it).
LOW STOCK	Supply lines with a minimum, at or below it.

In the top-bar layout three of them are on the line under the top menus: `Items 20 Links 20 Low 4`.

Note. The Summary counts *all* your rows, whatever search or filter you left on another page. To count a filtered view, look at the count in the page heading of that page, or export the view (Chapter 15); a Markdown export says how many rows it holds.

12. Checks

Checks (press **1**, then **v**: the second page of the Summary menu) reads all the tables and lists together and tells you what is wrong, what probably needs doing, and what is missing. You do not run it: it is brought up to date whenever your data changes, and the number of things that need attention is shown on every screen in the **alert badge**.

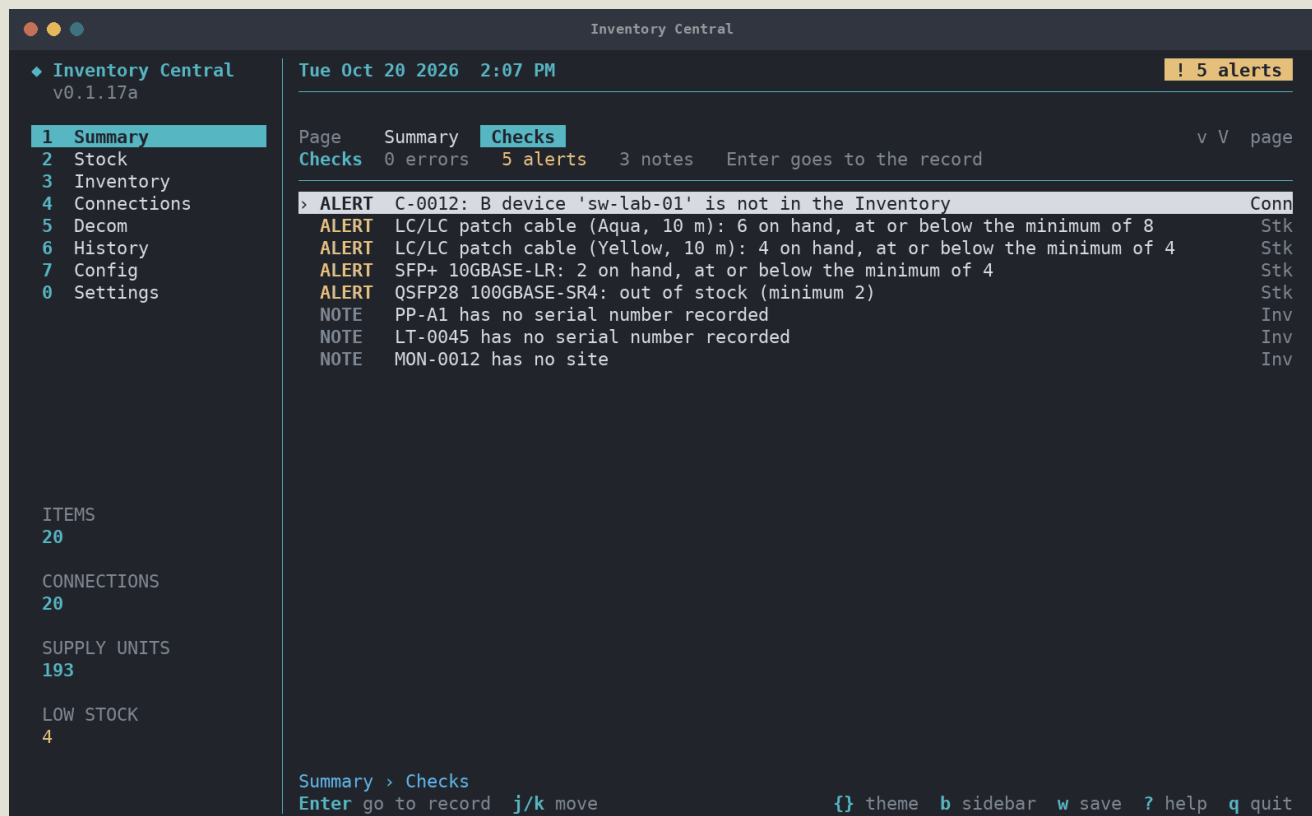


Figure 44. Checks for the demo data: no errors, five alerts and three notes.

12.1 Errors, alerts and notes

There are three kinds, listed in this order, and in the order of your records within each kind:

- **Errors** are contradictions in the data: two records that cannot both be right.
- **Alerts** are things that probably need you to act.
- **Notes** are gaps in the records.

The line at the top of the page counts each (0 errors 5 alerts 3 notes). The alert badge at the top right of every page counts errors and alerts together, and turns from yellow to red when there is at least one error. Notes never appear in the badge.

Errors

Check	What it says
Duplicate tag	Asset Tag SW-01 is used by more than one item
Duplicate hostname	Hostname sw-core-01 is used by more than one item (since 0.1.13a)
Tag is another item's serial	Asset Tag s1 is also the Serial Number of SW1; connections and ports that name it mean s1 (since 0.1.16a)
Duplicate cable ID	Cable ID C-0001 is used by more than one connection
Port conflict	Port Te1/1/3 on SW-CORE-01 is used by C-0004 and by C-0009 (A end): two cables in service claim the same port.
Same port at both ends	C-0007: both ends are the same port
Port name twice	SW-01 has two ports called Gi1/0/1
Config entry twice	Sites: hq is listed more than once

The forms refuse all of these, so they can only arrive through a file edited by hand or merged from elsewhere. A cable that is **Spare**, **Disabled** or **Decommissioned** does not hold a port, so it never makes a port conflict (section 7.3).

Alerts

Check	What it says
Active cable with an open end	C-0011 is Active but its B end has no port (or no device)
Hostname that is not one	LT-0099: hostname lt 0099: A hostname holds letters, digits, hyphens and dots, not ' ' (since 0.1.13a; the form refuses such a name, so it comes from an imported or hand-edited file)
Device not in the inventory	C-0012: B device 'sw-lab-01' is not in the Inventory
Serial number mismatch	C-0003: A serial FG100FTK1 differs from FG100FT0 in the Inventory
Out of stock	QSFP28 100GBASE-SR4: out of stock (minimum 2)
At or below minimum	SFP+ 10GBASE-LR: 2 on hand, at or below the minimum of 4
No quantity	...: no quantity recorded; it is counted as 0
Quantity that is not a number	...: the quantity 'twelve' is not a whole number; it is counted as 0
Two items on one rack unit	SRV-001 (U10-11) and SRV-002 (U11-12) are on the same units of rack HQ / Server room / A2
Past the top of the rack	NAS-01: U42-43 does not fit; rack A2 has 42 units
Port on no item	Port eth0: its device 'GHOST-1' is not in the Inventory
Wrong kind of port	C-0005: SRV-001 PSU2 is a Power in port on a DAC cable, or P-0009: both ends are Power in ports

Check	What it says
Active cable to an item out of service	C-1 is Active but its A device 'AT-1' is in stock, not deployed

The supply lines are named in full, with color and length (LC/LC patch cable (Aqua, 10 m)), because several lines share an item name.

Notes

Check	What it says
No serial number	PP-A1 has no serial number recorded
No site	MON-0012 has no site
Place not on the lists	MON-0012: Site 'Lab' is not on the Sites list in Config, and the same for a room of the site or a location of the room
Odd rack numbers	SW-EDGE-01: Position 'top' is not a whole number from 1 up, a Position with no rack as its Location, a Height that is not a number
Item out of service still in a rack	SW-SPARE-01: is in Stock but still has a rack (HQ / Server room / A1) (a shelf, desk, bin or closet is fine)
Cable port not on record	C-0004: SW-CORE-01 has no port Te9/9/9 on record (A end); P on Ports makes the missing ones
Config list	A floor or a room whose site, a room whose floor, or a location whose room, is not on its list; a rack size that is not a number
Bad purchase date	LT-0045: Purchase Date '12/31/24' is not a date (write it as YYYY-MM-DD)
Status not recognized	C-0020: Status 'live' is not one of the choices; it is treated as in service
Bad minimum	...: the minimum 'few' is not a whole number; no minimum is applied
Something about a file	A line of a settings file that was not understood, a settings file that could not be read, a data file part of which was not UTF-8 (only those bytes were read as Windows-1252), the unfinished end of a change history that was cut off (and where its bytes were kept), or a save of several files that was finished at start.

12.2 Values the forms would refuse

The forms refuse a date that is not a date, a quantity that is not a whole number and a status that is not one of the five. But a file edited by hand (or imported from elsewhere, or saved by a spreadsheet that rewrote the dates) can hold such values. Inventory Central never changes them behind your back and never silently counts them as something else. It keeps them as written, and Checks reports each one, so you can find it and decide.

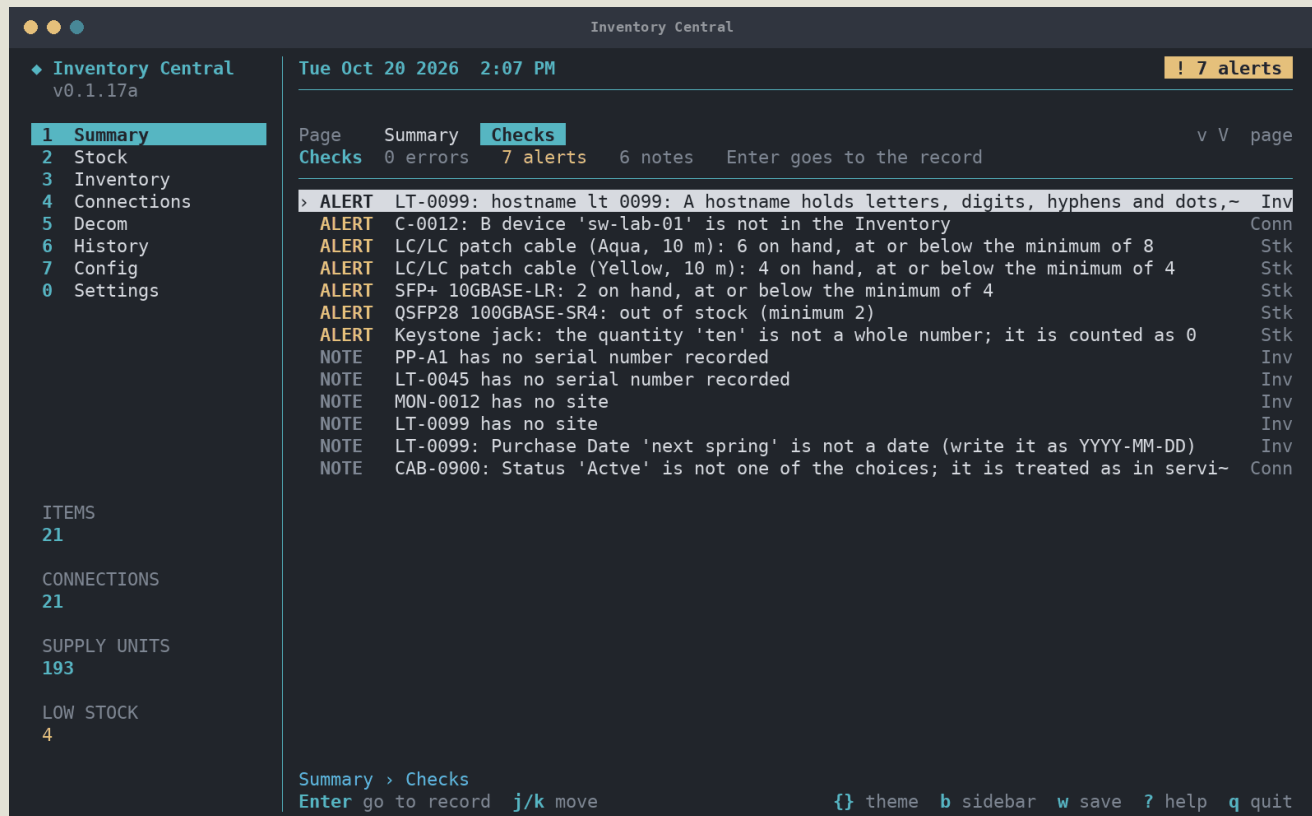


Figure 45. A data file with values the forms would have refused: a date that is not a date, a quantity that is text, a status outside the list. Each is reported, and the quantity is shown in the error color on the Supplies view.

A supply line whose quantity is not a number is counted as 0 in the totals, and it is reported only once: the "low stock" alert is not also raised for it, because a stock level worked out from a number that is not there would only mislead.

12.3 Going to the record

Move through the list with **Up** and **Down** and press **Enter** (or **e**) on a check. The program switches to the right page and view (a port on Connections' Ports, a location on the Locations list), clears any search, filter, site or type that was hiding the row, and highlights the row the check is about. Press **e** there to fix it. Checks about files and settings have no row to go to.

When there is nothing to report, the page says so: *Nothing to check: no errors, alerts or notes.*

Tip. Open Checks after an import (Chapter 16). It is the quickest way to find the rows that arrived with a missing serial number, an unknown device name or a status your list does not know.

13. History

Every change you make is recorded: each item, connection, supply line, port and Config entry that is added, changed or deleted, by whom and when. The record is a file of its own beside your data, the **change history** (`inventory.journal`), and History (press **6**) shows it as a table, newest change first (Figure 46).

No.	When	Who	Action	List	Record	Details
11	2026-10-20 14:07	alice	Added	Sites	Lab	Name: Lab; Addre...
10	2026-10-20 14:07	alice	Changed	Inventory	SW-CORE-02	Notes: Core swit...
8	2026-10-20 14:07	alice	Changed	Supplies	Cable labels, ro...	Quantity: 5 -> 6
6	2026-10-20 14:07	alice	Changed	Supplies	Cable labels, ro...	Quantity: 4 -> 5
4	2026-10-20 14:07	alice	Changed	Connections	P-0001	Status: Active -...
2	2026-10-20 14:07	alice	Changed	Inventory	UPS-01	Status: Deployed...

6 changes by 1 person - kept in memory only (demo data)
 DEMO DATA: changes are not saved (run without --demo for your own data)
 Enter details o sort r reverse / search {} theme b sidebar w save ? help q quit

Figure 46. History after a short session: an item moved to Decom, two stock changes, a note changed and a site added.

The history only grows. Nothing on this page can be edited or deleted: **a**, **e**, **d** and the other keys that change records say The history is the record of what was done: it cannot be changed here. Sorting, searching, filtering and exporting work as on every other list (Chapters 10 and 15), so you can, for example, filter **Who** for one person, or search for an asset tag to see everything that ever happened to it.

13.1 The columns

Column	What it holds
No.	The number of the change. The first change ever recorded is 1, and the numbers never skip: a gap would mean that something was taken out of the file.
When	The date and time of the change, in this computer's time zone. The file itself keeps the time in UTC, so it reads the same wherever it is looked at.

Column	What it holds
Who	The login name of the person who ran the program (the <code>LOGNAME</code> environment variable, or <code>USER</code>).
Action	Added, Changed, Deleted, or Exported. An export changes nothing, but a copy of the data has left the program, so it is recorded with the file name and the number of rows.
List	The list the record is in: Inventory, Connections, Supplies, Ports, or one of the Config lists.
Record	The record's name as the program's messages give it: an asset tag, a cable ID, a supply line, a port (SW-CORE-01 G11/0/1), a site.
Details	What changed. For a change, each field that differs, as <code>Field: before -> after</code> . For an addition or a deletion, the fields that have a value.

13.2 Reading one change

Enter or **Space** (or a second click on the row) shows the change in full, one field to a line (Figure 47).

```
Change FW-01
Number      2
When       2026-10-20 14:07
Who        alice
Action     Changed
List       Inventory
Record     FW-01
Details    Notes: Perimeter firewall
           HA pair planned -> Boxed, for IDF 2

Up/Down scroll Esc close
```

Figure 47. One change in full: two fields of an item changed in one edit.

13.3 One action, one change set

What you do with one key is recorded together. Deleting an inventory item for good also deletes its ports, so the history shows the item and each of its ports, with consecutive numbers and the same time; renaming a site lists the site and every floor, room, location and item that followed it. In the file, these lines form one **transaction**: the last of them is marked as the end, and a transaction that was never finished (because the computer stopped while it was being written) does not count.

The history is written, and handed to the disk, **before** the data files are saved. A change that reached your data is therefore always in the history.

13.4 The file

The change history is `inventory.journal` beside `inventory.csv` (with `-f kit.csv`, it is `kit.journal`). It is created at the first change, readable by you only (mode `0600`), and the program only ever adds to its end. It is plain text, one line per row changed, the columns separated by tabs, so ordinary tools can read it (the tabs are shown here as spaces):

```
IC1 7 4 2026-10-21T18:12:59Z  alice  tui  ins  sites  Lab  1  Lab  Building C  ...  #fb18a462
```

Column	What it holds
1	IC1: the format of the line.
2	The line's number, from 1, with no gaps.
3	The number of the transaction the line belongs to.
4	The time, in UTC.
5, 6	Who, and from where (tui for the terminal program).
7	What happened: ins (a row added: its values), del (a row deleted: the values it had), old and then upd (a row changed: its values before, then after), or aud (an event that changes nothing, such as an export).
8, 9	The list (items, connections, supplies, ports, sites ...) and the record's name.
10	1 on the last line of a transaction, 0 on the others.
11 onward	The row's values, in the order of the table's fields (Appendix B).
last	# and a checksum (CRC-32) of the rest of the line.

Tabs, line breaks and backslashes inside a value are written as `t`, `n` and `\`, so one line is always one line. Some ways to use it from a shell:

```
# everything alice changed, with the time and the record
awk -F'\t' '$5=="alice" {print $4, $7, $8, $9}' ~/.local/share/inventory-central/inventory.journal
# every change to one item
awk -F'\t' '$9=="SW-CORE-01"' inventory.journal
# a gap in the numbers: a line taken out of the file
awk -F'\t' 'NR>1 && $2 != p+1 {print "gap after", p} {p=$2}' inventory.journal
```

13.5 When something is wrong with it

- **The program stopped while writing.** The end of the file holds part of a change set. When the program next starts, it cuts that unfinished part off and says so on the Checks page (... bytes of an unfinished change were cut from the end of the change history). The data files were not yet written for that change, so nothing is lost that was saved. Since 0.1.16a, before an unfinished end is cut, its bytes are kept in `NAME.journal.cut-<UTC time>` (`inventory.journal.cut-20261023T091500Z`), and the note says where; if they cannot be kept, nothing is cut and no more is written to the file. A finished last change whose line only lacks its final newline (an editor or a copy tool may drop it) is kept, and the newline is added.
- **The file is damaged** somewhere other than its end: a line edited by hand, a line taken out, two lines swapped, a disk error. The program does not repair or rewrite it. It says where on the Checks page (Change history: ... is damaged at line 12 ...), leaves the file exactly as it is, and records no more changes in it; **your data is still saved as usual**. Keep the damaged file for reference, move it aside (`mv inventory.journal inventory.journal.damaged`), and the next change starts a new history.
- **It cannot be written** (a full disk, a folder you cannot write to). The message line says so, the Checks page lists it, and the data is still saved if it can be.

The checksum catches accidents, not deliberate editing: someone who can edit the file can also compute a checksum. If you need a history that cannot be quietly changed, copy the file regularly to a place the people who use the program cannot write to.

13.6 Backups and size

The file only ever grows at its end, so a copy taken at any moment, even while the program runs, is a good backup: an unfinished last change set in the copy is cut off when it is opened. Each row changed takes a line of a few hundred bytes; a thousand changes a week makes a file of a few tens of megabytes a year. The file is never cut down or rotated by the program.

With demo data (`--demo`) the history is kept in memory and goes when you quit, like the rest of the demo data.

Note. What the history is not, yet. It is a record. The program does not rebuild your data from it; the CSV files stay the data. A later version, in which several people share one inventory through a server, will keep the data in this kind of file. Values of extra columns that this version does not know (kept in the CSV files, section 17.1) are not recorded in the history.

14. The Settings Page

The Settings page (press **0**) holds the five things you can change about how the program themes and starts. It is the one page that is not about your records: nothing you do here touches the data files. Each choice is applied and saved the moment you make it, so there is nothing to confirm and no "apply" step.

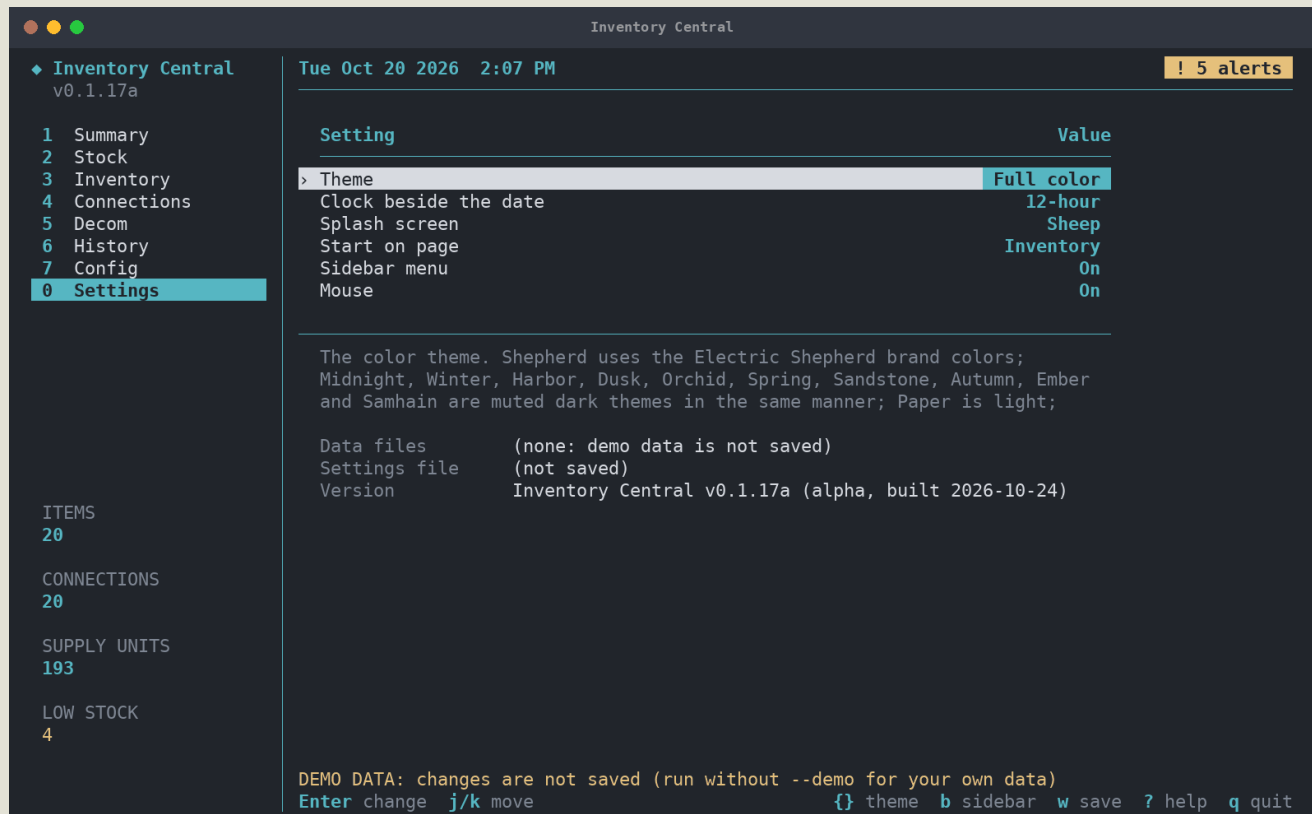


Figure 48. The Settings page. The line under the list explains the highlighted setting; below it are the places the program is reading and writing.

Up and **Down** (or **k** and **j**) move the highlight, and **Enter** (or **e**) opens a small menu for the highlighted setting. In that menu:

Key	What it does
Up / Down , k / j , Home / End	Move through the choices.
Enter or Space	Apply the highlighted choice and save it.
Esc , q or Ctrl-C	Cancel. Whatever you tried while moving is put back as it was.

The settings are remembered in a small text file of their own, apart from your data (section 14.8). The page also shows, under the list, which **Data files** and which **Settings file** are in use and the **Version** of the program, which is the first thing to quote if you ever report a problem.

14.1 Theme

The **Theme** row (called Colors before 0.1.17a) chooses one of seventeen color themes, listed from cool to warm after the default, with the one light theme and the one-color themes last. **{** and **}** step through them from any page, without opening the menu, and save the choice the same way; the message line names the theme and its place. Moving through the menu shows each one at once behind the menu, so you can judge it on your own screen before you choose; **Esc** puts the old theme back.

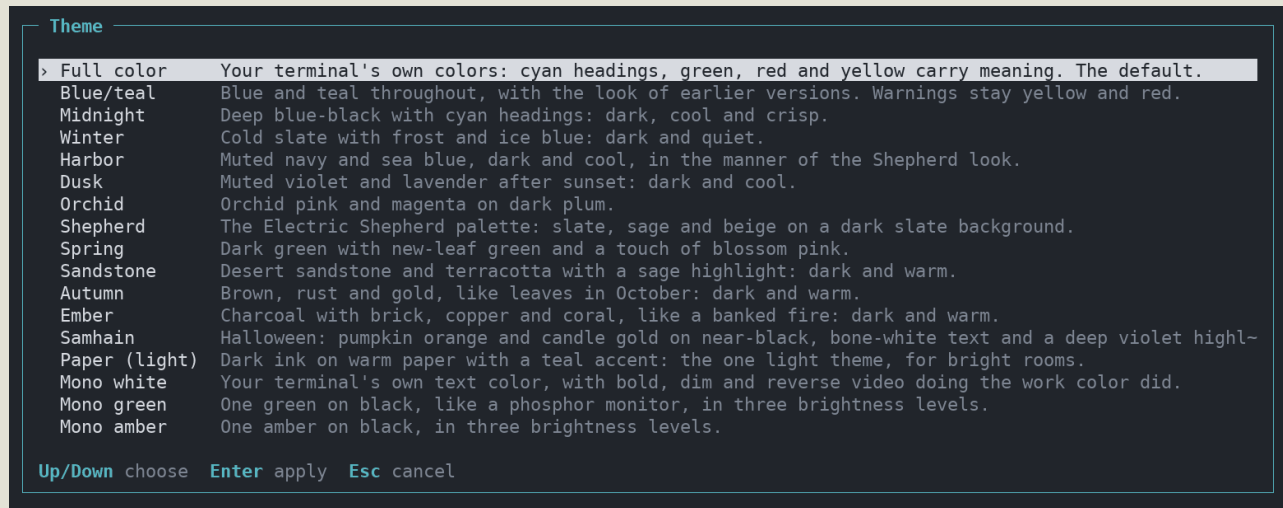


Figure 49. The Theme menu. Each theme is previewed on the page behind as you move.

Theme	What it is
Full color	The default. Your terminal's own colors: cyan headings and borders, green, yellow and red for what is good, doubtful and wrong. Text and background are your terminal's own, so a light terminal stays light.
Blue/teal	Blue and teal throughout, in the manner of earlier versions of the program. Warnings stay yellow and errors red.
Midnight	Blue-black with cyan headings: dark, cool and crisp. The coolest.
Winter	Cold slate with frost and ice blue: dark and quiet.
Harbor	Muted navy and sea blue, dark and cool, in the manner of Shepherd.
Dusk	Muted violet and lavender: dark and cool.
Orchid	Orchid pink and magenta on dark plum.
Shepherd	The Electric Shepherd palette: beige text on slate, sage for headings and the highlighted row. This theme paints its own dark slate background over the whole window.
Spring	Dark green with new-leaf green and a touch of blossom pink.
Sandstone	Desert sandstone and terracotta with a sage highlight: dark and warm.
Autumn	Brown, rust and gold: dark and warm.
Ember	Charcoal with brick, copper and coral: dark and warm.
Samhain	Halloween: pumpkin orange and candle gold on near-black, bone-white text and a deep violet highlight. The warmest.

Theme	What it is
Paper	Ink on warm paper with a teal accent: the one light theme, for bright rooms. It paints its own light background.
Mono white	No color at all. Your terminal's own text color, with bold , dim and reverse video doing the work that color did elsewhere; a supply that is out of stock is bold and underlined, one that is low only bold (since 0.1.16a). For printing, for screenshots in black and white, and for terminals where color is unwelcome.
Mono green	One green on black, in three brightness levels, like a phosphor monitor.
Mono amber	One amber on black, in three brightness levels.

Midnight, Winter, Harbor, Dusk, Orchid, Spring, Sandstone, Autumn, Ember and Samhain are the **muted dark themes**; Paper is made the same way on a light background. Like Shepherd, each paints its own background and is made of eleven colors with the same jobs in every one: text, panels, fields, dim text, headings, the selected row, errors, warnings, rules. Every one is held to the same contrast: text at least 7 to 1 against its background, headings and every warning color at least 4.5 to 1, rules at least 3 to 1 (the WCAG measures), and a test of the program checks it.



Figure 50. The Inventory page in Full color, Blue/teal, the three one-color themes and Shepherd. The meaning of the colors is the same in every one; only the paint changes.

Whichever theme you choose, color is never the only carrier of meaning. A highlighted row also has the › mark at its left edge; a problem is also worded ("5 alerts", "ALERT", "ERROR"); a low stock level is also a number you can read. That is why the one-color themes lose nothing.

Shepherd, the muted dark themes, Green and Amber paint a background color of their own. Full color, Blue/teal and Mono white leave the background alone. When you quit, the program restores your terminal's own colors.

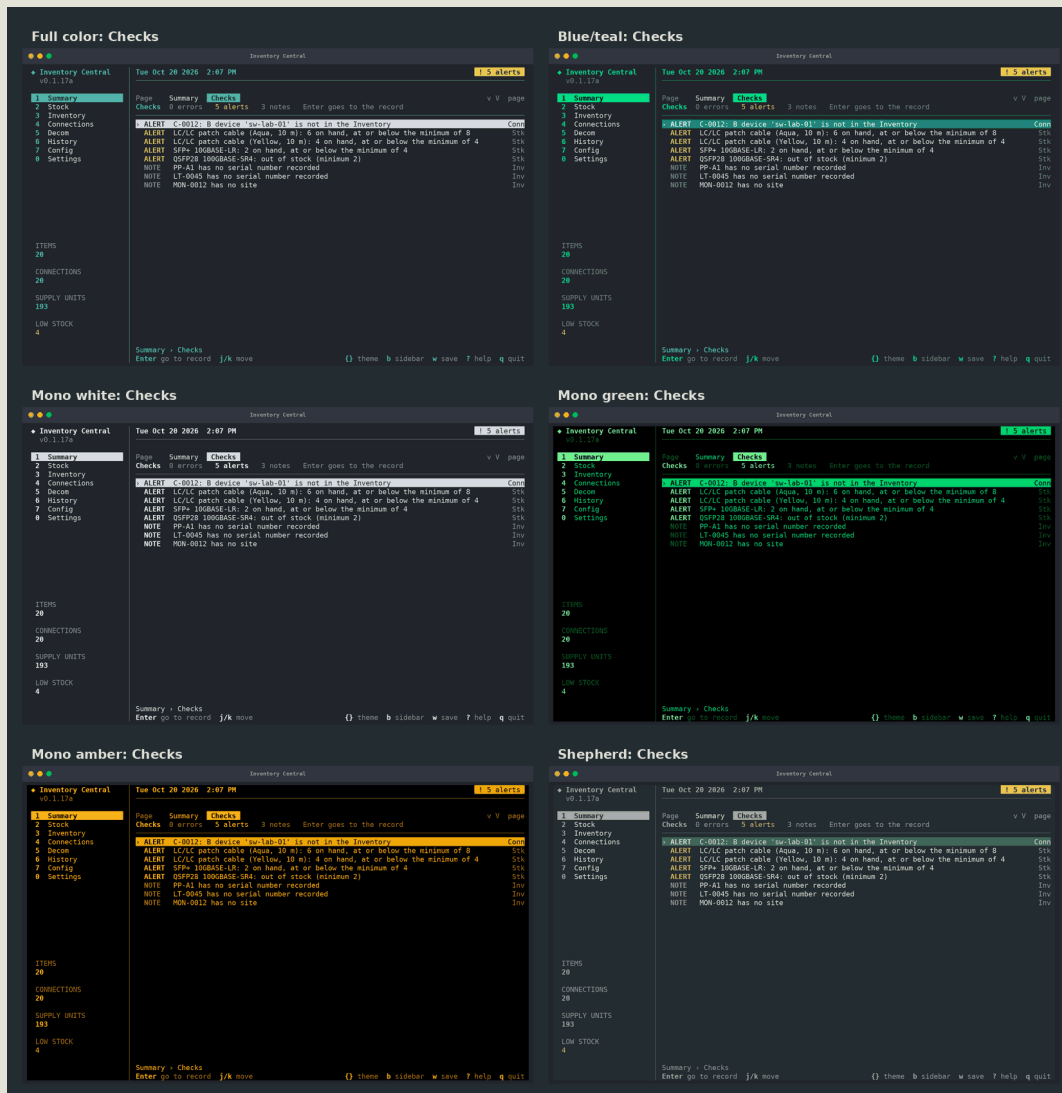


Figure 51. The Checks page in the same six themes.



Figure 52. The Inventory page in Shepherd and the seven muted dark themes, from cool to warm: Winter, Harbor, Dusk, Shepherd, Spring, Autumn, Ember and Samhain. (Midnight, Orchid, Sandstone and Paper, new in 0.1.17a, are not pictured.)

Note. How many colors your terminal needs. If the environment variable `COLORTERM` is `truecolor` or `24bit`, which is the case in nearly every current terminal, Shepherd uses exactly the brand colors. Otherwise the program uses the nearest of the 256 palette colors, which themes close but not identical. A terminal with fewer than 256 colors will show the Full color theme well and the others only roughly; Mono white is then the clear choice.

Choosing a theme for one run

`inventory -c shepherd` (or `--colors`) uses that theme for this run only and does not change the saved setting. The names are `full`, `teal`, `midnight`, `winter`, `harbor`, `dusk`, `orchid`, `shepherd`, `spring`, `sandstone`, `autumn`, `ember`, `samhain`, `paper`, `white`, `green` and `amber`. Opening the Theme menu while `-c` is in force shows the theme you have saved, and choosing one there replaces the `-c` theme for the rest of the run; **Esc** keeps it.

If the environment variable `NO_COLOR` is set to anything but an empty value, the program starts in Mono white, following the convention described at no-color.org. `-c` wins over `NO_COLOR` when both are given. Neither is written to the settings file.

14.2 Clock beside the date

The header shows the date and, if you wish, the time. The choices are **12-hour** (2:07 PM, the default), **24-hour** (14:07) and **Hidden**, which leaves only the date. The program reads your computer's clock once a minute, in your own time zone, using the standard `date` command, and so the time moves on without a key being pressed. If `date` cannot be run, it falls back to UTC.

14.3 Splash screen

The **Splash screen** row chooses what appears when the program starts (section 1.4).

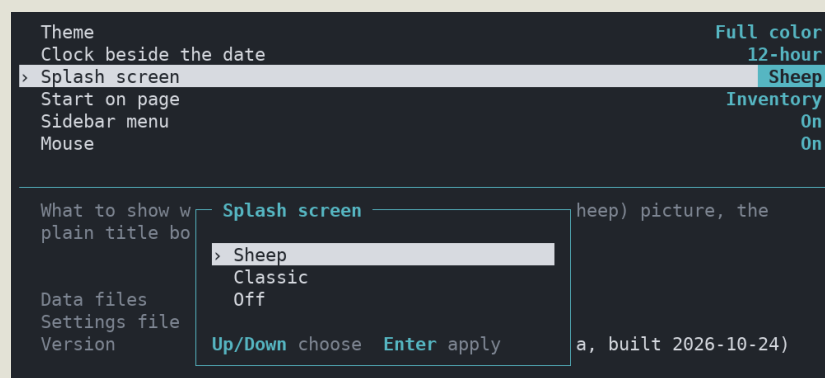


Figure 53. The Splash screen menu.

Choice	What you get
Sheep	The default. The Electric Shepherd sheep above the name of the program, the company, the version and the build date.
Classic	A framed title box with the same words, as in Budget Buddy.
Off	No splash; the program opens straight on its first page.

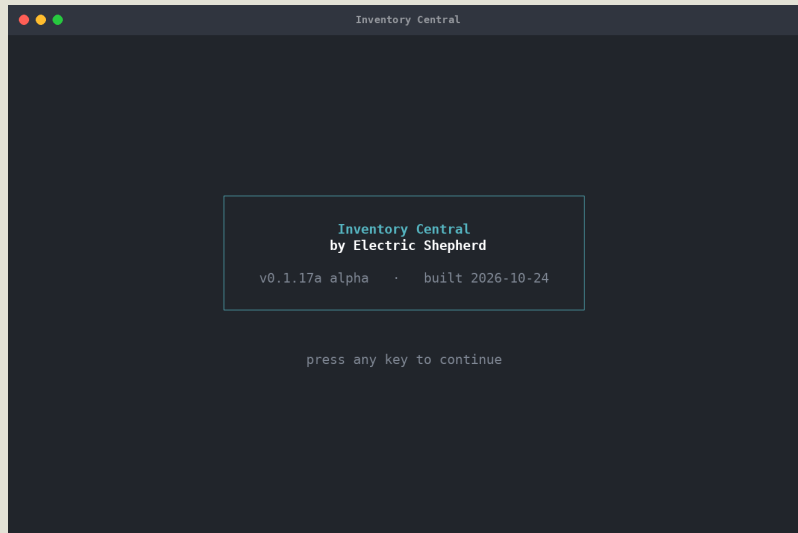


Figure 54. The Classic splash.

The sheep is always the same size and sits in the middle of the window. It needs a window at least 18 rows high and 26 columns wide, which an ordinary 80 by 24 terminal easily provides (Figure 55). A window too short for it, or a terminal that is not set to UTF-8, shows the Classic splash instead; a window too small even for that gets the same words without the frame.

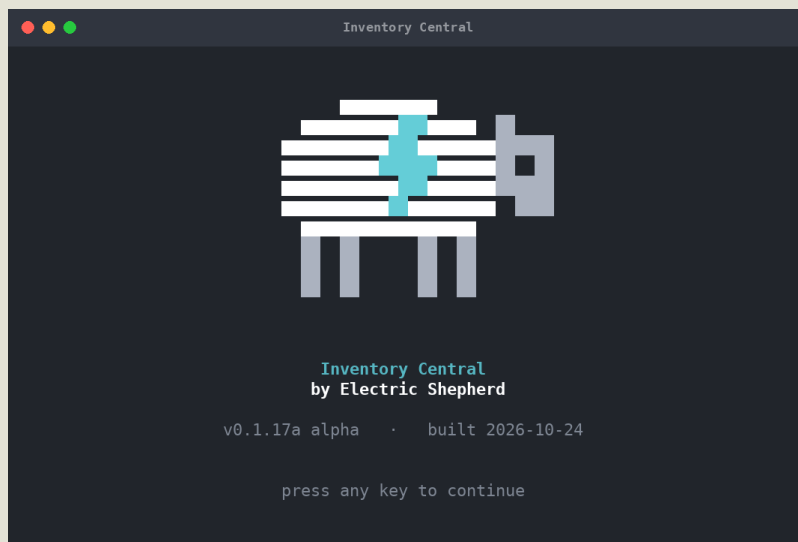


Figure 55. The sheep splash in an 80 by 24 window.

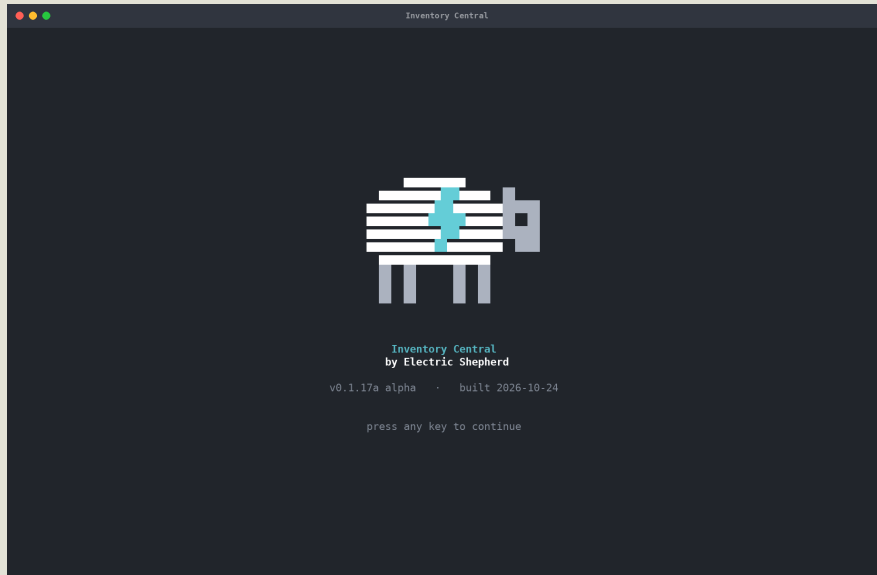


Figure 56. The same sheep in a large window (140 by 42). Only the space around it grows.

The choice here can be overridden for one run: `-n (--no-splash)` and `INVENTORY_NO_SPLASH=1` both skip the splash without changing the setting.

14.4 Start on page

The **Start on page** row chooses the menu that is open when the program starts: Summary, Checks, Stock, Inventory (the default), Connections, Decom, History, Config or Settings. This takes effect the next time you start the program. `-p PAGE` opens a different page for one run (Chapter 20).

14.5 Sidebar menu

The **Sidebar menu** row switches the sidebar **On** (the default) or **Off**. **b** does the same from any page and says so on the message line. The sidebar only appears in windows of 100 by 24 or more; with it off, those windows use the top bar too, and the page gets the whole width (Figure 3). The choice is saved like the others.

14.6 Mouse

The **Mouse** row switches mouse clicks and the wheel **On** (the default) or **Off** (section 3.6). While it is on, the terminal reports the mouse to the program, so dragging to select text needs **Shift** held in most terminals. Off, the program never asks for the mouse, and the terminal's own selection works as before. The change takes effect at once.

14.7 Web page (server)

This row appears only when the program is connected to a server (`-S` or `-R`) as someone with the **admin** role. It shows whether the server's web page (Chapter 19) is **Open** or **Closed**, and changing it opens or closes the page for everyone at once. It is not saved in the settings file: the server records it in its history, and every admin's program shows the same. Closing the page does not stop the server or the terminal programs connected to it.

14.8 The settings file

The settings are stored as six lines of plain text:

```
# Inventory Central settings. Edit here or in the program (Settings page).
colors=shepherd
clock=24h
splash=sheep
start=connections
sidebar=on
mouse=on
```

You may edit the file by hand. The accepted values are:

Key	Values
colors (or theme)	full, teal, midnight, winter, harbor, dusk, orchid, shepherd, spring, sandstone, autumn, ember, samhain, paper, white, green, amber
clock	12h, 24h, off
splash	sheep, classic, off
start	summary, checks, stock, inventory, connections, decom, history, config, settings (the older supplies means stock)
sidebar	on, off
mouse	on, off

Blank lines and lines that begin with # are ignored, and so are keys the program does not know. A value it does not understand leaves that one setting at its default, and the program says so on the Checks page as a note (settings line 3: bad value 'purple' for colors, with the name of the file in front); the other lines still apply. A missing settings file simply means the defaults. Since 0.1.16a, comments, blank lines and settings the program does not know are kept when it saves, and a byte-order mark at the start (as Notepad writes) is ignored. A settings file that cannot be read (or is larger than 64 KB) gives a note on the Checks page, the defaults are used, and the file is not saved over. In demo mode (- -demo) the settings are neither read nor saved.

15. Exporting Reports

Any list can be written to a file: as **CSV**, which a spreadsheet opens and which Inventory Central can read back in (Chapter 16), or as **Markdown**, a readable report that pastes into a wiki, a ticket or a README. You can write everything, or only the rows you are looking at, which makes a filtered report a matter of a few keys: filter, export, done.

Press **x** on any list: the Inventory, Stock (its items or its supply lines), Connections or its Ports, Decom, or a Config list.

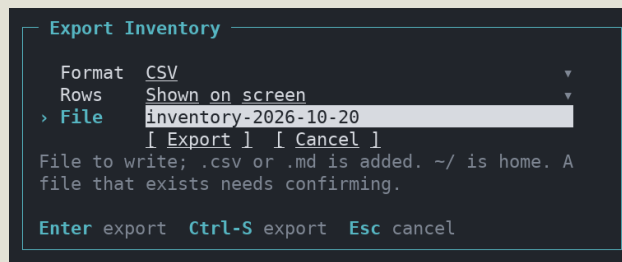


Figure 57. The export form. It opens on the **File** field, with a name already suggested.

15.1 The three questions

Field	What it asks
Format	CSV or Markdown . Enter opens the menu (Figure 58), or press Left and Right to switch without opening it.
Rows	Shown on screen follows the search or filter that is on (and the sort), so only the rows you can scroll through are written. Every row ignores any search or filter and writes the whole table. Both follow the current sort order.
File	Where to write it. See below.

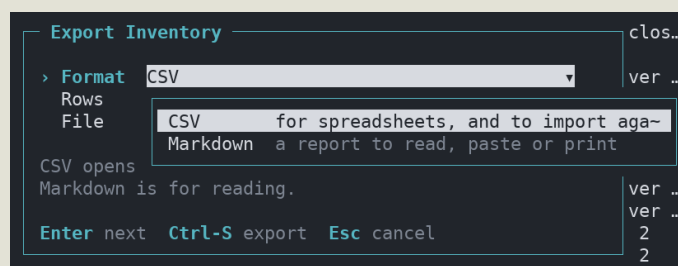


Figure 58. The Format menu.

Press **Ctrl-S** to write the file (**Esc** cancels). The message line then says how many rows were written and the full path of the file, for example Exported 20 rows to /home/you/inventory-report.csv (Figure 59).

Inventory Central v0.1.17a

Tue Oct 20 2026 2:07 PM ! 5 alerts

Site: All HQ Annex
Type: All types Access Point Firewall Laptop Monitor Patch Panel 17 items · sorted by Component Type ▲

Inventory > All sites > All types

Type ▲	Model	Serial	Asset Tag	Status	Room	Maker	Site	Loc. >
> Access Point	U6-Pro	F492BF5...	AP-101	Deployed	Floor 1...	Ubiquiti	HQ	
Access Point	U6-Pro	F492BF5...	AP-102	Deployed	Floor 2...	Ubiquiti	HQ	
Firewall	FortiGa...	FG100FT...	FW-01	Deployed	Server ...	Fortinet	HQ	A1
Laptop	ThinkPa...	LT-0045		Deployed	IT clos...	Lenovo	HQ	
Monitor	U2723QE	CN0K2PH6	MON-0012	Deployed		Dell		
Patch Panel	48-port...		PP-A1	Deployed	Server ...	Leviton	HQ	A1
PDU	AP8941	ZA22010...	PDU-A1	Deployed	Server ...	APC	HQ	A1
Router	EdgeRou...	F09FC2A...	RTR-01	Deployed	Server ...	Ubiquiti	HQ	A1
Server	PowerEd...	7XK2P43	SRV-001	Deployed	Server ...	Dell	HQ	A2
Server	PowerEd...	7XK2P47	SRV-002	Deployed	Server ...	Dell	HQ	A2
Server	ProLian...	CZ22310...	SRV-003	Deployed	Server ...	HPE	HQ	A2
Storage	RS3621x...	2130TWR...	NAS-01	Deployed	Server ...	Synology	HQ	A2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-01	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-02	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	Catalys...	F0C2145...	SW-CORE-01	Deployed	Server ...	Cisco	HQ	A1
Switch	Catalys...	F0C2145...	SW-CORE-02	Deployed	Server ...	Cisco	HQ	A1
UPS	SMX3000...	AS22102...	UPS-01	Deployed	Server ...	APC	HQ	A1

17 items in service - 11 types - 5 places - 3 on Stock
Exported 17 rows to /home/you/inventory-report.csv
a add e edit E cell d del m status {} theme b sidebar w save ? help q quit

Figure 59. The report was written. The message names the file.

The file name

The form suggests a name made of the page and today's date: `inventory-2026-10-20`, `connections-2026-10-20`, `stock-items-2026-10-20`, `cables-2026-10-20`, `locations-2026-10-20`. Change it freely. The rules are:

- The extension is added for you: `.csv` for CSV, `.md` for Markdown. If you type a name that already ends in `.csv`, `.md`, `.markdown` or `.txt`, it is used as typed.
- A name with no folder goes in the folder where you started the program. `~/` at the start means your home folder (`~/reports/switches`). Anywhere else, give the path.
- The folder must already exist. Inventory Central never creates folders for a report; it says The folder `/home/you/reports` does not exist and leaves the form open for you to correct.
- A file that is already there is not overwritten without asking: a box says **That file already exists. Replace it?** and shows the full name. `y` replaces it (atomically, so the old file is whole until the new one is complete), `n` or `Esc` goes back.
- A new file is created private, readable and writable by you only (mode `0600`). Change that with `chmod` if you want to share it.
- A report can never be written over one of the program's own files: the data files, their `.bak` backups, the lock, the change history and the settings file. The form refuses with *That is one of this program's own data files*. Choose another name, however the name is spelled, and also when it reaches one of them through a symbolic link.

15.2 What is in a CSV file

One header line of column names, then one line per row. Every field of the table is written, in the order of the table's columns (for the Inventory: Component Type, Model, Serial Number, Asset Tag, Status, Room, Manufacturer, Site, Location, Position (U), Height (U), Hostname, CPU, Memory, Purchase Date, PO, Project, Notes). The headings are the same names the forms use, so the file can be read back in as it is. Appendix C lists them for each table.

```
Component Type,Model,Serial Number,Asset Tag,Status,Room,Manufacturer,Site,Location,Position (U),Height (U),Hostname,CPU,Memory,Purchase Date,PO,Project,Notes
Access Point,U6-Pro,F492BF5A1101,AP-101,Deployed,Floor 1 office,Ubiquiti,HQ,,,ap-101,,,2024-05-06,P0-24-0201,Wi-Fi refresh 2024,PoE from SW-EDGE-01
Server,PowerEdge R740,7XK2P43,SRV-001,Deployed,Server room,Dell,HQ,A2,10,2,svr-001,2 x Xeon Gold 6230,384 GB DDR4,2022-11-08,P0-22-0612,,Hypervisor host 1
```

The file follows RFC 4180: fields containing commas, quotes or line breaks are wrapped in double quotes, and a quote inside is doubled. Values containing ; or a tab are quoted too (since 0.1.16a), so a spreadsheet that separates with semicolons keeps each in one cell. The lines end with `\n`, and the text is UTF-8 with no byte-order mark. Spreadsheets open it directly; if one guesses the wrong encoding, choose UTF-8 in its import dialog.

Careful. Formulas. A spreadsheet treats a cell that begins with `=`, `+` or `@` (or a `-` that is not the sign of a plain number) as a formula, and a value that someone typed or imported could use that to run something when you open your report. So the program puts an apostrophe in front of any such cell: `=1+1` is written as `'=1+1`, and shows up as text. Importing the file again takes the apostrophe off, so the round trip gives back exactly what you had. Plain negative numbers such as `-5` are left alone.

15.3 What is in a Markdown file

A heading, a line saying when it was made, a few facts about what the report holds, and a table:

```
# Inventory
*Exported 2026-10-20 14:07 by Inventory Central v0.1.17a*
- Rows: 5 of 17 items
- Showing only: any column contains "switch"
- Sorted by: Component Type, ascending
| Type | Model | Serial Number | Asset Tag | Status | Room | Manufacturer | Site | Location | ...
| --- | --- | --- | --- | --- | --- | --- | --- | --- | ...
| Switch | Catalyst 9300-48P | F0C2145X0AB | SW-CORE-01 | Deployed | Server room | Cisco | HQ | A1 | ...
```

The **Rows** line says how many rows are in the report out of how many the table has, so a reader can see at once that it is a part. **Showing only** says which search or filter produced it, in the program's own words (any column contains "switch", or Type contains "switch"); with **Every row** it reads Showing: every row. The table uses the short column names that are on screen (Type rather than Component Type, Pos rather than Position (U)), numbers are right-aligned, and `|`, `<` and `>` inside a value are escaped so that they cannot break the table or be read as markup. A line break inside a value becomes `<br>`.

Tip. For a report to hand to someone, filter the page first (for example, the Status column filtered on `spare`, or a search for `B2`), then press **x**, choose Markdown and keep **Shown on screen**. For a complete backup that can be imported again, choose **CSV** and **Every row**.

15.4 Exporting is read-only

Exporting never changes your records, and it works in - -demo mode too. If the report could not be written (a read-only folder, a full disk, a name that is a folder), the form stays open with the system's reason and you can try another place.

16. Importing CSV Files

Press **i** on any list to read a CSV file into that table: the Inventory (or Stock, or Decom: items are one table, and the Status puts each where it belongs), Connections or its Ports, the supply lines on Stock, or a Config list. The file can be one that Inventory Central exported, or a spreadsheet you saved as CSV: the columns may be in any order, the usual alternative headings are understood, and columns the program has no use for are ignored.

Importing is done in two steps so that nothing surprises you. First the program works out what *would* happen to every row and shows you a **preview**. Only when you press **Enter** on the preview is anything added or changed.

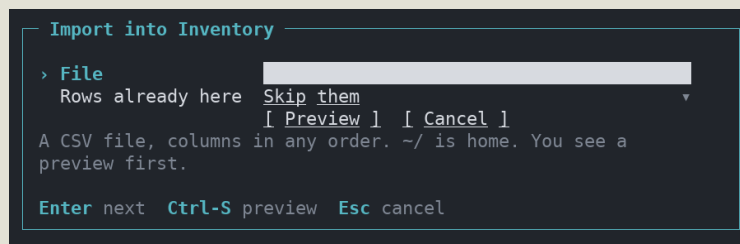


Figure 60. The import form: which file, and what to do with rows that are already in the table.

16.1 Step one: choose the file

Field	What it asks
File	The file to read. ~/ at the start means your home folder; a name with no folder is looked for where you started the program. It must be an ordinary file (not a folder).
Rows already here	Skip them leaves any row that is already in the table exactly as it is. Update them changes it to match the file, but only in the columns the file has, and an empty cell in the file empties that value.

The help line under the second field says what "already here" means for the table you are in (see the next section). Press **Ctrl-S** to read the file and see the preview. If the file cannot be used, the form stays open and says why in plain words: the file does not exist, it is a folder, a quote that was opened is never closed (with its line number), it has no rows, and so on.

16.2 What counts as "already here"

Table	A row in the file is the same as one already there when
Inventory	it has the same Asset Tag ; or, when it has no tag, the same Serial Number and Manufacturer . Upper and lower case do not matter.
Connections	it has the same Cable ID ; or, when it has none, the same two ports (device and port at each end), either way round.
Supplies	it is the same supply line: the same item, part number, type, connector, color and length.

Table	A row in the file is the same as one already there when
Ports	it names the same device and port.
Config lists	it has the same name (a floor or a room within its site, a location within its room, a model within its manufacturer).

These are the same rules that stop the Add form from creating a duplicate. A row that does not match anything is **new**.

16.3 What the program reads

- **Encoding and line ends.** UTF-8 (a byte-order mark at the start is ignored), with any bytes that are not valid UTF-8 read as Windows-1252 (the rest stays UTF-8; since 0.1.16a) and a note in the preview saying so. Lines may end in `n`, `rn` or `r`. A file with NUL bytes is refused as not being text.
- **Separator.** Values are separated by **commas**. A file separated by semicolons or tabs, as some spreadsheets save in some countries, is refused with a message saying so: save it again as "CSV (comma delimited)".
- **Quotes.** A value may be wrapped in double quotes, which is how a value that contains a comma or a line break is written; a double quote inside is doubled (`" "`). A line break inside a value is read as a space.
- **The heading row.** The first row is taken as a heading row when two or more of its names are ones the program knows, or (since 0.1.16a) when one cell is exactly a column's name (Appendix C). Names are compared ignoring capitals, spaces and punctuation, so `Asset Tag`, `asset_tag` and `ASSETTAG` all work, and the alternative names in Appendix C (`Make` for Manufacturer, `SN` for Serial Number, `Qty` for Quantity, and so on) are accepted. If it is not a heading row, the file is taken to have no heading, its columns are taken to be in the order of the Add form for that table, and the preview says so.
- **Columns the program does not know** are ignored, and the preview lists them (`Columns that are not used: Where`).
- **Empty lines** and rows with nothing in them are skipped. A row with more values than the heading has columns is refused with its line number, unless the extra values are empty.
- **Size.** A file may be up to 64 MB and 5000 rows. For more rows, split the file.
- **Spelling of menu values.** `switch` is stored as `Switch`, `fiber mm` as `Fiber MM`, `active` as `Active`: a value that matches one of the program's own choices, apart from capitals, takes the choice's spelling. A value that matches none of the choices is kept as typed (a Status must be one of the five, as in the form).
- **Dates** in the Purchase Date column are accepted in the same forms as the form takes (`2025-03-09`, `2025/3/9`, `2025.03.09`, `today`) and stored as `2025-03-09`.
- **Formula guards.** A leading apostrophe in front of a value that would otherwise look like a formula (`'=A1+1`) is taken off, undoing the protection that exports add (section 15.2).

Every new or changed row then has to pass the same checks as the Add and Edit forms: it needs a tag or a serial number (or a cable ID or a device, or an item), a Purchase Date that is a real date, a Quantity that is a whole number, no duplicate tag, cable ID or port in use, a port that its device has (when the

device has ports on record), a rack unit that is free, and no value longer than its field allows. **An import can never put in something the forms would refuse.**

There is one thing an import does that a form does not: an item names its site, room and location, which must be on the Config lists, and an import of items **adds the places the file names** that are not on the lists yet, for the rows that are imported. The preview lists them as a note (Also adds to the Config lists the places the file names that are not there yet: 1 site, 6 rooms, 4 locations; a location added this way gets its kind from its name, so Shelf 3 is a shelf). A whole inventory from a spreadsheet can therefore come in before any place is entered by hand. Ports must name an item that is already in the Inventory, so import the items first.

Note. A supply line always needs a **Quantity**, as in the form. A file that lists items but has no Quantity column therefore cannot add anything; add a Quantity column (with 0 where you do not know) and import again. The preview explains this on every row, so you will see it at once.

16.4 Step two: read the preview

```

Import into Inventory
new-gear.csv  rows already here: skip them
5 rows in the file: 2 new, 0 changed, 3 left as they are

  1 add      SW-LAB-01
  5 add      FW-02

Enter import  Up/Down scroll  Esc cancel

```

Figure 61. The preview with **Skip them** chosen: two new rows would be added, one row is already there and would be left alone, and one cannot be imported, with the reason.

The top line gives the file name and your choice for rows already here. The second line counts the rows in the file: how many are new, how many would change, how many are left as they were and how many cannot be imported. Below that come any notes about the file, and then one line per row that would do something, numbered by its place in the file (counting the first row below the heading as 1):

Mark	Meaning
add	A new row. Its name (asset tag, cable ID, supply line) follows.
change	A row already here that would be updated. Under it, one line per column that would change, as Model: U6-Pro -> U6-Lite.
PROBLEM	A row that cannot be imported, and the reason, in the words the form would use.

Rows that are already here and would stay as they are (because you chose **Skip them**, or because the file's values are the same as what is stored) are counted but not listed. Use **Up**, **Down**, **PgUp** and **PgDn** to scroll a long list; the preview keeps the first 1000 lines and says how many more there are. Then:

- **Enter** imports everything that is not a problem. Rows marked **PROBLEM** are left out; nothing about them is changed.
- **Esc** cancels. Nothing has been changed at all.

If nothing would be added or changed (every row is already here and left alone, or every row has a problem), the counts say so and the key line reads **Enter** **close**: there is nothing to confirm.

```

Import into Inventory
new-gear.csv  rows already here: update them
5 rows in the file: 2 new, 1 changed, 1 left as they were, 1 cannot be imported

1 add    SW-LAB-01
2 change AP-103
      Hostname: (empty) -> ap-103
4 PROBLEM AP-103 repeats an earlier row of this file
5 add    FW-02

Enter import Up/Down scroll Esc cancel

```

Figure 62. The same kind of preview with **Update them**: the rows already here would change, and each change is listed.

16.5 After the import

The data is saved at once, like every other change. The message line says what happened (Figure 63), for example Import from `inventory.csv`: 2 added, 1 updated, 1 left as they were, 1 not imported. If a search or filter is on, it adds how many rows are shown, since new rows that do not match it are not on screen.

Inventory Central v0.1.17a

Tue Oct 20 2026 2:07 PM ! 4 alerts

Site All HQ Annex v V site
Type All types Access Point Firewall Laptop Monitor Patch Panel [] type
Inventory > All sites > All types 19 items · sorted by Component Type ▲

Type ▲	Model	Serial	Asset Tag	Status	Room	Maker	Site	Loc. >
Access Point	U6-Pro	F492BF5...	AP-101	Deployed	Floor 1...	Ubiquiti	HQ	
Access Point	U6-Pro	F492BF5...	AP-102	Deployed	Floor 2...	Ubiquiti	HQ	
Firewall	FortiGa...	FG100FT...	FW-01	Deployed	Server ...	Fortinet	HQ	A1
Firewall	FortiGa...		FW-02	Deployed		Fortinet		
Laptop	ThinkPa...		LT-0045	Deployed	IT clos...	Lenovo	HQ	
Monitor	U2723QE	CN0K2PH6	MON-0012	Deployed		Dell		
Patch Panel	48-port...		PP-A1	Deployed	Server ...	Leviton	HQ	A1
PDU	AP8941	ZA22010...	PDU-A1	Deployed	Server ...	APC	HQ	A1
Router	EdgeRou...	F09FC2A...	RTR-01	Deployed	Server ...	Ubiquiti	HQ	A1
Server	PowerEd...	7XK2P43	SRV-001	Deployed	Server ...	Dell	HQ	A2
Server	PowerEd...	7XK2P47	SRV-002	Deployed	Server ...	Dell	HQ	A2
Server	ProLian...	CZ22310...	SRV-003	Deployed	Server ...	HPE	HQ	A2
Storage	RS3621x...	2130TWR...	NAS-01	Deployed	Server ...	Synology	HQ	A2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-01	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	2930F 4...	SG98KQ1...	SW-EDGE-02	Deployed	IDF 2	HPE Aru...	HQ	B2
Switch	C9300-2...		SW-LAB-01	Deployed		Cisco		
Switch	Catalys...	FOC2145...	SW-CORE-01	Deployed	Server ...	Cisco	HQ	A1
Switch	Catalys...	FOC2145...	SW-CORE-02	Deployed	Server ...	Cisco	HQ	A1
UPS	SMX3000...	AS22102...	UPS-01	Deployed	Server ...	APC	HQ	A1

19 items in service - 11 types - 5 places - 3 on Stock
Import from new-gear.csv: 2 added, 1 updated, 1 left as they were, 1 not imported
a add e edit E cell d del m status {} theme b sidebar w save ? help q quit

Figure 63. The message after an import.

Careful. There is no undo for an import. Before a large one, press **x** and export **Every row** to a CSV file, or copy the data files (Chapter 17). The automatic backup (`inventory.csv.bak`) holds the file as it was at the first save of the session, which may be earlier than you expect.

16.6 Moving data in from other sources

- **From a spreadsheet.** Make a heading row with names from Appendix C, one row per item, and save as CSV with UTF-8 and commas.
- **From another copy of Inventory Central.** Export **Every row** as CSV from the one, import that file with **Skip them** into the other. Rows the second copy already has are left alone.
- **To bring a column of changes in.** Put the **Asset Tag** (or Cable ID) first and only the columns you want to change after it; import with **Update them**. A bulk change of Project for a batch of items is a two-column file.
- **Ports for cables you already have.** Import the connections, then press **P** on Connections to make the ports they name (section 6.4).
- **From the earlier C version.** Its data file, `inventory.csv` in your home folder, is plain CSV in the same column order as the first nine columns of this version's. If the standard data folder holds no data yet, the program finds that file by itself and says so on the first screen (section 17.1); or point -f at it, or import it into an empty table with **i**.

17. Your Data: Files, Saving and Backups

Inventory Central keeps everything in ordinary text files that you own. There is no database, no hidden format and no account. This chapter says where the files are, exactly when and how they are written, and what protects them.

17.1 The data files

Each table and each Config list has a CSV file of its own. They live side by side, and the others are named after the first:

inventory.csv	the items (Inventory, Stock and Decom)
inventory-connections.csv	the Connections
inventory-supplies.csv	the supply lines of Stock
inventory-ports.csv	the Ports
inventory-sites.csv	the Config lists: sites,
inventory-floors.csv	floors (since 0.1.17a),
inventory-rooms.csv	rooms,
inventory-racks.csv	locations (racks, shelves, desks, bins, closets),
inventory-manufacturers.csv	manufacturers,
inventory-models.csv	models,
inventory-types.csv	types,
inventory-cpus.csv	CPUs,
inventory-memory.csv	memory
inventory-porttypes.csv	and port types
inventory.journal	the change history (Chapter {n:history})

Where the first one is, is decided in this order:

1. The file named with `-f FILE` (or `--file`) on the command line, or given as the one operand (`inventory ~/it/kit.csv`).
2. The file named by the environment variable `INVENTORY_FILE`.
3. A file called `inventory.csv` in your **home folder**, left there by the earlier C version of the program, **if** the standard folder (next) holds no data yet. The program says so on the first screen: `Using /home/you/inventory.csv` from an earlier version; the other files are created beside it.
4. The standard place: `$XDG_DATA_HOME/inventory-central/inventory.csv`, which is `~/.local/share/inventory-central/inventory.csv` unless `XDG_DATA_HOME` is set to an absolute path.

The other files take the first file's name without its `.csv`, plus `-connections.csv`, `-supplies.csv`, `-ports.csv`, `-sites.csv` and so on. A file given as `-f ~/ic-practice.csv` therefore has its companions `~/ic-practice-connections.csv`, `~/ic-practice-sites.csv` and the rest. A name that does not end in `.csv` gets the suffix and `.csv` added: `-f kit` uses `kit` for the inventory, `kit-connections.csv`, `kit-supplies.csv` and so on. The Settings page lists the files in use. Settings are kept apart (section 14.8).

The standard folder is created, private to you (mode `0700`), the first time it is needed. A folder you name yourself is never created; the program says that it does not exist and does not start.

A table whose file does not exist is simply empty, and the file appears at the first change. You can keep several separate inventories by using several `-f` names.

Tip. Use a shell alias or a small wrapper for each inventory you keep: `alias lab='inventory -f ~/it/lab.csv'`. Each name has its own files and its own lock.

17.2 What is inside the files

Each file is UTF-8 text in the common CSV form (RFC 4180): a header row, then one row per record. The header names are the field names of the forms, in the order of the forms (Appendix C). A value is wrapped in double quotes only when it has to be: when it contains a comma, a double quote, a line break, a `;` or a tab (since 0.1.16a, so that a spreadsheet set to separate with semicolons keeps it in one cell), or starts or ends with a space. Rows are written in the order they were entered; the order you see on screen is only a view and is not stored.

```
Serial Number,Component Type,Asset Tag,Manufacturer,Model,Notes,Hostname,Purchase Date,P0,Status,Site,Room,Location,...,Project
FOC2145X0AB,Switch,SW-CORE-01,Cisco,Catalyst 9300-48P,"Core switch, stacked with SW-CORE-02",sw-core-01,2023-03-14,P0-23-0117,Deployed,HQ,Server room,A1,...,
```

The inventory file of the earlier versions had nine columns; Status came at the end in 0.1.1a, and Site, Room, Rack, Position, Height, CPU and Memory after it in 0.1.2a. An older file is read with the new columns empty and gets them at the next save. The models file gained an Outlets column at its end in 0.1.4a. In 0.1.13a the inventory's **Location** column became **Hostname**, and **Project** came at the end: a file with a Location column is read with each location added to the item's Notes as `Location: ...` (the program says how many), so nothing is lost. In 0.1.17a the items' **Rack** column is headed **Location** (a file with a **Rack** heading is read into it), the locations file gained a **Kind** column, the rooms file a **Floor**, and the supplies file a **Class**, each at the end, so older files load as they are; and the floors are a new file.

Formulas (since 0.1.16a). A value that a spreadsheet would read as a formula (starting with `=`, `+`, `@`, or `-` not followed by a plain number) is saved with a leading apostrophe, which the program removes again on reading, so opening a data file in a spreadsheet cannot run something that someone typed. `=1+1` is saved as `'=1+1` and read back as `=1+1`; `-5` is left alone. Exports have always done the same (section 15.2).

Because it is plain CSV, you can open the files in a spreadsheet, search them with `grep`, compare them with `diff`, keep them under version control and back them up with whatever you already use. Exports (Chapter 15) differ from the data files only in column order and in being a copy for someone else to read.

Editing the files by hand

You may edit the files in a text editor or a spreadsheet **while the program is not running**. The program reads them with the same rules as an import (section 16.3): the heading names may be the alternative ones (`Make`, `SN`, `Qty`), the columns may be in any order, and a file saved in the Windows-1252 encoding is read correctly and saved again as UTF-8, with a note on the Checks page.

Two things are kept for you:

- **Columns the program does not know** are kept. If you add a column `Warranty` to the inventory file, it is not shown, but it is carried along unchanged and written back at every save, with each row's own value.

- **Values are never cut off.** A value longer than its field lets you type is kept exactly as it was read from the file, not shortened (an *import* does refuse such a value, section 16.3; the data files are trusted as they are). Since 0.1.16a such a record cannot be saved again until the value is shortened: the form says which field is too long (*Notes is longer than 1000 characters (1204)*).

The Checks page (Chapter 12) also reports problems that a hand edit can introduce: duplicate tags, a quantity that is not a number, a date that is not a date, a status that is not one of the choices.

A damaged file is never overwritten. If a file cannot be read (a quoted value that is never closed, a stray NUL byte, a file that is a folder, a file over 64 MB), the program refuses to start and names the file and the line:

```
inventory: d/inv.csv: line 2: a quoted value is never closed (a " was opened and has no matching ")
```

It leaves the file as it is. Repair it in an editor (the `.bak` file beside it is the copy from the start of your last session), and start again.

17.3 When it is saved

Every change is saved the moment you make it: when a form is saved, a row is deleted, a stock level is changed with `+` or `-`, or an import is confirmed. There is no unsaved state to lose, and no save command is needed. (`w` saves again, which only matters after a failed save, below.) Only the tables that changed are written; the other files are not touched. (Renaming a site writes the sites, and the floors, rooms, locations and items that follow it.)

Changing a **setting** writes only the settings file. Exporting a report writes only the report.

How a file is written

A save never rewrites the file in place. The program:

1. writes the complete new text to a temporary file beside the real one (named like `inventory.csv.12345.tmp`);
2. checks first that the text reads back as exactly the data in memory, and refuses to write at all if it does not;
3. flushes the temporary file to disk;
4. renames it over the real file, which is a single, atomic step; and
5. flushes the folder.

A crash, a power cut or a full disk at any point leaves either the whole old file or the whole new one, never half of each.

The files one action changes are saved together (since 0.1.16a). Renaming a site, say, changes the sites, floors, rooms, locations and items. The new versions are written first, then a short list `NAME.saving` (`inventory.saving`) naming them, then the files are replaced and the list removed. If the program stops in between, the save is finished the next time the files are opened, and the Checks page says so. So the files never disagree, with the items renamed and the locations still naming the old site. A file keeps its permissions, and a new one is created private (`0600`). If a data file is a **symbolic link**, the file it points to is updated and the link stays a link (which is how you can keep the data in a synchronized folder).

The automatic backup

The first time in a session that a table is saved, the program first copies the file as it stood to a `.bak` file beside it: `inventory.csv.bak`, `inventory-connections.csv.bak`, and so on for each file. Later saves in the same session do not touch it again, so the backup is **the state at the start of that session's first change to that table**. It is replaced the next time you run the program and change something. A backup that cannot be written stops the save, and says so.

Careful. The `.bak` file is a safety net against a mistake made in this session, not an archive. Back the files up yourself with your regular backup, and before something big (a large import, a bulk edit by hand) copy them first: `cp inventory.csv inventory.csv.before-import`.

When a save fails

If the disk is full, the folder has become read-only or the file system has gone away, the save cannot be completed. The program says so on the message line and keeps a red banner at the bottom of every page:

```
NOT SAVED (/home/you/.local/share/inventory-central/inventory.csv: No space left on device). Changes are in memory only; press w to try again.
```

Nothing is lost yet: the changes are in memory and the file on disk is still the last good one. Make room (or fix the folder), then press `w`. If you try to quit while changes are only in memory, a box warns you (The last save failed, so your changes are only in memory. Quit anyway and lose them?).

17.4 One copy at a time: the lock

Two copies of the program writing to the same files would overwrite each other's changes. To prevent that, a copy that starts creates a small lock file beside the first data file, named after it with `.lock` added (`inventory.csv.lock`), holding its process number. A second copy that finds a live lock refuses to start:

```
inventory: another copy of Inventory Central (process 4242) has these files open. Close it first, or if it is not running, delete /home/you/.local/share/inventory-central/inventory.csv.lock
```

The lock is released when the program quits, also after an internal error. If the program is killed (`kill`, or `kill -9`), a small helper copy of the program that watches over it removes the locks and puts your terminal back in order. If the terminal *window* is closed, the whole group is hung up, helper included, and the lock may stay behind; it does no harm, because the next start sees that its process is gone and replaces it. Details of how it works:

- A lock whose process **is no longer running** (the program was killed) is recognized as stale and taken over automatically, without asking. The program asks the system whether the process exists with `kill -s 0`, and with `ps` when it is not allowed to ask about another user's process. A lock that belongs to a **running** process is never broken.
- If two copies start at the very same moment, exactly one gets the lock.
- The lock is tied to the real location of the data. Two names that reach the same file through a symbolic link share one lock.
- A lock file that does not contain a process number is reported (`... is not a valid lock file. If no other copy is running, delete it`) and not guessed at.
- `--dump` only reads, so it works while another copy is running; it does not take the lock.

- **The lock says where its copy runs** (since 0.1.16a). It records the computer, its boot, the process-number space and the start time of the process. A lock made on another computer or in another container is never taken over: the program says so and names the lock; delete it only if you are sure no copy is running there. A lock made before this computer last started (after a power cut, say) is taken over, since nothing from before the restart can still be running. A process number that was given again to another process is recognized by its different start time. A lock written by 0.1.15a or earlier holds only the process number and is judged as before.
- **The program also locks** `NAME.journal` (`NAME.journal.lock`, since 0.1.16a), so it never runs on a folder an `inventoryd` server is using (the server takes the same lock), and it refuses a folder whose change history is a server's (Chapter 18). A dump still reads it.
- **Two computers at once.** The lock can tell that another computer made it, not whether the copy there still runs. Do not run the program on the same files from two machines at once.

17.5 Who can read the files

Files the program creates (data, backups, lock, reports, crash log, settings) are readable and writable by you only (0600); the folders it creates are 0700. Files that already exist keep the permissions you gave them. Inventory records can include serial numbers and rack locations that you may not want to share, so check the permissions if you put the files in a shared folder.

17.6 Demo mode

`inventory --demo` loads a small example office and reads and writes **no** files at all, not even the settings. It takes no lock either, so it can run beside a real session. The Settings page shows (`none: demo data is not saved`). Combining `--demo` with a file name is an error: the program refuses rather than leave you unsure whether your own data is being shown.

17.7 The crash log

If the program ever hits an internal error, it does not lose your data (everything was saved after your last change), and it tries to carry on. It writes what it knows to `inventory-central-crash.log` in the data folder (or, if there is none, as in demo mode, to `inventory-central-crash-NNNN.log` in the system's temporary folder, where NNNN is the process number), and tells you where on the message line. The file is private to you, and the program never writes through a symbolic link to make it. After three such errors in one session it stops. Please keep the file: it is the most useful thing you can send with a report of the problem.

18. Sharing the Data: the Server

Everything in the earlier chapters works on files that one person uses at a time. Inventory Central also comes with a server, `inventoryd`, which holds the data so that several terminal programs can work on it at once, each seeing the others' changes. The account that runs the server works on its data directly (`inventory -S`); other people reach it from their own computers through SSH (`inventory -R`), each under their own name and with their own role: viewer, editor or admin. Since 0.1.10a, `inventoryd setup` makes it a service that starts at boot and `inventoryd teardown` undoes that, saving the data as the terminal program's files first (sections 18.14 and 18.15). Since 0.1.11a there is also a **web page** for the data in a browser, behind nginx (Chapter 19). For setting all of this up step by step, read the **Server Setup Guide**, a separate PDF beside this manual: it explains what to do and why, what usually goes wrong, and how to take it down again.

Note. The server does nothing until you start it. Installing Inventory Central does not run it, and it installs nothing itself: it opens no network port (only a socket file inside its own folder), makes no service, creates no account and does not put itself in the background. Only an administrator running `inventoryd setup` as root changes that. If you never do, Inventory Central is a terminal program and nothing more.

18.1 What changes when you use it

- **The server holds the data**, in one file: `inventory.journal` in the server's folder. It is the change history of Chapter 13 in the server's form (IC2 lines, which also carry each row's number and version), and it is the data itself: when the server starts, it reads the file from the beginning and has every table as it stands.
- **The terminal program keeps no files** of its own while it is connected: no CSV files, no backup, no lock. Every change goes to the server as you make it.
- **Two people can work at once**. Each program follows the server about once a second, so a change one person makes appears on the other's screen without a key being pressed.
- **Nothing is overwritten unseen**. Every row has a version. A change says which version it was made to; if someone else changed or deleted the row in between, the server refuses the change, and your program shows the row as it now is (section 18.4).

18.2 Starting a server by hand

First, make a folder for it, private to you. The server refuses a folder that other people can reach.

```
mkdir ~/ic-server  
chmod 700 ~/ic-server
```

Then fill it from your CSV files, once. `-i` reads the inventory file and the files beside it (exactly the files the terminal program uses, Chapter 17) into the empty folder as one transaction, then reads them back and compares:

```
$ inventoryd -d ~/ic-server -i ~/.local/share/inventory-central/inventory.csv
inventoryd: imported 312 rows from /home/you/.local/share/inventory-central/inventory.csv
and the files beside it
```

The files themselves are not changed or moved. An import into a folder that already holds data is refused, and so are files with extra columns this version does not know, because their values would be lost.

Start it. It runs in the foreground until you stop it (`Ctrl-C` in its window, or `kill`). Run it in a window of its own, or under `nohup`, `tmux` or `screen`:

```
$ inventoryd -d ~/ic-server
inventoryd: serving /home/you/ic-server (312 rows, 312 journal lines)
on /home/you/ic-server/inventoryd.sock
```

Connect with `-S` and the socket's name, from as many terminal windows as you like:

```
inventory -S ~/ic-server/inventoryd.sock
```

The message line says `Connected to inventoryd v0.1.8a at ...: every change goes to the server`, and the Settings page shows the server in place of the data files.

18.3 Working with the server

Everything works as before: the pages, the forms, the keys, the mouse, the Checks, export. History (`6`) shows the server's history: everyone's changes, with **Who** taken from each person's login name, and a **Connected** row each time a program connects. `w` says `Every change goes to the server as you make it`; there is nothing to save.

18.4 When two people change the same row

Suppose you and a colleague both open the same item. Your colleague saves first. When you save, the server refuses your change, because the item is no longer the version you started from:

```
Not saved: item SW-CORE-01 was changed by someone else while you had it open (Notes now as they left it).
Your changes are kept here; check them and press Ctrl-S to save.
```

Nothing of yours was applied, not even the parts that did not clash: one action is all or nothing. Since 0.1.16a nothing you typed is lost either. The form opens again on the item as your colleague left it, with the fields you changed set to your values, and the message names the fields the other change touched (and any field you both changed, where yours is shown). Look at it (the History page shows exactly what changed), and press `Ctrl-S` again to save your version on top of theirs, or `Esc` to leave theirs. If your colleague deleted the item meanwhile, the form stays open as `Add item (deleted meanwhile)` with your values: `Ctrl-S` adds it again, `Esc` drops it.

A change made without a form (a stock level with `+`, a delete) is refused the same way when the row changed first, and the screen shows the row as it now is: `Not saved: ... was changed or deleted by someone else first.`

18.5 When the server cannot be reached

If the server stops while you are working, the next change says `Not saved: lost the connection to the server ...`, and the screen goes back to what the server last had. The program waits at most 10 seconds for any answer (the server is not answering (nothing for 10 s)), so a server that hangs never freezes the screen: `q` always quits. Since 0.1.16a it connects again by itself in the background, after 1 second, then 2, 4 and so on up to 30 seconds between tries, and says so on the message line (`... Trying again in 4 s; changes cannot be saved until then., then Connected again to ...`). When the server is running again, carry on. Over SSH these tries never ask anything (`ssh -o BatchMode=yes`): a key that needs a passphrase must be in an `ssh-agent`, or the message shows ssh's own reason (`Permission denied (publickey)`); only the first connection, before the screen starts, may ask. If the server is not running when you start the program, it stops with `cannot reach the server at ... (is inventoryd running, and is this its socket?)` and status 1.

18.6 When the server cannot record changes

If the server cannot write its journal (a full disk, say), it refuses every change from then on, with the reason, and still answers everyone's reading. Fix the cause and restart it. Before the server is told about any change, the journal has it: a change that was answered is never lost, even if the server is killed (this is tested by killing it repeatedly while a client writes).

18.7 Going back to files

`-e` writes the server's data as the terminal program's ordinary CSV files, which `inventory -f` then opens like any others. It reads the journal without changing it, so it can run while the server runs, and it never writes over files that are already there:

```
$ inventoryd -d ~/ic-server -e ~/it/inventory.csv
inventoryd: wrote 14 files: /home/you/it/inventory.csv and the files beside it
$ inventory -f ~/it/inventory.csv
```

The files are exactly what the terminal program would have written for the same data. Keep the server's folder as well: its journal is the complete history. `inventoryd teardown` (section 18.15) does this for you and also removes everything setup made.

18.8 Backing up the server

Copy `inventory.journal`. The file only grows at its end, so a copy taken while the server runs is a good backup: an unfinished last transaction in the copy is cut off when a server starts on it. To restore, stop the server, put the copy back, start it. A journal damaged anywhere but at its very end stops the server from starting (`will not start: ... is damaged at line ...`), and the file is left as it is.

18.9 People on other computers: SSH

Other people reach the server through SSH, which the computer already has and which already checks who they are. They never get a shell: their key is tied to a small program, `inventoryd-session`, which tells the server who they are and then passes the conversation through. The person cannot change that name, whatever they type.

On the server's computer, in the account that runs `inventoryd`, give each person's public key a line of its own in `~/.ssh/authorized_keys`:

```
restrict,command="/usr/local/bin/inventoryd-session -s /home/inv/ic-server/inventoryd-session.sock -u alice" ssh-ed25519 AAAA... alice@laptop
```

`restrict` turns off everything else SSH could do (a terminal, forwarding); `command=` runs the helper whatever the person asked for; `-u alice` is the name the server records for every change made with that key. Then say what alice may do, in the server folder's **users** file (section 18.10).

On the person's computer, they connect with `-R` and the account and host:

```
inventory -R inv@server.example.org
```

ssh asks for the key's passphrase if it needs one (an `ssh-agent` saves asking again). The message line says `Connected to inventoryd v0.1.16a at inv@server.example.org (SSH) as alice: every change goes to the server, and from there everything works as in section 18.3`. To use a different ssh command or wrapper, set `INVENTORY_SSH` to it; it is run with `-T -e none -- DESTINATION`.

18.10 Who may do what: roles

There are three roles. A **viewer** sees everything and may export. An **editor** may also change data. An **admin** may also see the audit events on the History page (who connected, who was refused, who exported) and is meant to look after the server; an admin is **not** an editor unless given both roles. A viewer's program says `you can look, not change` when it connects, and the keys that would change something answer `You can look but not change anything here (your roles: viewer)`.

The roles come from two files in the server's folder, read again at every new session (so a change applies at the next connection, without a restart). Neither may be writable by group or others; the server will not start otherwise. Setup makes them root's own, mode 644: the server only reads them, and its own account cannot change who comes in.

`users`: one person per line, `NAME ROLES`, and `disabled` to shut someone out without removing the line:

```
# name  roles
alice   editor
bob     viewer
carol   editor,admin
dave    editor      disabled
```

roles: one role per line, `ROLE GROUP . . .`. A person who is not in the users file gets each role one of whose groups they are in, as the computer reports it (`id -Gn NAME`):

```
viewer    inventory-viewers staff
editor    inventory-editors
admin     inventory-admins
```

A person in neither, or disabled, is refused at the door: their program stops with the `server refused: not let in: . . . is not a registered user (or is disabled)`, and the refusal is recorded. So are attempts to change data without the editor role.

18.11 Directory groups (LDAP, Active Directory) through sssd

Inventory Central has no LDAP code of its own. On a computer joined to a directory with `sssd`, `id -Gn NAME` already answers from the directory, so the roles file can name directory groups, and taking someone out of `inventory-editors` in the directory takes their editing away at their next session.

With `sssd`, people usually log in to the server as **themselves**, with keys the directory holds. Then the server's `sshd_config` sends members of a group to the helper, which, without `-u`, names the account it runs as:

```
Match Group inventory-users
    ForceCommand /usr/local/bin/inventoryd-session -s /srv/inventory/inventoryd-session.sock
    DisableForwarding yes
    PermitTunnel no
    AllowTcpForwarding no
    X11Forwarding no
    PermitTTY no
AuthorizedKeysCommand /usr/bin/sss_ssh_authorizedkeys
AuthorizedKeysCommandUser nobody
```

The helper then runs as each person, so the session socket must be open to their group: start the server with `-G`, after giving the server's folder to that group (`chgrp inventory-users /srv/inventory; chmod 750 /srv/inventory`). The session socket then has mode 0660 and the folder's group; the owner's socket stays private.

Careful. With `-G`, the name is only as safe as the group. Anyone in that group who can run programs on the server's computer could connect to the session socket without the helper and claim another name. `ForceCommand` alone does not stop that over SSH: it applies only to sessions, and `ssh -N -L` can still forward a socket of the person's own to the session socket. `DisableForwarding yes` (all forwarding, Unix sockets included; `AllowTcpForwarding no` covers only TCP) and `PermitTunnel no` close that, so keep them in the block. Make sure members of the group have no other way to run programs there (no local logins, no other shells). The `authorized_keys` layout, where the helper runs as the server's own account and the socket stays private (without `-G`), does not have this weakness. With a separate SSH account for the helper (not each person's own), give both the server and the helper `-k` with that account's and the server account's copies of one key (since 0.1.16a): then a group member who connects directly cannot name anyone. When the helper runs as each person (`ForceCommand`), no key can be kept from them; the server prints a note when started with `-G` and no `-k`.

18.12 Idle sessions

An SSH session that sends nothing for 15 minutes is closed by the server; for a person with the admin role, after 10 minutes (as the application security rules ask). The program notices and connects again by itself, through SSH again, without asking for a passphrase (section 18.5); a change made while it is away is not saved (`Not saved: lost the connection ...`). `-T` changes the limits, each at most 86400 seconds (a day). The owner's own connection (`-S`) is closed after an hour with no request (the program asks every second while it runs, so this only ends connections left behind).

Since 0.1.16a every connection must say `HELLO` within 30 seconds, and each line must arrive within 60 seconds of its first byte. At most 16 owner, 64 SSH and 64 web connections are served at once; one more is told the server is busy (`the server is busy (too many connections); try again in a moment`) and closed.

18.13 The server checks the data too

Since 0.1.10a the server holds every change to the same field rules as the terminal program's forms: required fields, lists (a type, a model or a place must be one that exists), numbers and dates, unique tags and serial numbers, rack units that are free, ports that are not taken twice. It judges the data as it would be after the whole change, so a rename that follows through to other tables goes through as one. A change that breaks a rule is refused with the rule's message (`Not saved: item NEW-1: Position (U): U40 is already filled by SW-CORE-01 (U40)`), and nothing of it is applied.

A row that already broke a rule before the change (data imported with problems, say) is not held to it: only **new** problems are refused, so you can still work on such rows and fix them. The Checks page (Chapter 12) lists what is wrong.

18.14 Setting it up as a service

On a computer that will serve the inventory to others, an administrator runs, as root:

```
inventoryd setup -n                # first: show every step, change nothing
inventoryd setup -i /path/to/inventory.csv
```

`inventoryd setup` runs `inventory-setup`, a shell script installed beside the programs (you can also run it directly). It:

1. checks that `inventoryd` and `inventoryd-session` are installed (in `/usr/local/bin`, or give `-b`);
2. makes the service account `inventory` (`-a` for another name), on Linux with `useradd --system`; on macOS it asks you to make the account first and shows the command;
3. makes the data folder `/var/lib/inventory-central` (`-d` for another; on macOS `/usr/local/var/inventory-central`), private, which is also the account's home;
4. with `-i`, imports your CSV files into it (`inventoryd -i`, which reads the result back and compares);
5. writes the `users` and `roles` files and `.ssh/authorized_keys`, each with examples and nobody let in yet;
6. writes and starts the service: a `systemd` unit on Linux (locked down: its own account, no network at all, the data folder the only place it may write), or a `launchd` job on macOS;
7. lists what is left for you: each person's key line and role, and how they connect.

Everything it makes is listed in a manifest of root's own outside the data folder (in `/var/lib/inventory-central-setup/`, on macOS `/var/db/inventory-central-setup/`), so teardown knows exactly what to undo; a `setup.manifest` left in the data folder by an older version is moved there. The `users`, `roles` and SSH key files are root's own, so the service account cannot change who comes in. Running it again stops the service while it works, adds only what is missing, follows no symbolic link in the data folder, and keeps files you have edited, except the unit (or launchd job) and the nginx site setup made itself: those it rewrites to its own template when they differ.

With `-g GROUP`, for people who log in as themselves (sssd, section 18.11), the folder and session socket are opened to that group (`inventoryd -G`), and setup prints the `Match Group` lines to add to `sshd_config`. It does not edit `sshd_config` itself.

Note. `-r ROOT` writes the files under another folder and only shows the system commands: for checking the result before you run it for real, or for building a package.

18.15 Turning it back into files: teardown

```
inventoryd teardown -n -d /var/lib/inventory-central -o /root/inventory.csv
inventoryd teardown -d /var/lib/inventory-central -o /root/inventory.csv
```

`inventoryd teardown` runs `inventory-teardown`, which stands alone: it needs `inventoryd` only to save the data. In this order, it:

1. stops the service, so the data stops changing;
2. saves the data as `/root/inventory.csv` and the files beside it, then **proves the copy**: it imports it into a scratch folder, exports that again and compares every byte. If anything fails (the export, the check, a file already there), it starts the service again and **removes nothing**;
3. copies the server's journal beside them (`inventory.server-journal`), so the change history is kept;
4. removes the service and the sockets, and moves the `users`, `roles` and key files into a `retired-` folder inside the data folder, so nobody can come in through SSH any more.

If setup made the web page, teardown first removes its nginx site (and its link), checks nginx's configuration, reloads nginx and takes nginx's account out of the server's group; the certificates are kept. It never deletes data: the data folder and its journal stay until you delete them. The service account stays unless you add `-A`. It lists what it cannot undo itself: lines you added to `sshd_config`, groups in your directory, the programs, a CA. Afterwards `inventory -f /root/inventory.csv` opens the same data as a terminal program. The saved files are root's own (mode 600; `chown` them for the person who will use them), and teardown refuses an output folder that others could write to (unless it is root's own and sticky, like `/tmp`).

18.16 Certificates for the web page

The web page lets people in by client certificate. `inventory-ca`, also installed beside the programs, keeps the certificate authority for that: `init`, `server`, `issue` (a .p12 for a person's browser), `revoke`, `crl`, `publish` (puts the CA certificate and the revocation list into the web server's folder), `check`, `list`. `CA-SETUP.md`, shipped with the programs, walks through it, and Chapter 19 sets the page up. SSH does not need it.

18.17 The server's options

```
inventoryd -d dir [-G] [-k keyfile] [-T seconds[,seconds]] [-W socket -K keyfile]
inventoryd -d dir -i file
inventoryd -d dir -e file
inventoryd setup [option...]
inventoryd teardown [option...]
inventoryd -h | -V
```

Option	Long form	What it does
-d DIR	--dir DIR	The server's folder. It must exist and be private to you (<code>chmod 700</code> ; with <code>-G</code> , <code>750</code>).
-i FILE	--import FILE	Fill an empty folder from <code>FILE</code> and the files beside it, then exit.
-e FILE	--export FILE	Write the data as <code>FILE</code> and the files beside it, then exit.
-G	--group	Open the SSH session socket to the folder's group (mode <code>0660</code>), for the <code>sssd</code> layout (section 18.11).
-k FILE	--session-key FILE	The SSH helper must send the key in <code>FILE</code> (<code>inventoryd-session -k</code>), since 0.1.16a: a file of this account's only, one line of 64 lowercase hex digits. See section 18.11.
-T S[,S]	--idle S[,S]	Close SSH sessions idle this many seconds: people, then admins (default <code>900,600</code> ; each at most <code>86400</code>).
-W SOCKET	--web SOCKET	Also answer the web page on <code>SOCKET</code> , an absolute path, for <code>nginx</code> (mode <code>0660</code> , the account's group; its folder must not be writable by everyone). Needs <code>-K</code> . See Chapter 19.
-K FILE	--web-key FILE	With <code>-W</code> , needed (since 0.1.16a): the key <code>nginx</code> sends with every request (<code>X-IC-Proxy-Key</code>); same file rules as <code>-k</code> . Setup makes it.
-h	--help	Print the usage text.
-V	--version	Print the version.

Exit status: 0 done, 1 an error (the message says which), 2 a mistake in the command line: also `-W` without `-K`, `-K` without `-W`, and `-k` or `-W` with `-i` or `-e`. A key file must be a plain file (not a symbolic link) owned by the account the server runs as, with no permissions for group or others; otherwise the server does not start and says why. The files in the folder: `inventory.journal` (the data), `inventory.journal.lock` (one server per folder), `inventoryd.sock` (the owner's socket, mode `0600`), `inventoryd-session.sock` (the SSH helper's socket, `0600`, or `0660` with `-G`), and the `users` and `roles` files you write.

The helper's options: `inventoryd-session -s SOCKET [-u NAME] [-k KEYFILE]`, and `-h`, `-V`. With `-k` (`--key`, since 0.1.16a) it sends the key in `KEYFILE`, which must be readable by its own account only. It ignores whatever command the person asked SSH to run.

18.18 What the server does not do yet

- **Closing the web page from the page itself.** An admin closes and opens it from the terminal program's Settings page (section 14.7).
- **Editing `sshd_config`.** Setup prints the lines for the group layout; adding them is up to you, as is removing them.
- **Asking the system who is on a socket.** Rust's standard library cannot ask who is at the other end of a Unix socket, so the server trusts its sockets' permissions, the session key (`-k`) and, in the `sssd` layout, `sshd`'s settings (`ForceCommand`, `DisableForwarding`, no other shells; section 18.11).
- **macOS has not been run.** The `launchd` job and the macOS steps are written to Apple's documentation but tested only as files; Linux is tested.

19. The Web Page

For the people who use the page, the **Web Page User Guide** (a separate PDF beside this manual, since 0.1.15a) explains it step by step, from installing a certificate to printing. Since 0.1.11a the server can show its data in a browser: the racks drawn as they stand, the equipment, the cabling, the supplies, the checks and the history, on a desk or on a phone. Since 0.1.12a, people with the editor role can also add, change and delete records there, held to the same rules as the terminal program's forms. Changes made anywhere appear on the page within a few seconds.

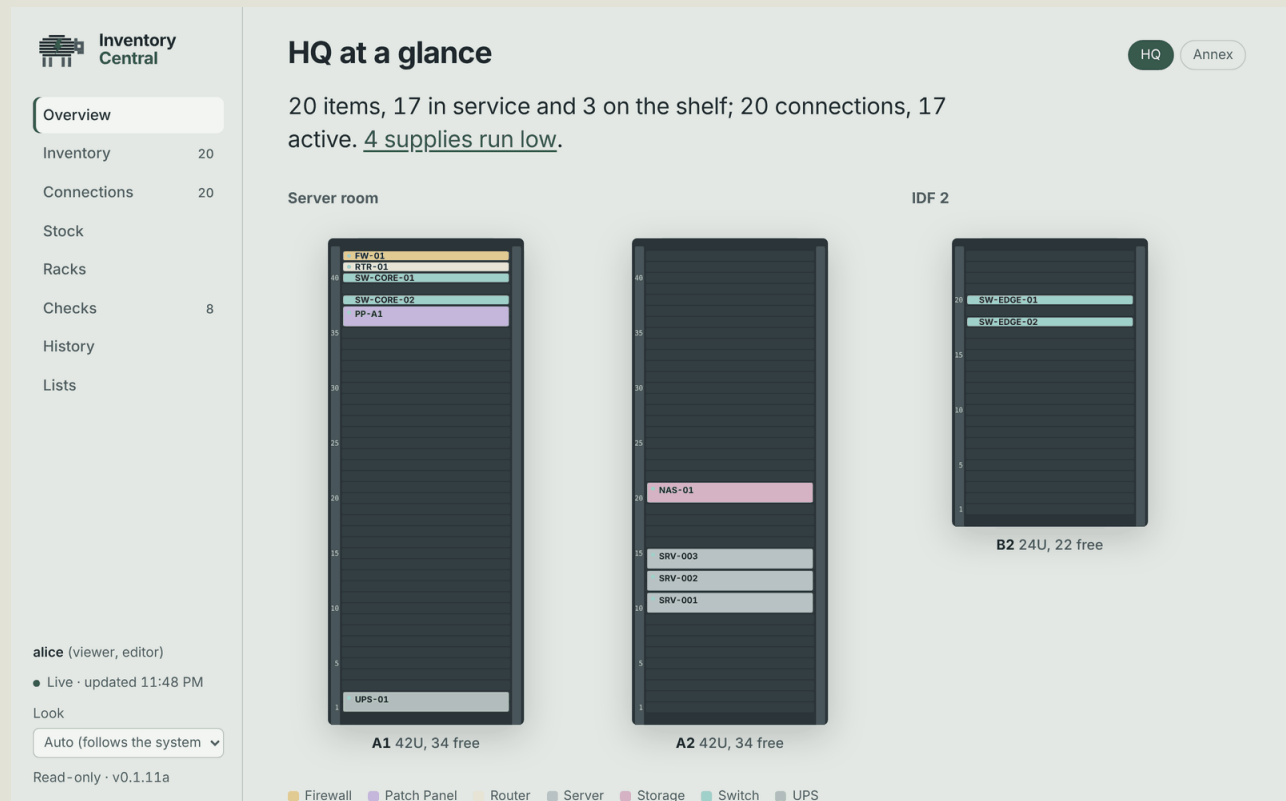


Figure 64. The Overview: a sentence that sums up the site, the site's racks drawn front view (one colour per type of equipment), and below them what needs a look and the latest changes.

Note. The web page does nothing until an administrator sets it up. It needs the server (Chapter 18), nginx in front of it, and a certificate for each person. Nothing of it runs, and no port is opened, unless `inventoryd setup -w` is run as root (section 19.9).

19.1 How people get in

There are no passwords. Each person has a **certificate** that your organisation's own certificate authority issued (`inventory-ca issue`, section 18.16), installed in their browser. nginx asks the browser for it, checks it against the authority and its revocation list, and only then passes the request on, saying whose certificate it was. The person's name is the certificate's name (its common name, such as `alice`), and their role comes from the same `users` and `roles` files as for SSH (section 18.10).

- **No certificate, a revoked one, or one from another authority:** nginx refuses the connection; the browser shows its own error.
- **A certificate but no role:** the page says so and shows no data (Figure 65). The refusal is recorded in the history, at most once in ten minutes for each name (since 0.1.16a; the next line recorded says how many more there were), so a page that keeps asking cannot fill the history.
- **A viewer, an editor or an admin:** the page opens. Only admins see the logins and refusals on the History page. Opening the page is recorded as a login at most once an hour.

A person's roles are worked out again at least every 10 seconds, and at once when the `users` or `roles` file changes: taking someone out of the `users` file counts at their next request (since 0.1.16a).

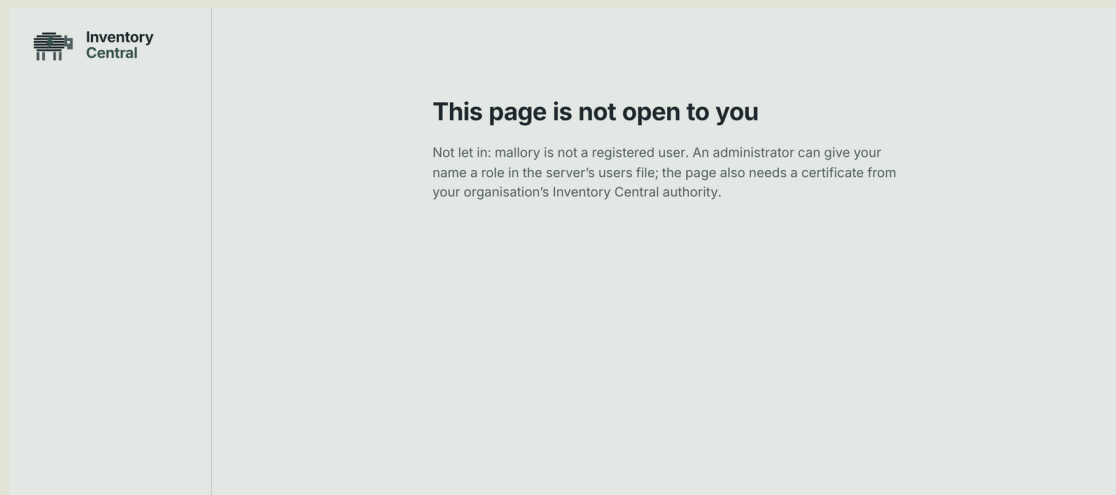


Figure 65. A person whose certificate checks out but whose name has no role.

19.2 Reading the page

The pages on the left (across the top on a phone) match the terminal program's:

- **Overview:** the site at a glance (Figure 64). With more than one site, the buttons at the top switch between them.
- **Inventory:** the equipment in service, with **On the shelf** (in stock or reserved), **Decom** and **All** a click away, as in the terminal program. Select a row for its details (Figure 66): every field, its ports with what each is connected to, and a link to its rack.
- **Connections:** the cabling, both ends; decommissioned ones under **Decom**.
- **Stock:** the supply lines, with a bar for each count against its minimum (amber at or below the minimum, red at none), **Supplies**, **Parts** and **Cables** to show one class and **Running low** to show only those, and the equipment on the shelf.
- **Racks:** each rack at full size, by room (Figure 67). Select a device for its details. Items at the site that are not in a rack unit are listed below.
- **Checks:** what the Checks page (Chapter 12) finds; select one to open the record.
- **History:** every change, newest first, grouped by change set, with what was changed from what to what (Figure 68). **Show older changes** reads further back.
- **Lists:** the Config lists (sites, floors, rooms, locations, types, makers, models and so on) and every port.

The screenshot displays the Inventory Central web interface. On the left is a sidebar with navigation links: Overview, Inventory (17), Connections (20), Stock, Racks, Checks (8), History, and Lists. Below the sidebar, user information for 'alice' is shown, including a 'Live' status indicator and a 'Look' dropdown menu set to 'Shepherd light'. The main content area is titled 'Inventory' and features tabs for 'In service', 'On the shelf', 'Decom', and 'All'. A search bar and buttons for 'Add an item' and 'Print' are present. Below the search bar, a table lists various equipment items. The 'Server' row for 'PowerEdge R740' with serial number '7XK2P43' and asset tag 'SRV-001' is highlighted. To the right of the table, the details for 'SRV-001' are shown, including fields for Serial Number, Component Type, Asset Tag, Manufacturer, Model, Notes, Hostname, Purchase Date, PO, Status (Deployed), Site, Room, Location, Position, Height, CPU, and Memory. Below the details, a section for 'Ports (6)' shows '3 connected' and '3 free' ports, with a list of network ports (eth0, eth1, eth2, eth3) and power ports (PSU1, PSU2).

Inventory

In service On the shelf Decom All

Search the inventory Add an item Print

The equipment in service. Select one for its details, ports and connections.

Type ▲	Model	Serial Number	Asset Tag
Access Point	U6-Pro	F492BF5A1101	AP-101
Access Point	U6-Pro	F492BF5A1102	AP-102
Firewall	FortiGate 100F	FG100FTK21001234	FW-01
Laptop	ThinkPad T14 Gen 4		LT-0045
Monitor	U2723QE	CN0K2PH6	MON-0012
Patch Panel	48-port Cat6A		PP-A1
PDU	AP8941	ZA2201001234	PDU-A1
Router	EdgeRouter 12	F09FC2A81B07	RTR-01
Server	PowerEdge R740	7XK2P43	SRV-001
Server	PowerEdge R740	7XK2P47	SRV-002
Server	ProLiant DL380 Gen10	CZ2231004K	SRV-003
Storage	RS3621xs+	2130TWR123456	NAS-01
Switch	Catalyst 9300-48P	F0C2145X0AB	SW-CORE-01
Switch	Catalyst 9300-48P	F0C2145X0AC	SW-CORE-02

SRV-001
Server, Dell, PowerEdge R740

Edit Delete...

Serial Number 7XK2P43
Component Type Server
Asset Tag SRV-001
Manufacturer Dell
Model PowerEdge R740
Notes Hypervisor host 1
Hostname srv-001
Purchase Date 2022-11-08
PO PO-22-0612
Status ● Deployed
Site HQ
Room Server room
Location A2
Position (U) 10
Height (U) 2
CPU 2 x Xeon Gold 6230
Memory 384 GB DDR4

Ports (6) · 3 connected, 3 free

Network (4)

eth0 SFP+ 10G eth1 SFP+ 10G eth2 SW-CORE... eth3 SW-CORE...

Power in (2)

PSU1 PSU2

Figure 66. The Inventory with an item's details open: its fields, then its ports and what each one connects to.

Every table works the same way: click a heading to sort by it (again to reverse), type in the search box to narrow the rows (the matching text is marked), and select a row for its details. Columns that no row fills are left out. Long text wraps instead of being cut, and the first column stays in view when a wide table scrolls sideways. **Esc** closes the details.

The dot beside **Live** at the bottom left is green while the page is following the server; it turns amber, with **Not reaching the server**, if it cannot, and goes back to green by itself.

Inventory Central

- Overview
- Inventory 20
- Connections 20
- Stock
- Racks**
- Checks 8
- History
- Lists

carol (viewer, editor, admin)
● Live · updated 11:48 PM
Look
Auto (follows the system) ▼
Read-only · v0.1.11a

Racks at HQ

Each rack as it stands, front view, by room. Select a device for its details.

Server room

42	FW-01	*****	42
41	RTR-01	*****	41
40	SW-CORE-01	*****	40
39			39
38	SW-CORE-02	*****	38
37			37
36	PP-A1	*****	36
35	Leviton 48-port Cat6A	*****	35
34			34
33			33
32			32
31			31
30			30
29			29
28			28
27			27
26			26
25			25
24			24
23			23
22			22
21			21
20			20
19			19
18			18
17			17
16			16
15			15
14			14
13			13
12			12
11			11
10			10
9			9
8			8
7			7
6			6
5			5
4			4
3			3
2			2

UPS-01

FW-01
Firewall, Fortinet, FortiGate 100F

Serial Number FG100FTK21001234
Component Type Firewall
Asset Tag FW-01
Manufacturer Fortinet
Model FortiGate 100F
Notes Perimeter firewall; HA pair planned
Purchase Date 2023-01-20
PO PO-23-0009
Status ● Deployed
Site HQ
Room Server room
Rack A1
Position (U) 42
Height (U) 1

Ports (10)

port1	SW-CORE-01 Te1/1/2
port2	RJ45 1G, free
port3	RJ45 1G, free
port4	RJ45 1G, free
port5	RJ45 1G, free
port6	RJ45 1G, free
port7	RJ45 1G, free
port8	RJ45 1G, free

Figure 67. The Racks page in dark mode, with the firewall selected.

Inventory Central

- Overview
- Inventory 20
- Connections 20
- Stock
- Racks
- Checks 8
- History**
- Lists

carol (viewer, editor, admin)
● Live · updated 11:48 PM
Look
Auto (follows the system) ▼
Read-only · v0.1.11a

History

Every change, newest first, with logins and refusals.

Oct 21, 2026, 11:48 PM carol on the web page	Opened the web page as editor, admin
Oct 21, 2026, 11:48 PM alice on the web page	Opened the web page as editor
Oct 21, 2026, 11:48 PM admin via import	Added port type C13 outlet Added port type C14 inlet Added port type RJ45 console Added port type QSFP28 100G Added port type SFP+ 10G Added port type RJ45 1G Added memory size 64 GB DDR4 Added memory size 128 GB DDR4 Added memory size 384 GB DDR4 Added cpu Xeon D-1541 Added cpu 2 x Xeon Silver 4214R Added cpu 2 x Xeon Gold 6230 and 286 more in the same change

Show older changes

Figure 68. The History page as an admin sees it: logins on the web page, then the changes.

19.3 Changing data

People with the **editor** role see **Add** on the Inventory, Connections, Stock and Lists pages, and **Edit** and **Delete...** in each record's details (and **- 1** and **+ 1** for a supply line, as the **-** and **+** keys do). The bottom left says **Can edit** instead of **Read-only**. Viewers and admins without the editor role see none of these.

- **Edit** and **Add** open the record as a form in the details panel, with every field. Fields with a fixed list (a type, a status) are menus; the others offer the same suggestions as the terminal program's form, narrowed the same way: the rooms of the chosen site, the locations of the chosen room, the models of the chosen maker, the ports of the chosen device. **Save** sends the record; **Cancel** or **Esc** drops it.
- **The same rules apply.** The server checks every value with the rules of the terminal program's form: a required field, a tag or serial number already in use, a rack unit already filled, a port already taken, a date or number that cannot be read. A refusal is shown under the field it concerns (Figure 69), and nothing is saved.
- **Renames carry through.** Renaming a site, floor, room or location in Lists changes the floors, rooms, locations and items that name it, and changing an item's tag moves its ports with it, all in the same change; the message says how many followed.
- **Deleting** asks first, in the details panel (Figure 70). As in the terminal program, an item or a connection is **moved to Decom**: an item leaves any rack it is in, and its connections in service are decommissioned with it. On Decom, **Delete** deletes it for good, and an item's ports go with it; to bring one back, **Edit** it and give it a status again. A site, floor, room or location still in use is not deleted, and the page says what still names it.
- **Two people at once.** The page sends each change with the version of the record it showed. If someone else changed or deleted the record in the meantime, nothing is saved. Since 0.1.16a a form keeps what you typed and says what the other person changed; **Save** again puts yours in place of theirs, on purpose this time, and **Cancel** shows theirs. A change without a form (**+ 1**, a move, a delete) shows the record as it now is, and you make your change again if it still applies. This is the same rule as between two terminal programs (section 18.4).
- **One change at a time.** While a change is on its way to the server, the buttons for further changes wait, so a second press of **+ 1** cannot be sent with the old version (since 0.1.16a).
- **Esc on a form you have changed** asks before discarding it (**Discard / Keep editing**); a second **Esc** discards. A form with nothing typed closes at once, as before.

Every change is recorded in the history under the person's name, marked **on the web page**.

Inventory Central

- Overview
- Inventory** 20
- Connections 20
- Stock
- Racks
- Checks 8
- History
- Lists

alice (viewer, editor)

● Live · updated 12:25 AM

Look

Auto (follows the system) ▾

Can edit · v0.1.12a

Inventory

All In service On the shelf Search the inventory

Add an item

Every piece of equipment. Select one for its details, ports and connections.

Asset Tag ▲	Type	Status	Manufacturer	Model
AP-101	Access Point	● Deployed	Ubiquiti	U6-Pro
AP-102	Access Point	● Deployed	Ubiquiti	U6-Pro
AP-103	Access Point	● Stock	Ubiquiti	U6-Pro
FW-01	Firewall	● Deployed	Fortinet	FortiGate 100F
LT-0045	Laptop	● Deployed	Lenovo	ThinkPad T14 Gen 4
LT-0046	Laptop	● Stock	Lenovo	ThinkPad T14 Gen 4
MON-0012	Monitor	● Deployed	Dell	U2723QE
NAS-01	Storage	● Deployed	Synology	RS3621xs+
PDU-A1	PDU	● Deployed	APC	AP8941
PP-A1	Patch Panel	● Deployed	Leviton	48-port Cat6A
RTR-01	Router	● Deployed	Ubiquiti	EdgeRouter 12
SRV-001	Server	● Deployed	Dell	PowerEdge R740
SRV-002	Server	● Deployed	Dell	PowerEdge R740

Edit AP-101

Checked by the same rules as the terminal program's form.

Serial Number
F492BF5A1101

Component Type
Access Point

Asset Tag
FW-01
That Asset Tag is already in use by another item

Manufacturer
Ubiquiti

Model
U6-Pro

Notes
PoE from SW-EDGE-01; moved to the east wall

Location
Floor 1 hallway

Purchase Date
2024-05-06

Save Cancel

Figure 69. Editing an item: the tag typed is already in use, so the server refuses it and the form marks the field.

Inventory Central

- Overview
- Inventory** 17
- Connections 20
- Stock
- Racks
- Checks 8
- History
- Lists

alice (viewer, editor)

● Live · updated 09:35 PM

Look

Shepherd light ▾

Can edit · v0.1.17a

Inventory

In service On the shelf Decom All

Search the inventory **Add an item** Print

The equipment in service. Select one for its details, ports and connections.

Type ▲	Model	Serial Number	Asset Tag
Access Point	U6-Pro	F492BF5A1101	AP-101
Access Point	U6-Pro	F492BF5A1102	AP-102
Firewall	FortiGate 100F	FG100FTK21001234	FW-01
Laptop	ThinkPad T14 Gen 4		LT-0045
Monitor	U2723QE	CN0K2PH6	MON-0012
Patch Panel	48-port Cat6A		PP-A1
PDU	AP8941	ZA2201001234	PDU-A1
Router	EdgeRouter 12	F09FC2A81B07	RTR-01
Server	PowerEdge R740	7XK2P43	SRV-001
Server	PowerEdge R740	7XK2P47	SRV-002
Server	ProLiant DL380 Gen10	CZ2231004K	SRV-003
Storage	RS3621xs+	2130TWR123456	NAS-01
Switch	Catalyst 9300-48P	F0C2145X0AB	SW-CORE-01
Switch	Catalyst 9300-48P	F0C2145X0AC	SW-CORE-02

MON-0012

Monitor, Dell, U2723QE

Move item MON-0012 to Decom? It becomes Decommissioned: it leaves any rack it is in, and its connections in service are decommissioned with it. It can be given a status again later. The history keeps a record of it.

Move to Decom Keep it

Serial Number CN0K2PH6

Component Type Monitor

Asset Tag MON-0012

Manufacturer Dell

Model U2723QE

Purchase Date 2023-08-30

PO PO-23-0402

Status ● Deployed

Add a port

Figure 70. Deleting asks first, in the details panel.

Note. Changes come only from the page itself. A change must carry the page's own token, given to the person when the page opens, be sent as JSON, and come from the same site; nginx passes POST only to the one address that takes changes. A link or a form on another site cannot change anything, even in a browser that holds a valid certificate.

19.4 Moving equipment in a rack

Since 0.1.14a an editor can move a device on the **Racks** page by dragging it with the mouse: to another unit of the same rack, or into another rack at the same site. While you drag, the place it would go is outlined, green when the units are free and red when something is already there (Figure 71). Let go to move it; press **Esc** before letting go to change your mind. A drop onto taken units, or one that would not fit, is refused at once and the page says what is in the way. The move is an ordinary change: the server checks the units again, records it in the history under your name, and other people see it within a few seconds.

Without a mouse, select the device and use **Up one unit** and **Down one unit** in its details, or give the device the focus in the drawing (with **Tab**) and press **Shift-Up** or **Shift-Down**. On a touch screen dragging is left off, so that a finger on a rack scrolls the page; use the buttons in the details. Moving to another rack without a mouse is done with **Edit**, as before.

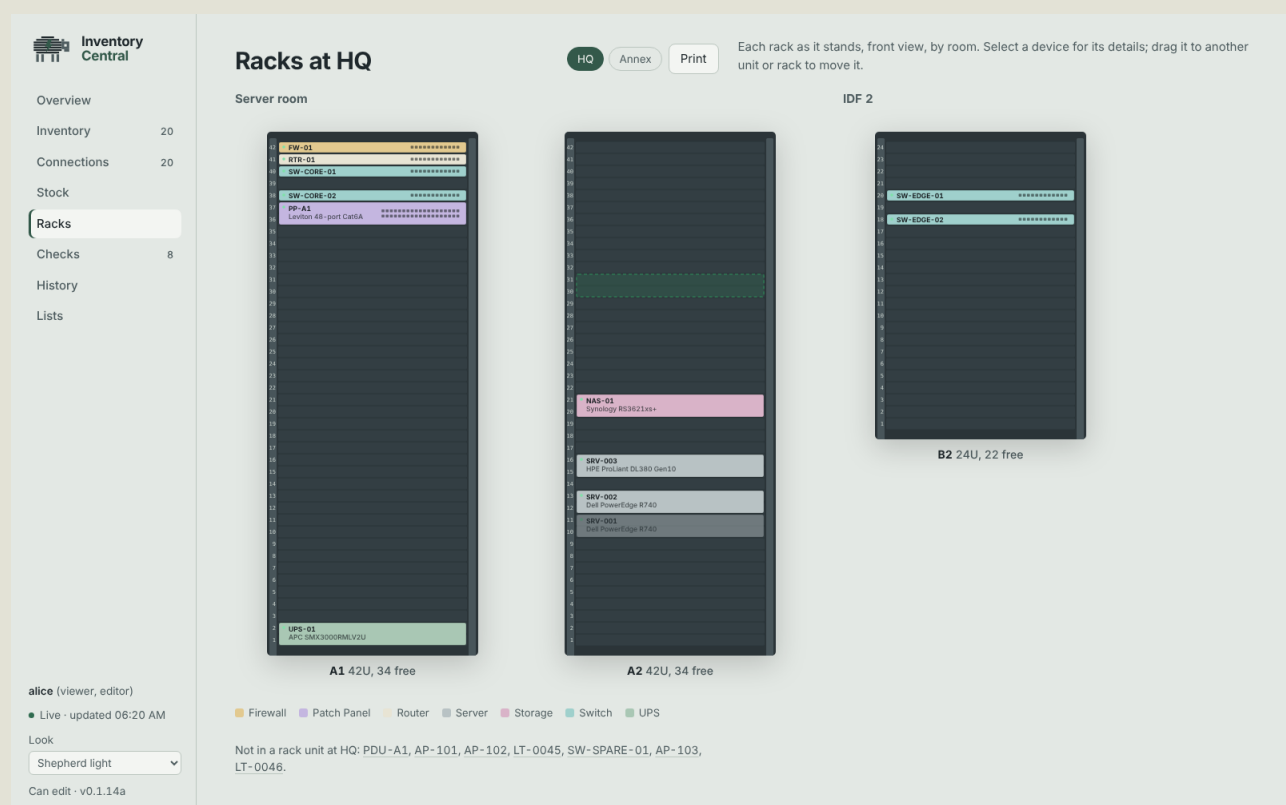


Figure 71. Dragging SRV-001 up its rack: the dashed outline shows where it will go, green because the units are free.

19.5 The port map

An item's details show its ports as a map (Figure 72): one tile per port, grouped by kind (network ports, power inputs, outlets), each with its name and either what it is connected to or its port type. Connected ports are shaded. Select a connected port to open its connection. For an editor, selecting a **free** port starts a new connection from it: the form opens with this device and this port as the A end, and you choose the other end. **Add or change ports** goes to the Ports list, searched for this device; a device with no ports offers **Add a port**.

The screenshot displays the 'Inventory' page in the 'Inventory Central' application. The main table lists various equipment items, including Access Points, Firewalls, Laptops, Monitors, Storage, PDUs, Patch Panels, Routers, Servers, and Switches. The 'SW-EDGE-01' switch is highlighted, and its port map is shown on the right side of the interface.

The port map for 'SW-EDGE-01' shows 52 network ports and a power input. The ports are organized into a grid, with each tile displaying the port name, type, and connection status. The 'Tel1/1/1 SW-CORE...' port is highlighted, indicating it is connected.

Asset Tag	Type	Status	Manufacturer	Model	Serial Number	Site	Room
AP-101	Access Point	Deployed	Ubiquiti	U6-Pro	F492BF5A1101	HQ	Floor 1
AP-102	Access Point	Deployed	Ubiquiti	U6-Pro	F492BF5A1102	HQ	Floor 1
AP-103	Access Point	Stock	Ubiquiti	U6-Pro	F492BF5A1103	HQ	Storage
FW-01	Firewall	Deployed	Fortinet	FortiGate 100F	FG100FTK21001234	HQ	Server
LT-0045	Laptop	Deployed	Lenovo	ThinkPad T14 Gen 4		HQ	IT Center
LT-0046	Laptop	Stock	Lenovo	ThinkPad T14 Gen 4	PF3XK210	HQ	IT Center
MON-0012	Monitor	Deployed	Dell	U2723QE	CN0K2PH6		
NAS-01	Storage	Deployed	Synology	RS3621xs+	2130TMR123456	HQ	Server
PDU-A1	PDU	Deployed	APC	AP8941	ZA2201001234	HQ	Server
PP-A1	Patch Panel	Deployed	Leviton	48-port Cat6A		HQ	Server
RTR-01	Router	Deployed	Ubiquiti	EdgeRouter 12	F09FC2A81B07	HQ	Server
SRV-001	Server	Deployed	Dell	PowerEdge R740	7X02P43	HQ	Server
SRV-002	Server	Deployed	Dell	PowerEdge R740	7X02P47	HQ	Server
SRV-003	Server	Deployed	HPE	ProLiant DL380 Gen10	CZ2231004K	HQ	Server
SW-CORE-01	Switch	Deployed	Cisco	Catalyst 9300-48P	F0C2145X0B0B	HQ	Server
SW-CORE-02	Switch	Deployed	Cisco	Catalyst 9300-48P	F0C2145X0B0C	HQ	Server
SW-EDGE-01	Switch	Deployed	HPE Aruba	2930F 48G PoE+	SG98KQ1234	HQ	IDF
SW-EDGE-02	Switch	Deployed	HPE Aruba	2930F 48G PoE+	SG98KQ1301	HQ	IDF
SW-SPARE-01	Switch	Stock	Cisco	Catalyst 9300-48P	F0C2145X0B0D	HQ	Storage

The port map on the right shows 52 network ports (RJ45 1G) and a power input (PSU1 C14 inlet). The ports are organized into a grid, with each tile displaying the port name, type, and connection status. The 'Tel1/1/1 SW-CORE...' port is highlighted, indicating it is connected.

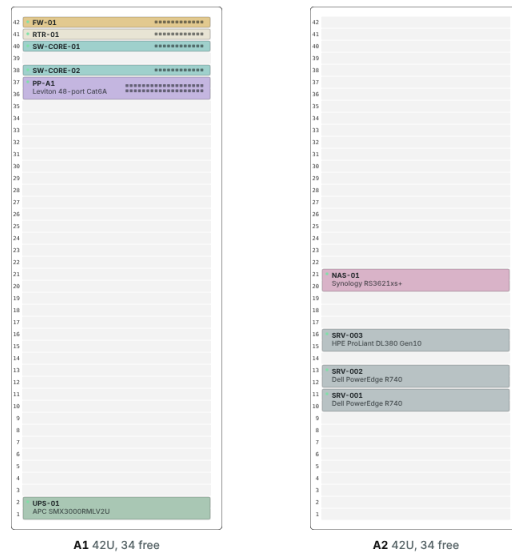
Figure 72. SW-EDGE-01's port map: 52 network ports and a power input, four of them connected.

19.6 Printing

Print on the Inventory, Connections, Stock and Racks pages prints the page as it is shown: searched, filtered and sorted. The menus, buttons and details panel are left out, the colours are made for paper, tables repeat their headings on every page and do not break a row across pages, and racks are not split between pages. A line under the title says who printed it, when, and how current the data was (Figure 73). The browser's own print command does the same. For wide tables, choose landscape in the print dialog.

Racks at HQ

Inventory Central · printed Oct 22, 2026, 06:20 AM by alice · data as of change 460

Server room**IDF 2**

■ Firewall
 ■ Patch Panel
 ■ Router
 ■ Server
 ■ Storage
 ■ Switch
 ■ UPS

Figure 73. The Racks page as printed: rack elevations for a wall or a work order.

19.7 Closing the page

An administrator can close the web page without stopping the server or taking the site down: in the terminal program, connected to the server as an admin, the Settings page has a **Web page (server)** row (section 14.7). **Closed**, the page shows nobody any data and says that an administrator closed it; an open record or form is cleared too. It opens again by itself, within seconds, when an admin sets the row back to **Open**. Both are recorded in the history. In the same way, a person whose role is taken away while the page is open sees it close (*This page is no longer open to you*), and it opens again once they are let in.

19.8 Looks

Look at the bottom left offers the terminal program's colour looks: **Auto** (the default) follows the computer's light or dark setting, **Shepherd light** is the light version of the Electric Shepherd palette, and the dark looks run from cool **Winter** to warm **Samhain**. The choice is kept in that browser. Every look keeps text readable (a contrast of at least 4.5 to 1); the racks keep their colours in every look, as real equipment would.

Inventory Central

Overview
Inventory 17
Connections 20
Stock
Racks
Checks 8
History
Lists

carol (viewer, editor, admin)
● Live · updated 09:35 PM
Look
Samhain
Can edit · v0.1.17a

Stock

All Supplies Parts Cables Running low Search supplies Add a supply line Print

Supplies, parts and cables with their counts against the minimum, and the equipment on the shelf (in stock or reserved).

Item ▲	Class	Category	Spec	Connector	Color	Length	Qty	N
Cable labels, roll	Supply	Labels and ties			White		4	
Cat6 patch cable	Cable	Copper patch cable	Cat6 UTP	RJ45	Grey	7 ft	15	
Cat6a patch cable	Cable	Copper patch cable	Cat6a UTP	RJ45	Blue	1 m	40	
Cat6a patch cable	Cable	Copper patch cable	Cat6a UTP	RJ45	Blue	3 m	35	
Fan module	Part	Spare part					3	
LC/LC patch cable	Cable	Fiber patch cable	OM4 duplex	LC/LC	Aqua	1 m	24	
LC/LC patch cable	Cable	Fiber patch cable	OM4 duplex	LC/LC	Aqua	3 m	18	
LC/LC patch cable	Cable	Fiber patch cable	OM4 duplex	LC/LC	Aqua	10 m	6	
LC/LC patch cable	Cable	Fiber patch cable	OS2 duplex	LC/LC	Yellow	3 m	12	
LC/LC patch cable	Cable	Fiber patch cable	OS2 duplex	LC/LC	Yellow	10 m	4	
MPO/MTP trunk, 12 fibre	Cable	Bulk fiber	OM4 duplex	MPO/MTP	Aqua	10 m	2	
Power supply, 800 W	Part	Spare part					2	
QSFP28 100GBASE-SR4	Supply	QSFP module	100GBASE-SR4	QSFP28			0	
SFP 1000BASE-T	Supply	SFP / SFP+ module	1000BASE-T	SFP			6	
SFP+ 10GBASE-LR	Supply	SFP / SFP+ module	10GBASE-LR	SFP+			2	

Figure 74. The Stock page in the Samhain look. Counts at or below their minimum are amber; none at all is red.

On a phone the pages run across the top, the racks scroll sideways, and the details fill the screen.

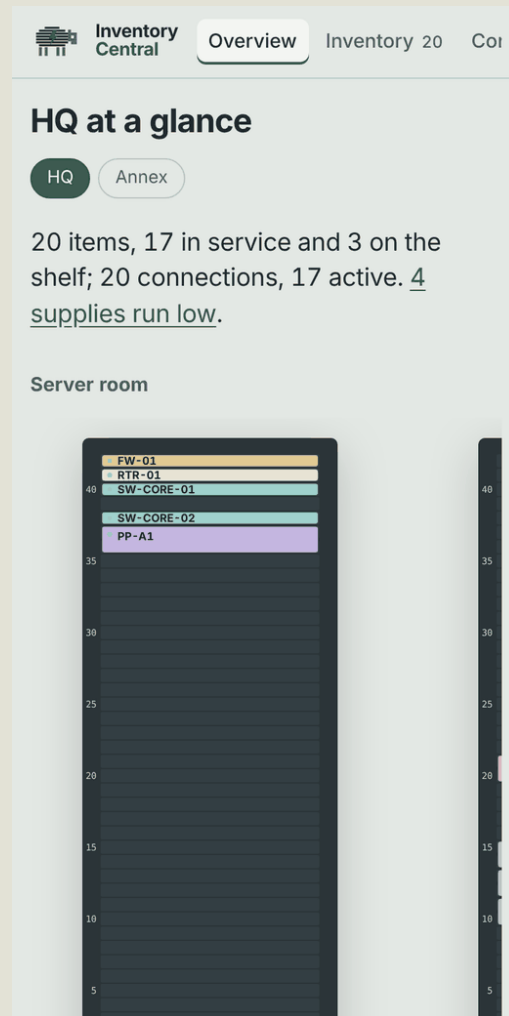


Figure 75. The Overview on a phone.

19.9 Setting it up

This is for the administrator of the server's computer. It assumes the server is set up as a service (section 18.14) and nginx is installed: the distribution's package on Linux, Homebrew's on macOS (since 0.1.14a; see the end of this section). The *Server Setup Guide* beside this manual goes through it step by step.

1. **Make the certificates** with `inventory-ca`, as `CA-SETUP.md` describes: the authority, limited to the page's host names (`init "Example Corp" inventory.example.org`), a certificate for the server's name (`server inventory.example.org`), and one for each person (`issue alice`).
2. **Put the server's files where setup looks for them**, `/etc/inventory-central/web` (or give another folder with `-c`): `server.crt` (the `-chain.crt` file), `server.key`, and, with `inventory-ca` `publish /etc/inventory-central/web`, `ca.crt` and `crl.pem`.
3. **Run setup with `-w`** and the name people will type, first as a dry run:

```
inventoryd setup -n -w inventory.example.org
inventoryd setup -w inventory.example.org
```

Setup checks that the four files are there before it changes anything. Then it makes a random key that nginx passes to the server with every request (`/etc/inventory-central/web/proxy-key`, the service account's alone, and `proxy-key.conf` beside it, root's own, which the site includes), starts the server with `-W /run/inventory-central/web.sock -K` and that key (systemd makes the socket's folder for the service, readable by the server's group only), writes the nginx site (`/etc/nginx/sites-available/inventory-central` with its link in `sites-enabled`, or `/etc/nginx/conf.d/inventory-central.conf`), adds nginx's account (`www-data`, or `nginx`; `-N` for another) to the server's group, runs `nginx -t` and reloads nginx. All of it is recorded for teardown. `-w` can be added to a server that was set up without it: run setup again with `-w`.

Finally, **give each person a role** under their certificate's name in the server's `users` file, and give them their `.p12` file and its password by separate ways.

The site setup writes listens on port 443 (and on port 80 only to send people to HTTPS), speaks TLS 1.2 and 1.3, requires a client certificate checked against the authority and its revocation list, passes GET and HEAD requests, and POST only to the address that takes changes, and replaces any name a browser might send with nginx's own finding. The server answers only requests that carry the key, so another program that can reach the socket (one running as nginx's account, say) cannot claim a name. A site written by 0.1.11a passes no POST: run `inventoryd setup -w` again to rewrite it. The server's web socket is not a network port: nginx reaches it as a file, and the server itself still has no network at all (section 18.14).

Careful. Keep the revocation list current. The list is good for 30 days, and once it runs out nginx refuses everyone. Renew it at least weekly on the computer that holds the authority, never from cron on the web server (`CA-SETUP.md` has a crontab line), or make a longer one by hand with `inventory-ca crl -d 90`. Put it in place with `inventory-ca publish` (from another computer: copy it to a new name, then `mv` it over the old one), then reload nginx. Use `publish` rather than copying the file over the old one: nginx 1.27.4 and later keep what they read from a file across reloads as long as the file's time and inode look unchanged, so a list copied over the old one in the same second can go unnoticed. `publish` writes a new file and renames it into place, which nginx always notices. (Testing this release found exactly that: a revoked certificate still let in after a reload.)

On macOS (since 0.1.14a) setup uses Homebrew's nginx (`brew install nginx`), under `/opt/homebrew` on Apple silicon or `/usr/local` on Intel Macs. The site goes into its `servers/` folder, which Homebrew's `nginx.conf` reads (setup says so if yours does not). The web socket is in `/usr/local/var/run/inventory-central`, a folder setup makes for the service account and a group of its own, `inventoryweb`, to which it adds the account and nginx's (`nobody` unless you give `-N`): a Mac account's own group is usually `staff`, which every person on the Mac is in, so it cannot be used. The `launchd` job gains `-W` and `-K`. Setup runs `nginx -t` and `nginx -s reload`; if nginx is not running yet, start it with `sudo brew services start nginx` (it must run as root to listen on port 443). The macOS steps have been tested with the files setup writes, not yet on a Mac.

To take the web page away again, `inventoryd teardown` removes the site with the rest of the setup (section 18.15).

19.10 What the page does not do yet

- **Importing and exporting** files stay in the terminal program; for a file, use its export (Chapter 15). The page prints (section 19.6).

- **Moving to another site** is done with **Edit**: dragging moves within a site's racks.
- **Port names** are changed on the Ports list, not on the map.

20. The Command Line

```
inventory [-Dn] [-c look] [-p page] [-f file | file | -S socket | -R host]
inventory -d [-n] [-c look] [-p page] [-s WxH] [-k keys]
           [-f file | file | -D | -S socket | -R host]
inventory -h | -V | -L
```

The command line follows the POSIX utility conventions: options are single letters that may be grouped (-Dn), a value may follow the letter after a space or directly (-f kit.csv or -fkit.csv), -- ends the options, and each has a long form as well. Options may come after the file name too, unless `POSIXLY_CORRECT` is set in the environment, in which case the first operand ends the options, as the standard says.

20.1 Options

Option	Long form	What it does
-f FILE	--file FILE	Use this data file for the inventory; the other tables and lists are kept beside it (section 17.1). The file can also be given as the one operand: <code>inventory kit.csv</code> . A lone - is refused, as the program cannot read a table from standard input.
-D	--demo	Start with the sample data. Nothing is read from or written to any file. Cannot be combined with a data file.
-d	--dump	Print one page as plain text and exit. Needs no terminal. See below.
-s WxH	--size WxH	With -d: the size of the page to print, columns by rows, each from 1 to 1000 (default 100x32). Example: <code>-s 120x34</code> .
-k KEYS	--keys KEYS	With -d: press these keys before printing. See below.
-p PAGE	--page PAGE	Open (or print) this page: summary, checks, stock, supplies, cables, inventory, connections, ports, decom, history, config, a Config list (sites, floors, rooms, locations, types, manufacturers, models, cpus, memory, port-types) or settings. The menu's number key (-p 4) and a few other spellings (items, assets, links, network, racks, decommissioned) are accepted too. Overrides the Start on page setting for this run.
-c THEME	--colors THEME	Use this color theme for this run only: full, teal, midnight, winter, harbor, dusk, orchid, shepherd, spring, sandstone, autumn, ember, samhain, paper, white, green or amber. Not saved (section 14.1).
-n	--no-splash	Skip the start-up splash screen.
-S SOCKET	--server SOCKET	Work on the data of a running <code>inventoryd</code> through its socket (<code>DIR/inventoryd.sock</code>) instead of files (Chapter 18). Cannot be combined with -D or a data file. With -d, prints a page of the server's data.

Option	Long form	What it does
-R HOST	--remote HOST	The same through SSH: [user@]host, where the key runs inventoryd-session (section 18.9). Runs ssh -T -e none -- HOST, or the program named by INVENTORY_SSH.
-h	--help	Print the usage text and exit.
-V	--version	Print the version line and exit, for example Inventory Central v0.1.16a (alpha, built 2026-10-23).
-L	--licence, --license	Print the terms of use and exit (section 1.7).

Mistakes in the command line are reported with a pointer to the help, and the exit status is 2:

```
$ inventory -p cabling
inventory: unknown page 'cabling' (use summary, checks, stock, inventory, connections, decom, history, config, a Config list such as sites or locations, or settings)
Try 'inventory --help' for more information.
```

-s and -k work only with -d, and -k not with -S or -R (since 0.1.16a); more than one data file, or a file with -D, is refused. -h, -V and -L win over the other options, in that order.

20.2 Printing a page as text: `--dump`

inventory -d draws one page exactly as the program would, as plain text (no colors, no escape sequences), writes it to standard output and exits. It does not need a terminal, so it works in scripts, in a cron job, over a pipe and in a bug report:

```
$ inventory --demo -d -s 100x24 -p checks | head -12
```

- The page is drawn at the size given with -s. The layout follows the size, so -s 80x24 gives the top-bar layout and -s 120x34 the sidebar layout (Chapter 2). Text is Unicode if the locale is UTF-8 and ASCII otherwise, as in the program.
- --demo together with -d always shows the same date and time (Tuesday, October 20, 2026, 2:07 PM), so the output is the same every time. That is how the screens in this manual were produced.
- A dump only reads. It does **not** take the lock, so it works while the program is running in another window, and it never writes your data files, whatever keys you press (a report that you ask for with the keys of the export form is written, since that is what you asked for). With -S or -R, -k is refused: a dump of a server's data only reads (since 0.1.16a).

-k KEYS presses keys first, as if you had typed them, and then prints the result. The keys are a list separated by spaces:

Written as	Means
a, Z, /, 3	That character.
Enter Esc Tab Shift-Tab Backspace Delete Insert Space	The key of that name (upper or lower case; Return, Escape, BackTab, BS and Del also work).
Up Down Left Right Home End PgUp PgDn	The movement keys.
F1 to F12	Function keys.
Ctrl-S, C-s, Alt-x, M-x	A key with Control or Alt.
text:Cisco	The characters C, i, s, c, o typed one after another; no spaces inside.
paste:dy	The characters d, y pasted, as a terminal sends pasted text (since 0.1.16a): to check that a paste is not taken for keys.

For example, to see the Connections page filtered on spare cables, and a form for a new supply line open on top:

```
inventory --demo -d -s 120x34 -p connections -k "f text:spare Enter"
inventory --demo -d -p locations -k "a"
inventory --demo -d -p supplies -k "a text:Cable Tab"
```

Because the keys really run through the program, `--dump -k` is also a convenient way to check what a sequence of keys does before you rely on it, and it shows exactly what a bug report should show.

20.3 Environment

Variable	Effect
INVENTORY_FILE	The data file to use when -f and an operand are not given (section 17.1).
INVENTORY_CONFIG	The settings file to use instead of <code>~/.config/inventory-central/settings</code> .
XDG_DATA_HOME, XDG_CONFIG_HOME	Where the standard data folder and settings folder are, when they are set to an absolute path. Otherwise <code>~/.local/share</code> and <code>~/.config</code> are used.
HOME	Your home folder: the default for the two above, the meaning of <code>~/</code> in the export and import forms, and where the earlier version's <code>inventory.csv</code> is looked for.
INVENTORY_NO_SPLASH	Set to anything but empty or 0 to skip the splash screen.
INVENTORY_ASCII	Set to anything but empty or 0 to draw with plain ASCII characters.
INVENTORY_UNICODE	Set to anything but empty or 0 to draw with Unicode even if the locale does not say UTF-8. <code>INVENTORY_ASCII</code> wins if both are set.
LC_ALL, LC_CTYPE, LANG	Looked at in that order for the words UTF-8 or utf8. With neither, the program draws in ASCII.

Variable	Effect
<code>NO_COLOR</code>	If set and not empty, the program starts in the Mono white theme (section 14.1).
<code>COLORTERM</code>	<code>truecolor</code> or <code>24bit</code> selects exact colors; otherwise the nearest of the 256 palette colors.
<code>COLUMNS, LINES</code>	Override the window size that the program asks the terminal for, when each is a whole number from 1 to 1000. Useful where the terminal does not report a size.
<code>POSIXLY_CORRECT</code>	The first operand ends the options.
<code>LOGNAME, USER</code>	The name recorded as Who in the change history (<code>LOGNAME</code> first; <code>unknown</code> if neither is set). Through SSH (<code>-R</code>) the server records the name the key gives instead.
<code>INVENTORY_SSH</code>	The program <code>-R</code> runs instead of <code>ssh</code> (for example a wrapper that adds options).

Inventory Central does not use `TERM` or the terminfo database. It assumes a terminal that understands the common xterm (ECMA-48) escape sequences, which every current terminal emulator does, and it finds the window size with `stty size`. The time comes from `date`, so your `TZ` setting applies.

20.4 Exit status

Status	Meaning
0	Done: the program was quit normally, or <code>-d</code> , <code>-h</code> , <code>-V</code> or <code>-L</code> finished.
1	A failure: the data could not be read (a damaged file, an unknown folder), another copy holds the lock, there is no terminal, or an internal error stopped the program. A message on standard error says which.
2	The command line was wrong.

Messages for the user go to standard error and the dump to standard output, so `inventory -d > page.txt` keeps the page and still shows any problem on the screen.

20.5 The helper copy

While the interactive program runs, you may see a second `inventory` process whose command line starts with `--restore-guard`. It is part of the program: a small helper that does nothing until the real program is killed. It then restores your terminal's settings and screen and removes the lock files (the data's and, since 0.1.16a, the change history's), which a program written with only the standard library cannot do by itself, as the standard library cannot catch signals. It stops by itself when the program quits normally. It is not meant to be run by hand.

21. When Something Goes Wrong

The problems below are the ones most likely to come up, with what they mean and what to do. The messages are quoted as the program words them. If yours is not here, section 21.7 says what to send.

21.1 Starting

inventory: stty failed: standard input is probably not a terminal. Run it in a terminal (or use --dump to print a page). The interactive program needs a real terminal on its standard input. You may be running it from a script, from a pipe (echo | inventory), through ssh without a terminal (ssh host inventory needs ssh -t host inventory), or from an editor's output pane. Run it in a terminal window, or use --dump to print a page as text (section 20.2).

inventory: another copy of Inventory Central (process 4242) has these files open. Another copy is using the same data files. Switch to it and quit it, or, if you are sure it is not running, delete the lock file the message names. A lock that was left behind by a copy that is gone is replaced automatically, so you should need to delete one by hand only if it is damaged, or if the other copy runs on another computer that shares the folder (section 17.4). Never delete a lock while another copy is running: that is what it is there to prevent.

inventory: /path/inventory.csv: line 12: a quoted value is never closed ... (or another message naming a file and a line). A data file cannot be read, and the program has refused to start rather than risk overwriting it. Nothing has been changed. Open the file in a text editor, go to the line named, and repair it; the .bak file beside it is a copy from the start of your last session (section 17.2).

inventory: the folder /somewhere does not exist You named a data file in a folder that is not there. Create the folder, or name another. Only the standard data folder is created automatically.

inventory: cannot examine the folder /somewhere: Permission denied The folder exists, or a folder above it does, but you are not allowed to look inside. Change its permissions, or name a file somewhere else. (The program does not guess: it never starts with an empty list because it could not look.)

The screen shows sample data, not mine. You started with -D or --demo. Run inventory without it.

My own data is not there. Check which files are in use: the Settings page lists the data files (press **0**). Perhaps INVENTORY_FILE is set in your environment, or you started with a different -f. The earlier version's inventory.csv in your home folder is used only when the standard folder has no data yet (section 17.1).

21.2 The screen

Frames and arrows appear as letters or boxes (lqqqk, â, ?). Your terminal and the program disagree about the character set. In a UTF-8 terminal, make sure the locale says so (echo \$LANG should contain UTF-8; export LANG=en_US.UTF-8 or your own language). If your terminal cannot show Unicode, set INVENTORY_ASCII=1 to draw with plain characters. If the locale is wrong but the terminal is fine, INVENTORY_UNICODE=1 forces Unicode (section 2.5).

The window shows "too small". The program needs at least 50 columns by 12 rows. Make the window larger; it carries on by itself within a second. While the window is too small, keys wait; only **q**,

Ctrl-C and **Ctrl-L** act. If you are sure the window is large enough but the program disagrees, your terminal may not report its size: set `COLUMNS` and `LINES` before starting.

Colors look dull, wrong or very different from the manual. Try another theme in Settings (**0**, Theme) or start with `-c white` to see the plain version. The Shepherd theme needs a terminal with 24-bit color to be exact (`COLORTERM=truecolor`); otherwise it uses the nearest of 256 colors. A terminal theme with an unusual palette changes the **Full color** theme, which uses the terminal's own colors. Light terminals work best with Full color, Paper or Mono white.

Part of the screen is garbled after something else wrote to the terminal. Press **Ctrl-L** to redraw it.

Columns are out of line when a value contains Chinese, Japanese or Korean characters, emoji or accents made of combining marks. The program counts one character as one column. Characters that a terminal draws two columns wide, and marks drawn on top of another character, push the rest of the row out of place. Your data is not affected; see Chapter 22.

The terminal is in a strange state after the program stopped (no echo, the screen stays on the program's colors). This happens only if the program was stopped in a way that gives the helper no chance to clean up, such as closing the window. Type `reset` and press Enter (you may not see what you type), or `stty sane`.

21.3 Keys

Esc reacts a moment late. A terminal sends the same first byte for **Esc** and for the arrow keys, so the program waits a tenth of a second to see whether more follows. Over a slow connection the wait can be longer. Nothing is wrong.

Home, End, PgUp or the function keys do nothing. Some terminals send codes for these that the program does not know. Every one has an alternative: **g** and **G** for first and last row, **Ctrl-S** for **F2**, **Up** and **Down** for paging when that is all you have.

A key I press in a form is ignored, or Ctrl-C does nothing. In a form **Ctrl-C** does not quit, so that a stray press cannot lose what you were typing. **Esc** cancels the form; then **q** quits. Everywhere else **Ctrl-C** is the same as **q**.

The keys act on a different row than I expect. Keys act on the highlighted row of the *visible* list, which is the sorted and filtered one. A filter or search may be hiding rows: the page heading says so (2 of 17 items), and **Esc** clears it. A site page or a type shows only that site or type: the page heading says which (Inventory > HQ > Switch), and **v** and **j** step on to All. An item that is not in service is on Stock or Decom, not on the Inventory.

h does not open the help. Since 0.1.1a **h** and **l** move between columns, as in vi. Help is **?** or **F1**.

The sidebar has gone. **b** switches it off and on, and the setting is saved. It also only appears in windows of 100 by 24 or more (section 14.5).

21.4 Records

The form refuses to save. Read the message; it names the field and the cursor is in it. The common ones: a missing tag and serial number, a tag or cable ID that already exists, a date that is not a date, a quantity that is not a whole number, and a port already used by another active cable (`Port Te1/1/3 on SW-CORE-01 is already used by C-0004`). To use a port that an old cable still holds, change that cable's status to Spare, Disabled or Decommissioned, or correct its port: a cable in one of those states does not hold its ports (section 7.3).

The form says a port is not on record, or a power port is refused. At an end whose device has ports on record, the port must be one of them: press `Ctrl-N` in the Port field for the list, add the port on Inventory: Ports, or press `p` on the item. A power port can only be used by a cable of Type Power, and a power cable joins a Power in port to a Power out port (section 7.4).

The form says a rack unit is already filled, or does not fit. Another item in service already has those units, or the item would stick out of the top of the rack. Check its Position and Height, and the rack's Units on Config: Locations. An item out of service takes no units.

A site, floor, room or location cannot be deleted. It is still in use, and the message says by what. Move or delete those items, locations, rooms and floors first (section 9.1).

The Checks page says a device of a connection "is not in the Inventory". The device name in a connection is compared with the **Asset Tag** (or the Serial Number) of your items, ignoring capitals. A different spelling counts as a different device. Pick the device from the suggestion list so that the name is exact (section 12.1).

I deleted something by mistake. There is no undo. The `.bak` file beside the data file holds that table as it was *before the first change of the current session*, so if you notice at once you can go back: quit the program, copy `inventory.csv.bak` over `inventory.csv` (and the same for the other files if they are affected), and start again. Everything you changed in that session is then undone, not only the mistake, and the backup is replaced the next time you run the program and change something, so it only covers the most recent session. If you export regularly (Chapter 15) or back the files up, restore from that instead; your own backup is always the better one.

A search or filter finds nothing, or less than I expect. A search looks in every column for the text, ignoring capitals; a filter looks only at the current column (Chapter 10). The text may be any part of a value. The message line says how many rows matched.

21.5 Saving and files

`NOT SAVED (...)`. Changes are in memory only; press `w` to try again. The file could not be written: the disk is full, the folder is read-only, or the file system has gone away. Your changes are still in memory, and the file on disk is the last good version. Fix the cause and press `w`. If you quit first, you lose them (section 17.3).

My settings are not remembered. Settings are saved when you choose them. If the settings file cannot be written, the program says `Could not save the settings:` and the reason. In demo mode (`--demo`) settings are never saved. A theme given with `-c` is for one run only, by design.

A report was not written. The export form stays open with the reason: the folder does not exist, a file of that name exists and you answered no, the name is one of the program's own data files, or the folder is not writable (section 15.1).

An import says the values are separated by semicolons. Your spreadsheet saved the file with semicolons (or tabs) instead of commas, as some do in some countries. Save it again as "CSV (comma delimited)" and import that (section 16.3).

An import adds nothing, and every row says "Quantity must be a whole number". A file of supply lines needs a Quantity column (section 16.3).

21.6 Other

The time or the date is wrong. The program asks the system's `date` command for your local time. Check the time zone of the system (`date`, and the `TZ` variable) and that `date` is on your `PATH`. If it cannot be run, the program falls back to UTC.

The program stops with "an internal error". It should never do that. Your data was saved after your last change. The details are in `inventory-central-crash.log` in the data folder (in demo mode, in a file of the system's temporary folder), and the message names the path.

21.7 Reporting a problem

Please send:

1. The output of `inventory --version`.
2. What you did, as exactly as you can remember, and what you expected.
3. The text of any message.
4. The crash log, if there is one.
5. If it concerns what a page shows, the text of the page: `inventory -d -s 120x34 -p PAGE`, with the same `-f` file, prints it. If you cannot share your data, `inventory --demo -d ...` with the `-k` keys that reproduce the problem is just as good, and carries no private data.

22. Limits and Things to Know

This is the honest list of what Inventory Central does not do, and where its edges are. Most of these are choices made to keep the program small, safe and dependency-free; none of them can lose or damage your records.

22.1 Size

- **How much data.** The tables are held in memory and every table is saved as a whole at each change. It is made for the records of one team or site: hundreds to a few thousand rows per table. It was tested with 5,000 items and 4,000 connections, which start in a fraction of a second. A data file may be at most 64 MB.
- **Imports** take at most 5,000 rows and 64 MB per file. Reading the preview of the largest file into a large table can take a few seconds, because every new row is checked against every row already there.
- **Field lengths** are fixed (Appendix B): for example an Asset Tag has 40 characters, a Hostname 253 and Notes 1000. The forms stop you typing more. Values that are longer in a file you edited by hand are kept as they are and never cut off, but such a record cannot be saved again until the value is shortened (since 0.1.16a; the form names the field).
- **Quantities** are whole numbers of up to nine digits (so up to 999,999,999). There are no negative quantities and no fractions: a roll of cable that is half used is a separate line, or a note.
- **Page size.** The program draws what fits in the window; below 50 columns by 12 rows it only asks for a larger window, and keys wait until it is larger (**q**, **Ctrl-C** and **Ctrl-L** still act). The window size is looked at about once a second, so a resize takes up to a second to show.
- **One search or filter at a time on a page.** A new one replaces the old one. The search, the filter and the sort order are remembered while the program runs, for each page and type, but are not saved to disk.

22.2 What it displays

- **Wide and combining characters are handled only in part.** Since 0.1.16a, characters that a terminal draws two columns wide (Chinese, Japanese and Korean text, most emoji) take two columns, and marks that combine with the letter before them and invisible format characters are not drawn, from a small table built into the program, so the rows of a table stay in line. But the columns are still sized by counting characters, so a value with wide characters is cut shorter than others; a combining mark is left out rather than combined (é typed as e and an accent shows as e); and the marks of scripts outside the table (most Indic scripts) still take a column each. The data itself is stored and exported unchanged.
- **Right-to-left text and control characters.** The program does not reorder text in Hebrew or Arabic; how it looks is up to your terminal. Since 0.1.16a, the invisible characters that change the direction of text, zero-width spaces and joiners, and a byte-order mark are removed from values as they are typed, pasted or read from a file, so **AT-1** with a hidden character in it is the same tag as **AT-1**. Control characters in a value are shown as spaces.

- **English only.** The program, its messages and its sort order are in English. Sorting uses a case-insensitive "natural" order in which AT-2 comes before AT-10, but it is not language-aware: accented letters are ordered by their code, not by the rules of a dictionary.
- **The summary's count column** is seven characters wide. A total of ten million units or more would touch the bar next to it; nothing is wrong with the number.
- **The mouse** uses the terminal's mouse reporting (the SGR form, with the older form understood too). Selecting text with the mouse while the program runs needs **Shift** in most terminals, or **Mouse** set to Off (section 14.6). A double click is two clicks: there is no timing, so the second click on a row opens it however slowly it comes, as long as nothing else happened in between.
- **The line under the table counts the menu, not the site.** The count in the page heading is the page's own; the line under the table counts everything the menu lists, at every site.

22.3 Keys

- **Ctrl-C** does not quit while a form is open, so that it cannot discard what you are typing. **Esc** cancels the form.
- **Pasting** relies on the terminal's bracketed paste, which the program switches on while it runs (since 0.1.16a) and every current terminal emulator supports: pasted text goes only into the field being changed. A terminal without it sends a paste as typed keys.
- **There is no undo**, apart from Decom: an item or a connection you delete goes to Decom first, and **R** there restores it. Deleting asks first, and only **y** deletes; an import shows its preview first; the automatic backup (section 17.3) is the safety net after a mistake.
- **Dates** are entered year first (2025-03-09) and shown the same way. There is one date field, the Purchase Date, and it holds a single day: warranty ends, install dates and the like belong in Notes.
- **One site per item, one location per item.** A bin on a shelf, or a desk in a closet, cannot be recorded as a location inside a location; give each its own name in the room.
- **One rack position per item.** An item in a rack fills a run of whole units from its Position up; half-depth or side-by-side mounting, zero-U PDUs at a position and blades in a chassis are not modelled (leave the Position empty for a zero-U PDU, as the demo does).
- **Connections name their devices by text.** A connection keeps a device name at each end: since 0.1.16a, a name that matches an item is stored as the item's Asset Tag (or its serial number when it has no tag). It does not follow a renamed item (its ports do). Checks reports the ends that no longer match.
- **There are no user-defined fields** and the columns cannot be reordered, renamed or hidden. Extra columns you add to a data file by hand are kept but not shown (section 17.2).

22.4 Files and systems

- **Operating systems.** Linux and macOS, and other systems with a POSIX shell and the standard `stty`, `date`, `sh`, `kill` and `ps` utilities. Native Windows is not supported; use WSL.
- **The Linux binary** that is built on a current Linux system runs on systems with a glibc of version 2.39 or newer (Ubuntu 24.04, Debian 13, Fedora 40 and later). On an older system, build from the source, which needs only Rust 1.97 or newer.

- **The lock cannot see into other computers.** It stops two copies on one machine. Since 0.1.16a it records the computer, its boot and its process-number space, so a lock made on another machine or in another container that shares the folder is never taken over; but the program cannot tell whether that copy is still running, and says so (section 17.4). A lock left from before this computer last started (after a power cut) is taken over by itself. Do not share live data files between two computers at once.
- **Replacing a file in place** (which is how a crash-proof save works) gives the new file your user as its owner and the file's usual permissions. Access control lists, extended attributes and other hard links to the old file are not carried over to the new one. If the data file is on a system where that matters, keep the data in a folder whose permissions you manage, or use a symbolic link, which the program follows and keeps.
- **The data files are not encrypted.** They are readable by you only (mode 0600) when the program creates them, and by nobody else unless you change that. If the disk itself must be protected, use an encrypted file system.
- **Network connections.** The program makes none. It does not discover devices, query switches or read port states; your connections are what you record.
- **Time zone and clock.** The time comes from the system's `date` utility and is read once a minute. It is shown only in the header and is never stored in a record.

22.5 What counts as a duplicate

Duplicates are detected only by the rules in section 16.2: the Asset Tag; the Serial Number together with the Manufacturer; the Cable ID; the same two ports; the same supply line; the same port name on one item; the same name on a Config list. Two items that differ only in a model number or a note are two items. If you want a different rule, the Checks page and a spreadsheet opened on an export are the way to look for near-duplicates.

22.6 What the Checks cannot know

The Checks compare your records with each other. They cannot know whether a cable really exists, whether a port is really in use, or whether the count of a supply is right. A device in a connection is matched to the inventory by its Asset Tag, then its Serial Number, by exact name (ignoring capitals); `SW-01` and `SW 01` are different devices (section 5.8).

Appendix A. Key Reference

Every key in Inventory Central, by where you are. The same list, in shorter form, is in the program under **?**. A key shown with a slash is a choice of keys that do the same thing. Capital letters need Shift.

A.1 On every page

Keys	Action
1 2 3 4 5 6 7 0	Open a top menu: Summary, Stock, Inventory, Connections, Decom, History, Config, Settings.
Tab / Right	Next top menu (wraps from Settings to Summary).
Shift-Tab / Left	Previous top menu.
v / V	Next or previous page of the menu: All, then each site. Summary: Summary, Checks. Config: Sites, Types.
] / [	Next or previous type: All types, each type on the page, then Supplies, Parts, Cables (Stock), Ports (Connections), Connections (Decom). Config: the lists of the page.
? / F1	Open the key reference.
b	Switch the sidebar off or on.
{ / }	Previous / next color theme (saved).
w / W	Save now.
Mouse	Click a top menu, a bar entry, a row (a second click opens it; with Shift, marks it), a heading (sort), a key hint, field, list entry or button; the wheel moves three lines.
Ctrl-L	Redraw the screen.
q / Ctrl-C	Quit. If the last save failed, asks first.

A.2 Every list

Inventory, Stock, Connections and its Ports, Decom, and the lists of Config.

Keys	Action
Up / k , Down / j	Move the highlighted row.
PgUp , PgDn	Move a page up or down.
Home / g , End / G	First or last row.
h / l	Move the highlight to the column on the left or right (the current column).
a	Add a row.
e / Enter	Edit the highlighted row.
c	Clone the highlighted row (the fields that must be unique are left empty).

Keys	Action
d / Delete	Items and connections: move the highlighted row (or every marked one) to Decom, after asking. On Decom, and on every other list: delete for good, after asking.
R	Decom: restore the row (or every marked one) as it was.
E	Change the highlighted cell only.
Shift-Down / J , Shift-Up / K	Mark the row and the next or previous one.
Insert , Shift-click	Mark or unmark one row.
Ctrl-A	Mark every row shown; again, unmark them.
M	Change fields of every marked row.
Space	View every field of the row.
o / s	Sort by the current column; again, to reverse.
0	Choose a sort column and direction.
Alt-1 to Alt-9	Sort by the 1st to 9th column from the left; again, to reverse.
r	Flip the sort direction.
/	Search every column.
f	Filter on the current column.
Esc	Unmark the marked rows; clear the search or filter.
x	Export to CSV or Markdown.
i	Import from a CSV file.
m	Items: the status form: Deployed, Reserved, Stock or Decommissioned, with a note.
t	Items: the next status at once (asks first when an item would leave a rack or cables).
C	Items: a quick clone, with no form.
B	Items: add items by scanning serial numbers or asset tags.
p	Items: make the item's ports.
P	Ports, Connections: make the ports that cables name but that are not on record, after asking.
+ / =	Supply lines: one more on hand.
- / _	Supply lines: one fewer on hand.

A.3 Summary, Checks and Settings

Keys	Action
Up / k , Down / j	Summary: scroll. Checks, Settings: move the highlight.

Keys	Action
PgUp , PgDn	Summary, Checks: a page at a time.
Home / g , End / G	First or last line, check or setting.
Enter / e	Checks: go to the record the check is about. Settings: open the menu for the setting.
Space	Settings: the same as Enter .

A.4 In a form

Forms are used for adding and editing, and also for sorting, searching, filtering, exporting and importing. A form for adding or editing a record moves and changes in two steps (section 3.3):

Keys	Action, in a record form
j / Down / Tab , k / Up / Shift-Tab	Moving: next or previous field (then the Save and Cancel buttons).
g / Home , G / End	Moving: first or last field.
Enter / e / Space	Moving: start changing the field (its value is selected; a menu or suggestions open).
Enter	Changing: keep the change and go back to moving.
Esc	Changing: put the field back. Moving: cancel the form.
Tab / Down , Shift-Tab / Up	Changing: keep the change and move on.
Left / Right	Moving, on a menu field: the previous or next choice.
Ctrl-S / F2	Save, from either.

The other keys below apply while changing a field, and in the other forms throughout:

Keys	Action
Tab / Down	Next field; after the last, the Save and Cancel buttons.
Shift-Tab / Up	Previous field.
Enter	On a menu field: open the menu. On a text field: go to the next field; from the last field, save. On a button: press it. In a one-line search or filter form: apply it.
Ctrl-S / F2	Save (or export, or preview, or apply, according to the form), from any field.
Esc	Close an open menu or suggestion list; otherwise cancel the form.
Left / Right	On a menu field: the previous or next choice. On a text field: move the cursor. On the buttons: Save or Cancel.
Home / Ctrl-A , End / Ctrl-E	Start or end of the text.
Backspace , Delete	Delete before or after the cursor. On a closed menu: empty the field.
Ctrl-U , Ctrl-K	Delete everything before or after the cursor.
Any character	On a text field: type it (replacing a selected value). On a menu field: open the menu and narrow it.
Ctrl-N / Ctrl-P	On a field with suggestions: show the list, then step to the next or previous suggestion.

A.5 In a menu or suggestion list

Keys	Action
Up , Down	Move the highlight.
PgUp , PgDn	Move eight entries at a time.
Home , End	First or last entry.
Enter	Take the highlighted entry and go to the next field (from the last field, save). In a list of suggestions with nothing highlighted: keep what you typed and go to the next field.
Tab	Take the highlighted entry (if any) and go to the next field.
Shift-Tab	Close the list and go to the previous field.
Ctrl-S / F2	Take the highlighted entry of a menu, and save.
Esc	Close the list.
Typing	Narrow the list to what contains the typed letters, any case; entries that start with them come first, and the match is underlined.

A.6 In other dialogs

Dialog	Keys
Splash screen	Any key continues (and is used up).
Key reference	Up / k , Down / j , PgUp , PgDn / Space , Home / g , End / G scroll; Esc , q , ? , F1 , Enter or Ctrl-C close.
View of a row	Up / k , Down / j , PgUp , PgDn scroll; e edits the row; Esc , q , Space , Enter or Ctrl-C close.
Confirmation	y / Y / Enter yes. n / N / Esc / q / Ctrl-C no. Other keys are ignored. A delete, and a move to Decom, take y only.
Scanning	Each scan is added at its own Enter (a scanner sends one). f then Enter , Enter on an empty scan, or Esc finishes.
Import preview	Enter / y imports. Esc / q / n / Ctrl-C cancels. Up / k , Down / j , PgUp , PgDn / Space , Home / g , End / G scroll.
Settings menu	Up / k , Down / j , Home / g , End / G move. Enter / Space applies. Esc / q / Ctrl-C cancels.

Appendix B. The Fields

Every field of the tables and the Config lists: its name in the form and as a heading in CSV files, its name as a table column (when that is shorter), its kind, the most characters it takes, and what it holds. Fields are listed in the order of the forms, which is also the order of a data file. The **Kind** is *text*, *date* (year first), *number* (a whole number), *menu* (one of a fixed list) or *text + list* (anything may be typed; a list of suggestions is offered). The lists of choices follow.

Inventory

Field	Column	Kind	Max	What it holds
Serial Number	Serial Number	text	64	The manufacturer's serial number. An Asset Tag or a Serial Number is required.
Component Type	Type	text + list	40	What kind of equipment it is. Pick from the list or type your own.
Asset Tag	Asset Tag	text	40	Your label on the equipment. Must be unique. Needs a tag or a serial number.
Manufacturer	Manufacturer	text + list	64	Who made it. Manufacturers from Config and ones already used are offered.
Model	Model	text + list	80	The model name or number. Models of the manufacturer, from Config, are offered.
Notes	Notes	text	1000	Anything worth remembering.
Hostname	Hostname	text	253	Network name, as sw-core-01. No two items share one.
Purchase Date	Purchase Date	date	16	Year first: 2025-03-09. Slashes and dots are accepted, and so is the word today.
PO	PO	text	40	The purchase order number.
Status	Status	menu	16	Deployed: in service. Reserved, Stock: on the shelf. Decommissioned: retired.
Site	Site	menu	64	The site it is at. Sites are kept under Config (6).
Room	Room	menu	64	The room, from the rooms of the Site. Choose the Site first.
Location	Location	menu	64	A rack, shelf, desk, bin or closet in the room. Shelves and bins hold many items.
Position (U)	Pos	number	4	The lowest rack unit it fills, counted from the bottom: 12. Needs a Rack.
Height (U)	U	number	4	How many rack units tall it is: 1 for a 1U switch. Empty counts as 1.
CPU	CPU	text + list	80	The processor, for example 2 x Xeon Silver 4410Y. CPUs from Config are offered.

Field	Column	Kind	Max	What it holds
Memory	Memory	text + list	60	The memory, for example 128 GB DDR5. Memory from Config is offered.
Project	Project	text + list	80	The project the item belongs to, if any. Projects already used are offered.

Connections

Field	Column	Kind	Max	What it holds
Cable ID	Cable ID	text	40	The cable's label. Must be unique. Needed unless a device is given at either end.
Status	Status	menu	16	Whether the connection is in use.
Type	Type	text + list	24	The kind of connection: copper, fiber, direct-attach, console, and so on.
Speed	Speed	text + list	16	The link speed, for example 10G.
A Device	A Device	text + list	64	The A-end device. Picking from the inventory fills in location and serial. (Suggestions: the inventory.)
A Port	A Port	text + list	40	The A end's port, one of the device's ports (Ctrl-N lists them): Gi1/0/24.
A Rack/Location	A Location	text + list	100	Rack, unit or room at the A end.
A Serial	A Serial	text	64	Serial number of the equipment (or module) at the A end.
B Device	B Device	text + list	64	The B-end device. Picking from the inventory fills in location and serial. (Suggestions: the inventory.)
B Port	B Port	text + list	40	The B end's port, one of the device's ports (Ctrl-N lists them).
B Rack/Location	B Location	text + list	100	Rack, unit or room at the B end.
B Serial	B Serial	text	64	Serial number of the equipment (or module) at the B end.
Cable	Cable	text + list	40	The cable grade: Cat6a, OM4, OS2, and so on.
Color	Color	text + list	20	The color of the cable or jacket.
Length	Length	text + list	20	How long the cable is, with its unit: 3 m, 10 ft.
VLAN	VLAN	text	40	The VLAN or network the link carries, or 'trunk'.
Optic / SFP	Optic	text + list	64	The transceiver, for example SFP-10G-SR. Modules from your supplies are offered.
Notes	Notes	text	1000	Anything worth remembering.

Supplies

Field	Column	Kind	Max	What it holds
Item	Item	text	80	What it is, in your own words, for example 'LC-LC duplex patch cable'. Required.
Category	Category	text + list	40	The kind of supply. Pick from the list or type your own.
Part / Model No.	Part No.	text	64	The manufacturer's part or model number.
Manufacturer	Manufacturer	text + list	64	Who made it. Manufacturers from Config and ones already used are offered.
Type / Spec	Spec	text + list	80	The specification: fiber grade, copper category, or optic standard.
Connector	Connector	text + list	40	The connectors, for example LC/LC, or the module form factor.
Color	Color	text + list	20	The color of the cable or jacket.
Length	Length	text + list	20	The length, with its unit: 3 m, 10 ft.
Speed	Speed	text + list	16	The speed rating, for example 10G.
Quantity	Qty	number	9	How many on hand: a whole number. On the list, + and - add or remove one.
Minimum Qty	Min	number	9	Optional. The Checks page warns when the quantity is at or below this.
Location	Location	text + list	100	Where it is stored: a room, shelf or bin.
Notes	Notes	text	1000	Anything worth remembering.
Class	Class	menu	10	Supply, Part or Cable: its type on Stock. Blank: from the category.

Ports

Field	Column	Kind	Max	What it holds
Device	Device	text + list	40	The item the port is on: its asset tag. It must be in the Inventory. (Suggestions: the inventory.)
Port	Port	text	40	The port's name as printed on the device: Gi1/0/1, eth0, PSU1, Outlet 3. Unique on its device.
Kind	Kind	menu	16	What the port is for: Network, Power in (an inlet), Power out (an outlet), Console, Management, Other.
Port Type	Type	text + list	40	The connector or module: from the Port types list under Config, or your own.
Speed	Speed	text + list	12	The port's speed, for example 1G or 10G. Empty for power ports.
Notes	Notes	text	1000	Anything worth remembering.

Sites

Field	Column	Kind	Max	What it holds
Name	Name	text	80	The site: a building or a campus. Must be unique.
Address	Address	text	200	Where it is, if you want it on record.
Notes	Notes	text	1000	Anything worth remembering.

Rooms

Field	Column	Kind	Max	What it holds
Site	Site	menu	64	The site the room is in. Add the site first if it is not listed.
Name	Name	text	80	The room: Server room, IDF 2, Store room. Unique within its site.
Notes	Notes	text	1000	Anything worth remembering.
Floor	Floor	menu	64	The floor it is on, from the site's floors; may be left empty.

Locations

Field	Column	Kind	Max	What it holds
Site	Site	menu	64	The site the location is in.
Room	Room	menu	64	The room the location is in. Choose the Site first.
Name	Name	text	80	Its name: A1, Shelf 3, Desk 12, Bin A-12, Closet. Unique within its room.
Units	Units	number	4	Racks only: how many rack units it has, 42. Equipment is checked against it.
Notes	Notes	text	1000	Anything worth remembering.
Kind	Kind	menu	10	Rack, Shelf, Desk, Bin or Closet. Only a rack has units; any number of items share the others.

Manufacturers

Field	Column	Kind	Max	What it holds
Name	Name	text	80	The manufacturer. Must be unique.
Notes	Notes	text	1000	Anything worth remembering.

Models

Field	Column	Kind	Max	What it holds
Manufacturer	Manufacturer	text + list	64	Who makes it. Manufacturers from the list above are offered.
Model	Model	text	80	The model name or number. Unique for its manufacturer.
Type	Type	text + list	40	What kind of equipment it is. Fills in the type of items of this model.
Height (U)	U	number	4	How many rack units tall it is. Fills in the height of items of this model.
Ports	Ports	number	4	Network ports on a new unit: offered when an item of this model is added.
Power Inputs	Power	number	3	Power inputs on a new unit: offered when an item of this model is added.
Notes	Notes	text	1000	Anything worth remembering.
Outlets	Outlets	number	3	Power outlets on a new unit (a PDU, a UPS): offered when an item of this model is added.

Types

Field	Column	Kind	Max	What it holds
Name	Name	text	80	A kind of equipment: Server, Switch, UPS. Must be unique.
Notes	Notes	text	1000	Anything worth remembering.

CPUs

Field	Column	Kind	Max	What it holds
Name	Name	text	80	A processor, for example Xeon Silver 4410Y. Must be unique.
Notes	Notes	text	1000	Anything worth remembering.

Memory

Field	Column	Kind	Max	What it holds
Name	Name	text	80	A memory size or kind, for example 128 GB DDR5. Must be unique.
Notes	Notes	text	1000	Anything worth remembering.

Port types

Field	Column	Kind	Max	What it holds
Name	Name	text	80	A kind of port: RJ45 1G, SFP+, C13 power in. Must be unique.
Class	Class	menu	16	What the port is for. Power ports are connected with power cables.
Speed	Speed	text + list	16	The speed of a network port, for example 10G. Empty for other ports.
Notes	Notes	text	1000	Anything worth remembering.

Floors

Field	Column	Kind	Max	What it holds
Site	Site	menu	64	The site the floor is in.
Name	Name	text	80	The floor: Ground, 1, 2, Basement, Mezzanine. Unique within its site.
Notes	Notes	text	1000	Anything worth remembering.

B.1 Choices and suggestions

The standard choices of the fields that have them. A menu accepts only these (or empty, where it allows that); a text field with suggestions offers them first, followed by the values already used in that column.

Field	Choices
Inventory: Component Type	Server, Switch, Router, Firewall, Laptop, Desktop, Workstation, Monitor, Storage, UPS, PDU, Access Point, Patch Panel, Printer, Phone, Tablet, Other
Inventory: Status	Deployed, Reserved, Stock, Decommissioned
Connections: Status	Active, Planned, Spare, Disabled, Decommissioned
Connections: Type	Copper, Fiber MM, Fiber SM, DAC, AOC, Console, Stack, Power, Other
Connections: Speed	10M, 100M, 1G, 2.5G, 5G, 10G, 25G, 40G, 50G, 100G, 200G, 400G
Connections: Cable	Cat5e, Cat6, Cat6a, Cat7, Cat8, OM1, OM2, OM3, OM4, OM5, OS1, OS2, Twinax, Coax
Connections: Color	Blue, Yellow, Orange, Aqua, Lime, Green, Red, Black, White, Grey, Purple, Pink, Violet
Connections: Length	0.5 m, 1 m, 2 m, 3 m, 5 m, 7 m, 10 m, 15 m, 20 m, 30 m, 50 m, 100 m, 1 ft, 3 ft, 5 ft, 7 ft, 10 ft, 15 ft, 25 ft
Supplies: Category	Fiber patch cable, Copper patch cable, SFP / SFP+ module, QSFP module, DAC / AOC cable, Bulk fiber, Connector / adapter, Power cable, Rack hardware, Spare part, Labels and ties, Tools, Other

Field	Choices
Supplies: Type: Spec	OM3 duplex, OM4 duplex, OM5 duplex, OS2 duplex, OS2 simplex, Cat5e UTP, Cat6 UTP, Cat6a UTP, Cat6a shielded, 1000BASE-T, 1000BASE-SX, 1000BASE-LX, 10GBASE-SR, 10GBASE-LR, 25GBASE-SR, 40GBASE-SR4, 100GBASE-SR4, 100GBASE-LR4
Supplies: Connector	LC/LC, LC/SC, SC/SC, LC/ST, ST/ST, MPO/MTP, RJ45, SFP, SFP+, SFP28, QSFP+, QSFP28, Bare / pigtail
Supplies: Color	Blue, Yellow, Orange, Aqua, Lime, Green, Red, Black, White, Grey, Purple, Pink, Violet
Supplies: Length	0.5 m, 1 m, 2 m, 3 m, 5 m, 7 m, 10 m, 15 m, 20 m, 30 m, 50 m, 100 m, 1 ft, 3 ft, 5 ft, 7 ft, 10 ft, 15 ft, 25 ft
Supplies: Speed	10M, 100M, 1G, 2.5G, 5G, 10G, 25G, 40G, 50G, 100G, 200G, 400G
Supplies: Class	Supply, Part, Cable
Ports: Kind	Network, Power in, Power out, Console, Management, Other
Ports: Speed	10M, 100M, 1G, 2.5G, 5G, 10G, 25G, 40G, 50G, 100G, 200G, 400G
Locations: Kind	Rack, Shelf, Desk, Bin, Closet
Models: Type	Server, Switch, Router, Firewall, Laptop, Desktop, Workstation, Monitor, Storage, UPS, PDU, Access Point, Patch Panel, Printer, Phone, Tablet, Other
Port types: Class	Network, Power in, Power out, Console, Management, Other
Port types: Speed	10M, 100M, 1G, 2.5G, 5G, 10G, 25G, 40G, 50G, 100G, 200G, 400G

Appendix C. CSV Columns and Alternative Headings

An exported CSV file has one heading for each field, exactly as in Appendix B. When a file is imported (or read as a data file), a heading is understood if it matches, ignoring capitals, spaces and punctuation, the field's name, its short column name, or one of the alternative names below (shown in the form in which they are compared: lower case, with no spaces or punctuation). `Asset Tag`, `asset_tag`, `ASSET-TAG` and `assettag` are all the same heading. If two columns of a file mean the same field, the first is used and the second is treated as an extra column and ignored.

The first row is taken to be a heading row if it contains at least two names the program knows, or (since 0.1.16a) one cell that is exactly a field's name, short column name or alternative heading (ignoring capitals and surrounding spaces). An alternative heading that stands in its own field's column position is not counted. Otherwise the file is taken to have no headings and its columns to be in the order of the form.

C.1 Inventory

Field	Alternative headings
Serial Number	serialnumber, serial, sn, serialno, serialnum
Component Type	componenttype, type, category, devicetype, assettype
Asset Tag	assettag, tag, assetid, assetnumber, inventorynumber
Manufacturer	manufacturer, make, vendor, brand, mfr
Model	model, modelnumber, modelname
Notes	notes, note, comments, comment, description
Hostname	hostname, host, fqdn, dnsname, computername, devicename
Purchase Date	purchasedate, date, purchased, dateofpurchase, bought
PO	po, ponumber, purchaseorder, pono
Status	status, state, deployment, stockstatus, inservice
Site	site, sitename
Room	room, roomname
Location	rack, rackname, racknumber, place
Position (U)	position, rackposition, ustart, rackunit, startunit, uposition
Height (U)	height, rackunits, uheight, units, heightu
CPU	cpu, processor, processors
Memory	memory, ram, mem
Project	project, projectname, projectcode, job

C.2 Connections

Field	Alternative headings
Cable ID	cableid, cable, id, label, cablelabel, linkid, cablenumber
Status	status, state
Type	type, connectiontype, linktype, medium, media
Speed	speed, linkspeed, rate, bandwidth
A Device	adevice, devicea, fromdevice, sourcedevice, sidea, aend, adevicename, switch, switchname
A Port	aport, porta, fromport, sourceport, adeviceport
A Rack/Location	alocation, locationa, arack, racka, aracklocation, fromlocation
A Serial	aserial, serialaend, aserialnumber, serialnumbera, sna
B Device	bdevice, deviceb, todevice, destinationdevice, sideb, bend, bdevicename
B Port	bport, portb, toport, destinationport, bdeviceport
B Rack/Location	blocation, locationb, brack, rackb, bracklocation, tolocation
B Serial	bserial, serialbend, bserialnumber, serialnumberb, snb
Cable	cabletype, cablegrade, cablemodel, grade, category, fibertype
Color	color, colour, cablecolor, jacketcolor
Length	length, cablelength, len
VLAN	vlan, vlanid, network, vlans
Optic / SFP	optic, optics, optic/sfp, opticsfp, sfp, transceiver, module, sfpmodel
Notes	notes, note, comments, comment, description

C.3 Supplies

Field	Alternative headings
Item	item, name, description, supply, product, itemname
Category	category, type, kind, group
Part / Model No.	partno, partnumber, part, modelnumber, model, sku, mpn, modelno
Manufacturer	manufacturer, make, vendor, brand, mfr, supplier
Type / Spec	spec, typespec, specification, fibertype, standard, type
Connector	connector, connectors, connectortype, ends, formfactor
Color	color, colour, jacketcolor
Length	length, len, size
Speed	speed, rate, rating, bandwidth
Quantity	quantity, qty, count, onhand, stock, amount

Field	Alternative headings
Minimum Qty	minqty, minimumqty, minimum, min, reorderlevel, reorderat, reorderpoint, minimumquantity
Location	location, storage, shelf, bin, room
Notes	notes, note, comments, comment
Class	class, supplyclass, linetype

C.4 Ports

Field	Alternative headings
Device	device, item, assettag, tag, host
Port	port, portname, name, interface
Kind	kind, class, porttype2, purpose
Port Type	porttype, type, connector, media
Speed	speed, rate, bandwidth
Notes	notes, note, comments, comment, description

C.5 Sites

Field	Alternative headings
Name	name, title
Address	address, street, where
Notes	notes, note, comments, comment, description

C.6 Rooms

Field	Alternative headings
Site	site, sitename
Name	name, title
Notes	notes, note, comments, comment, description
Floor	floor, level, storey, story

C.7 Locations

Field	Alternative headings
Site	site, sitename
Room	room, roomname
Name	name, title
Units	units, rackunits, height, size, u
Notes	notes, note, comments, comment, description
Kind	kind, locationkind, locationtype, type

C.8 Manufacturers

Field	Alternative headings
Name	name, title
Notes	notes, note, comments, comment, description

C.9 Models

Field	Alternative headings
Manufacturer	manufacturer, make, vendor, brand, mfr
Model	model, modelnumber, modelname, name
Type	type, devicetype, category
Height (U)	height, rackunits, uheight, units, heightu
Ports	ports, portcount, networkports, portqty
Power Inputs	powerinputs, power, powerin, psu, psus, powersupplies
Notes	notes, note, comments, comment, description
Outlets	outlets, outletcount, poweroutlets, receptacles

C.10 Types

Field	Alternative headings
Name	name, title
Notes	notes, note, comments, comment, description

C.11 CPUs

Field	Alternative headings
Name	name, title
Notes	notes, note, comments, comment, description

C.12 Memory

Field	Alternative headings
Name	name, title
Notes	notes, note, comments, comment, description

C.13 Port types

Field	Alternative headings
Name	name, title
Class	class, kind, use, purpose, category
Speed	speed, rate, bandwidth
Notes	notes, note, comments, comment, description

C.14 Floors

Field	Alternative headings
Site	site, sitename
Name	name, title
Notes	notes, note, comments, comment, description

Appendix D. If You Used an Earlier Version

Inventory Central replaces the earlier C and ncurses program, `inventory`. Its data file opens as it is and its sorting, searching and filtering work as they did. Some keys changed on the way, first in 0.1a, so that all the programs of the Electric Shepherd family answer to the same keys, and again in 0.1.1a, to follow vi. This appendix lists everything that is not the same, so that nothing about it surprises you.

D.1 Keys that changed

C version, or 0.1a	Now	Why
h opened the key reference (0.1a)	? or F1 . h and l move the current column.	vi keys: h left, l right, as j and k are down and up.
[and] moved the current column (0.1a); h , l , Left and Right did in the C version	h and l . [and] step through the types of a menu (since 0.1.17a; before, they changed the page).	One pair of keys for columns.
1 to 9 sorted by the column of that number (C version)	1 to 7 and 0 open a top menu. Alt-1 to Alt-9 sort by the column of that number, counting from the left.	The digits are the menu keys.
o opened the sort dialog and s sorted by the current column (0.1a)	o sorts by the current column (and s still does); 0 opens the sort dialog; r flips the direction.	The quick sort is the common one.
v showed a row in full (0.1a)	Space shows it. v and V step through the views of a page.	Pages now have views.
3 was the Supplies page (0.1a), then Stock	2 is Stock; the Supplies, Parts and Cables are its last types (2 then [). The settings word <code>supplies</code> and <code>-p supplies</code> still work.	Spare equipment and supplies are both stock.
1 Inventory, 2 Connections, 4 Summary, 5 Checks, 6 Config, 7 History (0.1.16a)	1 Summary (with Checks, v), 2 Stock, 3 Inventory, 4 Connections, 5 Decom, 6 History, 7 Config (since 0.1.17a).	The menus run from the overview to the details.
v stepped through all the views of a page, sites, Ports and lists alike (0.1.16a)	v steps the sites (or Summary's and Config's two pages), 1 the types and lists. The Ports are a type of Connections.	Two bars, one for where and one for what.
d deleted an item or a connection (0.1.16a)	It moves it to Decom (5), where R restores it and d deletes it for good.	Nothing is lost by one key.
Enter on a menu only opened it (0.1a)	A second Enter takes the choice and moves on, and Enter on the last field saves. Forms also end with Save and Cancel buttons.	A form can be filled in with Enter alone.

C version, or 0.1a	Now	Why
Q (capital) quit (C version)	Only q and Ctrl-C quit.	A capital letter that only looks like another key is easy to press by accident.
On the Type field, PgUp and PgDn cycled through the types (C version)	Type is a field with suggestions: start typing, or press Enter or Ctrl-N to open the list.	The same menus and suggestions work in every field that has them.

Everything else is the same: **a** add, **e** and **Enter** edit, **c** clone, **d** and **Delete** delete, **/** search, **f** filter, **Esc** clear, **w** save now, **Ctrl-L** redraw, and in a form **Tab**, **Up**, **Down**, **Ctrl-S**, **F2**, **Esc**, **Ctrl-A**, **Ctrl-E**, **Ctrl-U** and **Ctrl-K**.

D.2 What else is different

- **The table starts with the Type.** The columns are now Type, Model, Serial Number, Asset Tag, Status, Room, Manufacturer, Site, Location, Position, Height, Hostname, CPU, Memory, Purchase Date, PO, Project and Notes. The **form** and the **data file** keep the earlier order, Serial Number first, with the newer fields after the first nine, so a file written by an older program is read as it is.
- **The data file** is still `inventory.csv`; the other tables and the Config lists are kept in files beside it (section 17.1). If the standard folder holds no data and an `inventory.csv` is found in your home folder, the program uses that file, and says so. Your old file is not changed except by what you do in the program, and it gains the usual backup.
- **A new standard place for the data**, `~/.local/share/inventory-central/`, for new installs. Use `-f` or `INVENTORY_FILE` to keep the file where it is, as before.
- **Longer values are never cut.** The C version silently shortened a value that was longer than its field when it read a file, and the shortened text was written back at the next save. Inventory Central keeps what is in the file.
- **Saving.** Files are written atomically and keep their permissions, a symbolic link is followed rather than replaced, a lock stops two copies from writing the same file, and the first save of a session keeps a `.bak` copy (Chapter 17).
- **Quit.** The program no longer asks when you quit: there is never anything unsaved. It asks only if the last save failed.
- **Building.** No C compiler, `make` or `ncurses` is needed. `cargo build --release` builds it, with only the Rust standard library (section 1.3).

Appendix E. Release Notes

Every build raises the version. The file `CHANGELOG.md` in the source has the full notes of each.

E.1 Version 0.1.17a, October 2026

The working-with-many-items release, and a new way round the screen.

- **Top menu, page, type** (section 2.2). Eight top menus in a new order: **1** Summary (with Checks on its second page), **2** Stock, **3** Inventory, **4** Connections, **5** Decom, **6** History, **7** Config, **0** Settings. Under a menu, a site bar (**v**) and a type bar (**1**), and a page heading that says where you are: `Inventory > HQ > Switch`.
- **Each item is listed once**: in service on the Inventory, in stock or reserved on Stock, decommissioned on Decom. **Reserved** and **Decommissioned** are new statuses; **m** is a status form with a note and **t** steps the status (section 5.3).
- **Decom** (section 5.7): **d** on an item or a connection moves it to Decom; there **R** restores it as it was and **d** deletes it for good. The web page does the same.
- **Many items at once** (section 5.5): mark rows with **Shift-Down** or **J**; **d**, **m** and **M** (change fields of every marked row) act on all of them; **B** adds items by scanning with a barcode scanner; **C** clones at once; **E** changes one cell.
- **Locations and floors** (section 9.1): Racks became Locations, each a rack, shelf, desk, bin or closet; only racks have units, and shelves and bins are shared. A new Floors list: `Site > Floor > Room`. Config has two pages, Sites and Types.
- **Supply lines have a Class**: Supply, Part or Cable, each a type on Stock; **Parts** are new (Chapter 8).
- **-** and **n/a** in the Asset Tag or Serial Number mean "none"; item columns start with Type, Model, Serial Number, Asset Tag, Status, Room.
- Four more themes, Midnight, Orchid, Sandstone and the light **Paper**; "Colors" is now "Theme". The sheep splash is redrawn.

E.2 Version 0.1.16a, October 23, 2026

A security and reliability release, from a full design review of the programs, the server, the web page and the scripts.

- **Your typing is safer**. Pasted text goes only into the field being changed and never runs as keys (section 3.3); a delete takes **y** only, and **Enter** keeps the record; keys wait while the window is too small.
- **Your files are safer**. The files one action changes are saved together, and a save cut short is finished at the next start (section 17.3); spreadsheet formulas are guarded in the data files too; a file with a few bytes that are not UTF-8 keeps the rest of its accents; a hand-made file with one exact column name is read with its heading row; the change history keeps the bytes it cuts; the settings file keeps your comments and unknown lines.
- **The lock knows where it was made** (section 17.4): a lock from another computer or container is never taken over, and the program never runs on a server's folder.

- **Working with the server:** an edit saved after someone else changed the row is refused and the form reopens with your changes, naming what they changed; **Ctrl-S** again saves on top (section 18.4). A server that stops answering never freezes the screen, and the program connects again by itself (section 18.5).
- **Connections and ports name an item by its Asset Tag** (or serial number), whichever you type; a tag may not be another item's serial number (section 5.2). Every save refuses a value longer than its field allows.
- **The server** has limits on connections and time, and an optional session key for the SSH helper (-k); the web page needs a key that only nginx has (-K; run `inventoryd setup -w` again after upgrading). Setup leaves the `users`, `roles` and SSH files root's own and keeps its record outside the data folder. The certificate authority is limited to the web page's host names (`inventory-ca init` takes them).
- **The web page** keeps what you typed on a conflict, asks before **Esc** discards a changed form, sends one change at a time, shows an **Offline** badge on a phone, and closes for a person whose role is taken away.

E.3 Version 0.1.15a

- **The Web Page User Guide**, a separate PDF for the people who use the page: certificates, finding your way, looking things up, changing data, moving equipment, printing, phones, and what messages mean.
- Fixed on the web page: the History page showed "[object HTMLModElement]" instead of a change's fields; the Overview's recent changes listed "old items"; new records started blank where the terminal program fills in values (a connection Active, a rack 42U); choosing a connection's device now fills in that end's place and serial, as in the terminal program.

E.4 Version 0.1.14a

- **Moving equipment on the web page** (section 19.4): drag a device to another unit or rack; **Up one unit** and **Down one unit** in its details, or **Shift-Up** and **Shift-Down** on the drawing.
- **The port map** (section 19.5): an item's ports as tiles; a free port starts a new connection from it.
- **Printing** (section 19.6): Inventory, Connections, Stock and Racks print cleanly, with who printed them and when.
- **The web page on macOS** with Homebrew's nginx (section 19.9).
- Fixed: the details panel of the web page showed the word "null" above its buttons.

E.5 Version 0.1.13a

- **Record forms move and change in two steps** (section 3.3): **j** and **k** move between the fields, **Enter** starts changing one, **Enter** again keeps the change and **Esc** puts it back. Search, filter, sort, export and import forms are as before.
- **A narrowed list highlights the first entry that starts with what you typed**, so **Enter** takes it.
- **Hostname replaces Location** on items: a network name, kept in lower case, unique; the Checks page finds duplicates and names that are not hostnames. **Project** is a new field at the end. Old

locations are kept: a file or a server history from before reads each location into the item's Notes as `Location: ....`

- **The *Server Setup Guide***, a separate PDF beside this manual, walks an administrator through setting up the server, SSH, the certificate authority and nginx, keeping it running, what goes wrong, and taking it down.
- Teardown and the certificate authority script are sturdier: teardown carries on past a removal step that was already done by hand, trusts only a root-owned `setup.manifest`, and `inventory-ca -k` accepts a relative path.

E.6 Version 0.1.12a

- **Changing data on the web page** (section 19.3): editors add, change and delete records, and count supplies up and down, held to the terminal program's rules; refusals are shown at the field; renames carry through; conflicts show the record as it now is.
- **Closing the web page** (section 19.7): an admin can close and reopen it from the Settings page of the terminal program.
- Run `inventoryd setup -w` again after upgrading, so that nginx passes changes to the server.

E.7 Version 0.1.11a

- **The web page** (Chapter 19): the racks drawn as they stand, the inventory with each item's ports and connections, the cabling, the supplies against their minimums, the checks and the history, in nine looks, on a desk or a phone; read-only, following the server live. People get in with a certificate from your own authority and a role in the `users` file.
- `inventoryd -W` answers the page on a socket that only nginx reaches; `inventoryd setup -w HOST` writes the nginx site, and teardown removes it.
- `inventory-ca publish` puts the CA certificate and revocation list in place so that nginx always notices a new list.

E.8 Version 0.1.10a

- **The server checks the data** (section 18.13): every change is held to the same field rules as the forms, judged on the data as it would be after the whole change; only new problems are refused.
- `inventoryd setup` (section 18.14): the service account, the data folder, an optional import, the `users`, `roles` and `authorized_keys` files, and a locked-down systemd unit (or a launchd job on macOS); `-n` shows every step first, and everything made is listed for teardown.
- `inventoryd teardown` (section 18.15): saves the data as the terminal program's files and proves the copy byte for byte before it removes anything; never deletes data.
- `inventory-ca` and `CA-SETUP.md` (section 18.16): the certificate authority for the coming web page.

E.9 Version 0.1.9a

- **SSH access** (section 18.9): `inventory -R user@host` reaches the server through SSH, where the key's forced command is `inventoryd-session`; the server records the name the key gives, which the person cannot change.
- **Roles** (section 18.10): viewer, editor, admin, from the server folder's `users` file or from groups (`id -Gn`, so directory groups through `sssd`); read at every session; disabled and unknown people refused, and every refusal recorded. Viewers can look, not change; only admins see the audit events.
- **Idle limits** for SSH sessions: 15 minutes, 10 for admins (`-T`). `-G` opens the session socket to the folder's group.

E.10 Version 0.1.8a

- `inventoryd`, **the server** (Chapter 18): holds the data in its journal, numbered rows with versions; `-i` fills it from your CSV files, `-e` writes them back; one server per folder, a private folder and socket, no network port, nothing installed or started for you.
- `inventory -S SOCKET` works on the server's data: several people at once, each seeing the others' changes within a second; a change to a row someone else changed first is refused and the row shown as it now is.
- The History page shows the server's history, with who connected.

E.11 Version 0.1.7a

- **The change history** (Chapter 13): every add, change and delete, and every export, is recorded in `inventory.journal` beside your data, written before the data files are saved; one action is one change set.
- **The History page** (**7**): the history as a list, newest first, with who, when and what changed. It cannot be edited; it can be sorted, searched, filtered and exported.
- Under the surface: the stock **+**/**-** rule and deleting (with an item's ports, and the guard on sites, rooms and racks in use) are now library functions, ready for the shared server of a later version.

E.12 Version 0.1.6a

- **The mouse** (section 3.6): click pages, views, rows (a second click opens), column headings (sort), key hints, form fields, menu entries and the form buttons; the wheel scrolls. **Mouse** on the Settings page switches it off.
- **{ and }** step through the color looks, which are now listed from cool to warm.
- **Samhain**, a Halloween look: pumpkin and candle gold on near-black.
- **The current column is shown in full**; long headings use short forms (Maker, Serial, Purchased) before they are cut, and cut text ends in
- This manual, updated for the above.

E.13 Version 0.1.5a

- **Connections are made on ports.** At an end whose device has ports on record, the port must be one of them; the Port field lists the device's ports and says which are taken (section 7.2).
- **Power cords.** A Power cable joins a Power in port to a Power out port; a data cable cannot use a power port (section 7.4). The demo has five.
- **Headings on site views:** room, rack and device on Connections, room and rack on the Inventory.
- **P** makes the ports that existing cables name but that are not on record (section 6.4).
- Checks for a cable port that is not on record and for a port of the wrong kind. An item at a site no longer needs a Location as well; the Summary counts items by place.
- This manual, rewritten for 0.1.1a to 0.1.5a.

E.14 Version 0.1.4a

- **Ports** (Chapter 6): a table of every item's ports, made with the item from its model's counts and a naming pattern, or with **p**; they follow a renamed item and go with a deleted one. Models gain an Outlets count.
- **Site views** on Inventory and Connections (**v**).

E.15 Version 0.1.3a

- Menus and suggestion lists put the choices that start with what you typed first, underline the match and show the typed letters. **b sidebar** is listed in the key hints.

E.16 Version 0.1.2a

- **Config page** (Chapter 9): sites, rooms, racks, manufacturers, models, types, CPUs, memory and port types. Items gain Site, Room, Rack, Position, Height, CPU and Memory, with rack-unit rules; renaming a place carries through, and a place in use cannot be deleted. Imports add the places they name.
- **Save and Cancel buttons** in every form.
- **Sidebar switch** (**b**) and **six muted dark looks**: Harbor, Dusk, Autumn, Ember, Spring and Winter.

E.17 Version 0.1.1a

- vi keys (**h** **l** columns, **[** **]** pages), **?** and **F1** for help, **o** sorts the current column and **O** opens the sort dialog.
- **Stock page** with Items, Supplies and Cables; every item is Deployed or Stock, and **m** marks it.
- Forms can be filled in with **Enter** alone. The terms of use are built in: `inventory --licence`.

E.18 Version 0.1a

The first alpha of the Rust edition. It replaced the C and ncurses program with one written in Rust with only the standard library; added the Connections and Supplies pages, the Summary and the Checks, CSV import with a preview and CSV and Markdown export, six color looks, a Settings page, a lock, automatic backups and atomic saves; and kept the Inventory page, its data file and its sorting, searching and filtering.

Known limits are listed in Chapter 22.