

FiberQ — User Guide

Version 1.0.0 · MIT licence · AutoCAD 2015 and newer, 64-bit Windows

FiberQ draws single-line schematics of FTTH and GPON access networks. Every fiber carries its own data inside the drawing, so the schematic is also the source for the bill of materials and the optical loss budget.

What FiberQ does	Cables
Requirements	Housings
Installing	Splitters
If the tab does not appear	Closures
Ribbon, menu, palette	Working with fibers
Settings and rules	Reports and checks
Language	Your data
Troubleshooting	Licence and support

What FiberQ does

Cables	Duct, aerial, micro duct, indoor. Fiber and tube colours from TIA-598 or IEC 60304, or your own standard via FQCC .
Housings	ODF modules, patch panels, FDH cabinets, MDU cabinets, FAT boxes — from 1 to 576 fibers depending on the type . Fibers land on real ports. The catalogue is a plain text file you can extend.
Closures	Cut a fiber, add a splice, mark a fiber as through, join two cables fiber by fiber or a whole group at once, add a slack loop.
Splitters	1:2 to 1:64 from the ribbon and palette; any ratio from FQSPLITTER . Standalone, or inside an FDH, MDU or FAT.
Terminals	OLT and ONT symbols with PON ports and drop cables.
Per-fiber data	Label, purpose, single-mode or multi-mode, status (designed or as-built), free-text description, real cable length in metres.
Reports	Bill of materials to the command line or into the drawing, optical loss budget likewise, symbol legend, fiber route listing, drawing check.

Four ways in: the **FiberQ** ribbon tab, the **FiberQ** pull-down menu (visible only with `MENUBAR = 1`), the palette (`FIBERQ` or `FQ`), and the command line.

Requirements

Operating system	64-bit Windows. There is no Mac build.
Product	AutoCAD and the verticals: Map 3D, Civil 3D, Architecture, Mechanical, Electrical.
Release	AutoCAD 2015 (R20.0) or newer. No upper limit.
Not supported	AutoCAD LT — it lacks the Visual LISP functions FiberQ needs.
Installer	Windows PowerShell 5.1, shipped with Windows 10 and 11.

FiberQ is written entirely in AutoLISP and Visual LISP, with DCL dialogs built at run time. No .NET, no ObjectARX. Developed and tested on **AutoCAD Map 3D 2021 (R24.0)** under Windows 11.

The installer creates no registry entries of its own. FiberQ does keep two things in your AutoCAD profile, which lives in the registry: the interface language, and AutoCAD's record that the FIBERQ customization group is loaded. Both survive an uninstall — see [Uninstalling](#).

If `fq-kucista.lsp` is missing, everything else still works, but there are no FAT boxes, no patch panels and no MDU cabinets.

`FQVERSION` (`FQVERZIJA`) **is the first thing to run if anything looks wrong**. It prints the version, which parts loaded, the host product and release, the active colour code standard, and where your catalogues were read from and will be written.

`FQHELP` (`FQPOMOC`) **opens this guide**. The PDF is installed inside `FiberQ.bundle`, beside `LICENSE`, and the command hands it to whatever program opens PDFs on your machine. The same thing is the last button on the palette, under *Data and checks*. With the interface in Serbian you get the Serbian guide instead; if only one of the two is present, you get that one whichever language you are in.

Installing

PowerShell execution policy

Windows refuses to run `.ps1` files by default — *running scripts is disabled on this system*. Two ways around it:

```
# One-off. Nothing on the machine is changed permanently.
powershell -NoProfile -ExecutionPolicy Bypass -File .\install.ps1

# Or once, for your account only. No administrator rights needed.
Set-ExecutionPolicy -Scope CurrentUser RemoteSigned
```

Files extracted from a downloaded ZIP stay marked as internet content and remain blocked even under `RemoteSigned`. Clear the mark once:

```
Get-ChildItem -Recurse | Unblock-File
```

Bundle installer — recommended

Close AutoCAD, then from the folder you extracted FiberQ into:

```
cd <extracted folder>\ribbon
.\install.ps1
```

- The `ribbon` folder must stay a subfolder of the folder holding `fq-sema.lsp`, `fq-kucista.lsp`, `fq-jezik.lsp` and `LICENSE`. The script stops with `Nema fajla: ...` if one is missing. Its own messages are in Serbian.
- **It stops if AutoCAD is running**, because AutoCAD can write its own copy of the CUIX over the new one on exit. `-Force` overrides that.
- It runs `test-cuix.ps1` and `test-lisp.ps1` first and stops if either fails, rather than installing a package it knows is broken.
- `-AllUsers` installs to `%ProgramData%\Autodesk\ApplicationPlugins\` and needs an elevated prompt.

It installs to:

```
%APPDATA%\Autodesk\ApplicationPlugins\FiberQ.bundle\
  PackageContents.xml
  LICENSE
  Contents\  fq-jezik.lsp  fq-sema.lsp  fq-kucista.lsp
            fq-ribbon.lsp  FiberQ.cuix  FiberQ_SR.cuix
```

Start AutoCAD. The ribbon tab **FiberQ** appears.

A `.dat` file left inside `Contents` by an older release is moved out to `%APPDATA%\FiberQ\` — or **deleted**, if a file of that name is already there. The installer says which it did. This applies to `fq-boje.dat` and `fq-kucista.dat`; `fq-kablovi.dat` was never written there.

Manual load

Use this if you cannot run PowerShell, or want to run from a working copy.

1. Put `fq-jezik.lsp`, `fq-sema.lsp`, `fq-kucista.lsp`, `fq-ribbon.lsp`, `FiberQ.cuix` and `FiberQ_SR.cuix` in one folder. The distributed layout splits them between the root and `ribbon\`, so copy them together or add both folders in step 2.
2. *Options* → *Files* → *Support File Search Path* — add that folder.
3. `APpload`, select `fq-ribbon.lsp`, press **Load**.
4. Press **Contents...** and add `fq-ribbon.lsp` to the **Startup Suite**.

`fq-ribbon.lsp` is the only file you load by hand. It pulls in the three libraries at the first FiberQ button or command, so `FQCABLE`, `FQFAT` and the rest are undefined until then.

To run from a working copy without touching the support path, in `acaddoc.lsp`:

```
(setq *FQ-DIR* "c:\\path\\to\\working\\copy\\ribbon")
(load "c:\\path\\to\\working\\copy\\ribbon\\fq-ribbon.lsp")
```

`*FQ-DIR*` is a single folder, so the folder with the three libraries still has to be on the support file search path. Set it **before** loading. `(setq *FQ-RIBBON-AUTO* nil)` stops the ribbon from loading itself.

Where FiberQ looks for its files

1. the folder in `*FQ-DIR*`, if set
 2. `%APPDATA%\Autodesk\ApplicationPlugins\FiberQ.bundle\Contents\`
 3. `%ProgramData%\...\FiberQ.bundle\Contents\`
 4. the bare file name through the support file search path — **last**, so an old loose copy next to someone's drawing cannot win
-

If the FiberQ tab does not appear

1. `FQSTANJE` . *Unknown command* means no LISP is loaded: `APpload` → `...\FiberQ.bundle\Contents\fq-ribbon.lsp` → **Load**, then **Contents...** → **Startup Suite**. Use `FQSTANJE`, not the alias `FQSETTINGS` — only `FQSTANJE` answers with just `fq-ribbon.lsp` loaded.
2. `FQRIBBON` . Loads the ribbon by hand and prints the CUIX path it used. If it cannot find the file, add `...\FiberQ.bundle\Contents` to the support file search path. If it says the ribbon is already loaded, go on.
3. `CUILOAD` . The loaded groups must include **FIBERQ**. If they do and the tab is still missing, it is not in your workspace: `CUI` → *Customize Workspace* → tick **FiberQ** under *Ribbon Tabs* → *Done* → *OK*.
4. **Installer run with** `-Force` **while AutoCAD was open?** AutoCAD may have written its old CUIX back on exit. Close it and run `install.ps1` again.
5. **Blank or wrong icons?** Stale cache. Close AutoCAD, delete every `.mnr` from `Contents`, restart.

Once the CUIX has loaded it stays registered in the AutoCAD profile. You do this only the first time.

The ribbon, the menu and the palette

Panel	Contents
Cables	Split button Cable : Duct (FQCDUCT), Aerial (FQCAERIAL), Micro duct (FQCMICRO), Indoor (FQCINDOOR), manual entry (FQCABLE). Plus Cable with bends (FQBEND), Align to cable (FQALIGN), Change category (FQCABLECAT), Colour code (FQCC), Type SM/MM (FQFIBERTYPE)
Closures and joining	Closure split button: N (FQCLOSIN), NL latent (FQCLOSINL), dialog (FQCLOSURE), command line (-FQCLOSURE). Join two (FQJOIN), Join group (FQJOINING), Splice (FQSPLICE), Vertical drop (FQDROP), Cut (FQCUT), Through (FQTHROUGH), Slack 15 m (FQSLACK15), Slack, enter length (FQSLACK)
Splitters	Six ratios FQSPL2 ... FQSPL64 , any other ratio (FQSPLITTER), Splitter into housing (FQSPLIN)
ODF and panels	ODF module 576...12, patch panel 12...144, MDU 12...144. Each list ends with other , which opens the shared housing dialog on that type: FQODFDLG , FQPP , FQMDU
Cabinets and boxes	FDH 12...96 (FQFDH*), other (FQFDHDLG). FAT 1...24 (FQFAT*), other (FQFAT). Mounting: Indoor (FQMNTU), Outdoor (FQMNTS), Free-standing (FQMNTM). Add a fiber (FQADD), Branch out (FQBRANCH), All types... (FQKUC)
OLT and ONT	FQOLT8 , FQOLT16 , ONT subscriber (FQONTK), ONT full entry (FQONT)
Fiber purpose	Ten purposes, FQPURWORK and friends. Fiber description (FQDESC)
Data and checks	FQINFO , Fiber route (FQROUTE), FQBOM , BOM to drawing (FQTABLE), Legend (FQLEGEND), Budget (FQBUDGET), Budget to drawing (FQBUDGETTAB), Check (FQCHECK), Layers (FQLAYERS)
Settings	Designed (FQSTDES), As-built (FQSTASB), Settings (FQSETTINGS), Horizontal (FQORIH), Vertical (FQORIV), Preview (FQDRAG), Snap (FQSNAP), Widen housing (FQWIDEN), Palette (FQ)

Ten buttons are split buttons: the arrow opens the list, the top half repeats your last pick. The label and icon never change.

The capacity buttons open no dialog. They ask you to click a fiber of the cable and — for a housing without a splitter — how many fibers go onto connectors; Enter takes all, 0 takes none and leaves the bundle for **Add a fiber**. With no FiberQ fiber in the drawing, or if you cancel the pick, the housing is placed standalone. Model, connector (SC/APC until changed) and mounting come from your last housing-dialog choice. The **other** entries at the end of each list open the full dialog instead.

A capacity button for an FDH, MDU or FAT of 12 ports and up always puts a splitter in: 12 → 1:8, 24 → 1:16, 48 → 1:32, 72 and 96 → 1:64. For an empty cabinet or another ratio use the dialog, or FQSPLIN afterwards.

No button in the Settings panel shows its state; each prints the current setting line after you press it.

The pull-down menu

The same items, twelve submenus plus **Palette**. It appears only with MENUBAR set to 1 . Two commands have no ribbon button: **Save catalogue** (FQCATSAVE , menu only) and **Save cable catalogue** (FQCABLESAVE , menu and palette). **Cable length** (FQLENGTH) is on neither — type it.

The palette

`FIBERQ` or `FQ` opens *FiberQ CAD — schematic elements*. It is a modal dialog, not a docked tool palette: it closes on a button press, the command runs, and it returns if *Keep the palette open* is ticked. One press is one undo step.

Not on the palette: cable with bends, change category, the closure dialog, slack with entered length, patch panels, MDU cabinets, FAT boxes, the shared housing dialog, the mounting buttons, Add a fiber, Branch out, Fiber description, ODF 576 and 288, FDH 12 and 72, full ONT entry, Save catalogue.

ODF and FDH behave differently on the palette:

	Ribbon	Palette
Values from	Last housing-dialog choice	Command-line questions
Model, connector, rack strip	Not asked	Asked
Splitter inside	Where the catalogue allows	Never
Label	Next free number	Asked

FDH 48 on the ribbon gets a 1:32 splitter; on the palette it is empty.

Settings and rules

Setting	Set with	Default
Status of new elements	<code>FQSTDES</code> / <code>FQSTASB</code> , palette	Designed
Orientation	<code>FQORIH</code> / <code>FQORIV</code> , palette	Horizontal
Mounting of FDH cabinets	<code>FQMNTU</code> , <code>FQMNTS</code> , <code>FQMNTM</code>	Indoor
Preview with the mouse	<code>FQDRAG</code> , palette	On
Snap to connector	<code>FQSNAP</code> , palette	On
Cable widens the housing	<code>FQWIDEN</code> , palette	On
Colour code standard	<code>FQCC</code>	TIA-598
Fiber type	<code>FQFIBERTYPE</code> , cable dialog	SM

`FQSETTINGS` (`FQSTANJE`) prints status, orientation, the active colour code, and all three switches — preview, snap and **whether a new cable widens the housing at its end**. Mounting, fiber type and remembered model are not in it.

All of these live in the session and reset when AutoCAD closes. The language is stored in your profile; so are standards saved by `FQCC` and catalogues saved by `FQCATSAVE` / `FQCABLESAVE` — only the *active choice* resets.

Four rules worth knowing before you start:

- **FiberQ draws only in the world coordinate system.** In a rotated or shifted UCS every drawing command stops and tells you to run `UCS` then `W` . Reading commands work either way.
- **Object snap does not work during the mouse preview.** Press **O** at the point prompt for one normal pick with osnaps. `FQSNAP` controls FiberQ's own snapping, which works **only while the**

preview runs and catches the first connector of an ODF, cabinet or box frame. `FQDRAG` turns the preview off for good, and with it both.

- **Mirroring is not a setting.** A cable *ending* at a housing draws that housing mirrored, body to the right of the connector column, and says so. Fibers arriving from the left or passing through keep the normal side. A mirrored housing has its ports facing away, so no new cable can start from it — snap the *end* of a cable onto its connectors.
- **Alignment is one-shot.** After a housing or closure is placed, FiberQ offers to align the next cable with it, and the next cable uses that offer up.

Language

FiberQ runs in English and Serbian. Switch with `FQLANG` or `FQJEZIK`. The prompt is bilingual — keywords `English` and `Srpski` — so you can always read your way back. There is no button for it; type it. The choice is stored in your AutoCAD profile and survives restarts.

Changes	When
Prompts, messages, bracketed keywords	Immediately
Palette and dialogs	Next time you open them
Layer descriptions	New layers at once; run <code>FQLAYERS</code> for existing ones
Ribbon and pull-down menu	Immediately, if the ribbon is loaded

English and Serbian are two ribbon packages, both installed; `FQLANG` swaps them.

What does not change

- **Command names.** `FQKABL` stays `FQKABL`, so scripts and macros keep working. English names are **aliases** — `FQCABLE`, `FQFAT`, `FQCHECK` and about seventy more. Both work in both languages.
- **Layer names.** `FQ_*`; only the Description follows the language.
- **Keys in the drawing.** Element type, status, and the codes for purpose, category, mounting and fiber type are fixed. A drawing made in one language reads in the other, and its bill of materials comes out in the language you are using now.
- **Typed keywords.** Translated once, then frozen. They are not always the label: *Spare* is typed `Unused`, *Through* is typed `Transit`, because a keyword may not contain a space or start another.
- **Your catalogues and text.** `fq-kucista.dat` and `fq-kablovi.dat` are language-neutral. Labels, models, addresses and closure locations pass through unchanged.

`FQMNTU`, `FQMNTS` and `FQMNTM` have no English alias on purpose: they pass a language-neutral catalogue name.

Pick the language before you start a drawing. Element labels are text, not keys, so they follow the language: a cabinet placed in Serbian is `ODO 1`, the next one in English is `FDH 1`. Numbering counts per prefix, so both carry number 1. Nothing breaks, but labels end up mixed.

Cables

A cable is usually drawn first, but not always: housings and splitters can be placed on their own and a cable snapped on afterwards.

Each fiber is one polyline carrying its own record — cable label, tube, position, ordinal, colour, status, purpose, fiber type. The polyline is drawn in a neutral colour; the fiber colour appears in its label and tube ellipse. Line colour is reserved for **purpose**.

Command	Serbian	What it does
FQCABLE	FQKABL	Cable by hand, all questions on the command line
FQCABLECAT	FQKABLK	Asks the category, then opens the catalogue dialog
FQCAERIAL FQCDUCT FQCMICRO FQCINDOOR	FQKVAZ FQKPOD FQKMIK FQKINS	Catalogue dialog on one category
FQBEND	FQLOM	Cable that turns at right angles
FQALIGN	FQPORAVNAJ	Line the next cable up with an existing one
FQLENGTH	FQDUZINA	Write the real route length in metres
FQFIBERTYPE	FQVRSTA	Set SM or MM on selected fibers
FQCC	—	Colour code standard editor
FQCABLESAVE	FQKABLSAVE	Save the cable catalogue to disk

From the catalogue

- **Cable type** — edit box plus list. A name not in the list is added to that category, inheriting the capacity list of the selected type.
- **Capacity** — tubes × fibers per tube.
- **Orientation** — Horizontal, Vertical up, Vertical down.
- **Colour code** — the selected standard becomes active for everything drawn afterwards.
- **Label** — empty means automatic: K1 , K2 , ...
- **Length** — in drawing units; a decimal comma is accepted. Below 12 is raised to 12.
- **Drag the length** — works only while the mouse preview is on.

By hand

FQCABLE asks: number of tubes <2>, fibers per tube <6>, line length <40.00>, cable type, label <auto>, status <D>, fiber type <SM>, insertion point.

A cable drawn this way **carries no category**, so use the catalogue commands when the category has to reach the bill of materials. A type typed here is not added to the catalogue.

What gets drawn

- One polyline per fiber, with an extra gap at every tube boundary.
- An ellipse at each end of every tube, in that tube's colour, numbered.

- A label on every fiber at both ends: ordinal, colour name, group mark past the end of the colour table — `13.Blue-p` , `25.Blue-2p` . The middle is left clear for a closure.
- Cable label and type above the first fiber: `K1 TOSM 03 (2x12)` .

Each fiber is identified as `label/tube/fiber` — `K1/2/6` .

On a short cable only the left label column is drawn, and FiberQ says so. Above 200 fibers you confirm before drawing; above 240 the preview thins to the first and last fiber of each tube, though the cable is drawn in full.

Placing it

A silhouette follows the cursor, first fiber in red. Click places it, **O** gives one classic pick with object snap, any other key cancels. Near an ODF, patch panel, FDH, MDU or FAT frame the cable jumps onto the nearest numbered connector and takes over the frame's angle; the command line names the connector.

Orientation

Angle	Meaning
0	Horizontal, running right
90	Vertical, running up
270	Vertical, running down

There is no 180. To run a cable left, draw it right and put the housing at the far end, or use `FQBEND` . An element attached to an existing cable takes that cable's angle.

Colour code standards

Standard	Sequence starts
TIA-598	Blue, Orange, Green, Brown, Slate, White, ...
IEC 60304	Red, Green, Blue, Yellow, White, Grey, ...

Both ship with FiberQ. TIA-598 is the base standard: it cannot be renamed or deleted, and is put back if a saved file lacks it. Fibers beyond the table repeat the colours with a group mark; **the shipped standards** use `-p` , `-2p` , `-3p` and so on — the marks are part of each standard and you can change them.

`FQCC` opens the editor: create or copy a standard, rename or delete it (delete asks twice; the last standard cannot be deleted), build the sequence from the twelve known colours or add your own with **Other colour...**, reorder a slot, replace one, **remove a colour**, paste the whole sequence as a comma-separated list, set the group marks, and make a standard active.

Edits apply at once; **Save** writes `fq-boje.dat` . Once that file exists it replaces the shipped list outright — standards are **not merged**, so a standard added in a later version appears only after you delete or edit your file. Only TIA-598 comes back automatically. Changing the active standard does not repaint fibers already drawn.

The cable catalogue

Four categories — Aerial, Duct, Micro duct, Indoor — each with its cable types and capacity lists.

`FQCABLESAVE` writes `fq-kablovi.dat` ; it is plain text and can be edited by hand.

Saving does not freeze it. While the shipped catalogue is unchanged nothing is added, so a type you deleted stays deleted. When a later version ships a new catalogue, every shipped type your file lacks is appended and reported at start-up. Type names are yours and are never translated.

Cables with bends

FQBEND asks the same questions as **FQCABLE** except the length, then the path is clicked. Click bends 90° and starts the next segment; Enter finishes; **U** undoes the last bend; **O** gives one bend point with object snap.

The axis alternates after each bend; orientation sets only the first segment. Fibers keep their order through the turns and **carry the angle of the last segment**, so that end behaves like the end of a straight cable.

If a segment is so short that the bundle would fold back, FiberQ refuses the path and draws nothing, naming the segment and the fiber. If only the first or last segment is too short for labels, the cable is drawn without labels at that end.

Lining up with an existing cable

FQALIGN sets up the *next* cable. Run it, then pick either a fiber or the frame of an ODF, cabinet, box or closure.

Picking a fiber asks how to align:

Option	Effect
Grid	Takes over the whole row list of the reference cable, fiber for fiber
Start	Takes the first row only; the natural pitch continues

Use Grid when the two cables have different fibers per tube — the extra tube gaps accumulate and the rows drift apart.

Picking a frame asks nothing: it takes the frame's angle and row, and if the frame has a connector column the next cable starts at the first connector with no point picked.

The alignment is used once and then cleared. Running **FQALIGN** again cancels a pending one. A mirrored housing cannot be aligned from.

Real length

The drawing is a schematic: 116.9 units is not 116.9 metres, so the real length is **written**, not measured. **FQLENGTH** asks for one fiber, then the real length of that cable in metres. Enter keeps what is stored; negative values are refused. If the fiber runs through several joined cables, FiberQ asks once per cable label.

Two consequences: it writes across the **whole drawing**, not a selection, because fibers of one cable also sit inside housings and closures; and the value is stored **per cable label**, so all twelve fibers of K1 carry the same length, as do both legs of a cut fiber. **FQINFO** shows it as **K1:120.50 + K2:80.00** . No report totals it — the bill of materials counts fibers, not metres.

Housings: ODF, patch panel, FDH, MDU, FAT

Type	Label prefix	Capacities	Splitter inside
ODF module	OR	12, 24, 48, 72, 96, 144, 288, 576	no
Patch panel	PP	12, 24, 48, 72, 96, 144	no
FDH cabinet (Serbian ODO)	FDH	12, 24, 48, 72, 96	from 12 up
MDU cabinet	MDU	12, 24, 48, 72, 96, 144	from 12 up
FAT box (Serbian ZOK)	FAT	1, 2, 4, 6, 8, 12, 16, 24	from 12 up

Drawing keys stay ODO , ZOK and IPAN ; only what you see says FDH, FAT, MDU.

Three ways to place one

How	Commands
Capacity button — draws at once from your last dialog choice	FQODF12 ... FQODF576 , FQFDH12 ... FQFDH96 , FQPP12 ... FQPP144 , FQMDU12 ... FQMDU144 , FQFAT1 ... FQFAT24
FQKUC — one dialog for all five	FQODFDLG , FQPP , FQFDHDLG , FQMDU , FQFAT open it on one type
One question per line, for scripts	-FQODF , -FQPP , -FQFDH , -FQMDU , -FQFAT

What you set

Field	Values
Capacity	from the catalogue of that type
Model	maker and type, for the bill of materials. Starts empty . Remembered per type for the session; FQCATSAVE keeps it
Connector	SC/APC , SC/UPC , LC/APC , LC/UPC , LSH/APC , FC/PC , ST/PC ; the -FQ forms take any text
Mounting	FDH indoor / outdoor / free-standing, MDU indoor / free-standing, FAT wall / pole / indoor; none for ODF and patch panel
Address	printed above the frame — cabinets and FAT boxes only
Used	connectors taken; empty means as many as the cable brings. Ignored when a splitter is in
Rack and module	ODF and patch panel only, 1 to 24 modules, drawn one under another. The module goes into the fiber address: OR 1 L1, M2/45

FQMNTU , FQMNTS and FQMNTM set mounting for **FDH cabinets only**.

On a cable, or standing alone

Pick **A cable** and click a fiber near the cable end — the housing takes that cable's angle and row spacing. Pick **Nothing** and place it yourself; it then also sets alignment for the next cable.

When it goes onto a cable and **Used** was left empty, FiberQ asks how many fibers go onto connectors. Enter takes as many as the cable brings, **up to the capacity**; 0 leaves the bundle to be connected later with FQADD .

The frame always reaches down to the lowest fiber of the bundle, even where there are more fibers than connectors. At the far end of a cable the housing is drawn mirrored, so the frame does not lie over the

bundle.

Which fiber gets which port

- **ODF, patch panel** — forward from 1; every connector drawn.
- **FAT** — backward from the capacity; every connector drawn, including empty ones, so a FAT 12 shows 1...12.
- **FDH, MDU** — backward from the capacity; only used ports drawn. FDH 48 with four fibers uses 48, 47, 46, 45.
- **With a splitter** — outputs 1...n from the top. In a FAT the feeding fiber sits on the last connector. In a cabinet the feed keeps the highest port unless you change it in `FQSPLIN`.

If the ratio does not fit the free ports, FiberQ says so and draws as many outputs as fit.

`FQADD` — one more cable, or more fibers

Click the frame, then a fiber of the cable entering it, then say how many more to connect; `0` only widens the frame. Works on all five types, not on a closure, and the cable must run in the frame's orientation.

A housing that already holds a splitter is **stretched, not redrawn**. In an FDH or MDU cabinet free ports below the bundle can still be added; fibers left without a port stay spare.

If the old drawing sits on a locked or frozen layer nothing is redrawn, and the command says so.

`FQBRANCH` — a cable leaving the housing

Click the frame; answer tubes, fibers, length, label, status. **Elbow** connects the fibers at the connector column and widens the frame; **Straight** connects nothing. A second branch goes left of the first.

A bundle wider than the frame runs out to its left, and the command warns you. Refused on a housing with a splitter and on a mirrored housing — use `FQBEND` or `FQDROP` there.

`FQDESC` — text next to a fiber

Click a fiber and type the text. It is drawn above the fiber and stored with it; empty text deletes it.

The catalogue

`FQCATSAVE` writes the catalogue, with your models and the connector list, to `fq-kucista.dat`. The file is read line by line, never executed, so you can edit it by hand. New shipped types are appended at start-up and reported; your own entries are never changed, and a type you deleted stays deleted.

Splitters

`FQSPLITTER` draws a splitter anywhere. It asks for the ratio K (Enter gives 1:32), a label (Enter gives the next free `SP1`, `SP2`, ...) and the status. The body carries the ratio, the loss and the label.

ratio	1:2	1:4	1:8	1:16	1:32	1:64
loss	3.6 dB	7.3 dB	10.5 dB	13.7 dB	17.0 dB	20.5 dB

Any other K is computed as $0.8 + 3.01 \times \log_2(K)$; K of 1 or less counts as 0.0 dB.

The ribbon panel and palette group carry FQSPL2 ... FQSPL64 — no questions, next free label, current status, you only pick the point.

Two grips, the I key

The splitter is placed by the point you are connecting it with, and there are two. Press I during the preview to switch; the command line names the active one.

grip	use it when
end of the first output (default)	the splitter feeds housing ports
end of the input	the splitter sits on a cable fiber

The grip always starts at the first output and is not remembered between commands. The first output is drawn red in the preview.

Fiber ends are not snap targets. FiberQ snaps only to the connector ports of housing frames. To hang a splitter on a fiber end press O during the preview — that one pick falls back to a normal pick with AutoCAD object snap. With FQDRAG off there is no preview, and I is unavailable.

Splitter into a drawn housing

FQSPLIN adds a splitter inside a housing already in the drawing. Click the frame; the ratio is always the last question, defaulting to the largest standard ratio that still fits. Any ratio from 1:2 to 1:64 can be chosen by hand, in the dialog or on the command line.

- **FDH / MDU with a cable in it** — click a fiber, answer which port feeds the splitter (Enter takes the highest), then the ratio.
- **FAT with a cable in it** — click the feeding fiber; the feed goes to the last connector. With more than one fiber on the connectors FiberQ refuses and tells you to draw the box with the splitter from the start.
- **Empty housing** — nothing is clicked; the feed goes to the bottom row.

Limits

- **One splitter per housing.** FQSPLIN refuses a second.
 - **A cascade is a picture, not data.** Nothing links a standalone splitter to its fibers, so FQROUTE will not follow it and the bill of materials counts two unrelated splitters.
 - **The budget therefore asks both stages by hand** — see below.
 - **A splitter in a closure goes in as a standalone symbol.** FQSPLIN takes only FDH, MDU and FAT.
 - **A standalone splitter inside a housing frame stops that housing from being redrawn.** If the splitter belongs to that housing, use FQSPLIN .
 - **Output spacing ignores the tube gap.** Outputs step one grid row each; on a cable with more than one tube there is an extra gap at each tube boundary, so outputs crossing it miss their fibers — with a 1:8 on 2 tubes × 6 fibers, outputs 7 and 8 miss. Drag them by hand. Splitters **inside** a housing are never affected.
 - **No 180° orientation,** so outputs can never face left.
-

Closures

FQCLOSURE opens the closure dialog: type, label, where it sits, the number, status. Then click a fiber of the cable, drag the **left** frame edge across the bundle, and drag the **right** edge for the width. Enter at the right edge accepts 8.00 drawing units; the frame cannot go below 1.60.

type	meaning
N	ordinary closure
NL	latent closure; fibers pass through uncut

The label is written above the frame. A bare number is enough — 5 for an NL gives NL 5. Empty gives the next free number of that type.

Where it sits — manhole, pole, buried, in duct, on facade, or other; written under the label. Two check boxes speed up a run: *Continue* reopens the dialog after each closure, *Same cable* keeps the cable so you only drag the two edges.

FQCLOSN and **FQCLOS**NL open the same dialog with the type preset. **-FQCLOSURE** asks everything on the command line, for scripts; there the place is free text — *in the basement* is kept and read back unchanged.

Notes:

- The frame is read back from its own geometry, so **MOVE** and **STRETCH** work as expected.
- A closure widens by itself for a new cable ending at it while **Widen housing** is on. It never shrinks.
- The bundle height is measured at the **left** edge. If the frame reaches past a bend, FiberQ says so and leaves the frame where you dragged it.
- A closure has no connector column, so no splitter can be drawn into it.

Working with fibers

command	Serbian	what it does
FQCUT	FQPRESECI	cuts a fiber inside a closure or housing
FQJOIN	FQSP0J	joins two fibers, draws the splice point
FQJOING	FQSP0JG	joins a whole group at once
FQSPLICE	FQSPLAJS	splice point on a single fiber
FQTHROUGH	FQPROLAZ	makes a fiber pass through, uncut
FQDROP	FQODVOD	takes a bundle vertically out of a cable

Cutting

FQCUT cuts a fiber **inside** an element. The element is found from the frame, so nothing is asked. Both legs stay, turned black and marked as cut.

Click the side you are retiring. That leg keeps the label with **-U** appended and loses its terminations; the other keeps the identity. The prompt also offers **Window** — first corner inside the frame, then drag across the rows; the first corner decides which side is retired.

Joining

FQJOIN asks for two fibers — click each near the end being joined. The joint lands in the **middle of the closure** that end lies in, and FiberQ prints which closure it found. The two fibers become one polyline with a splice dot, carrying over the chain, terminations and cable lengths of both.

Ends not in the same row are stepped over on a free column. Joints that cannot be placed without overlapping are reported at the end; a wider frame usually fixes it.

FQJOINING does the same for a group: click the **first** fiber of cable A and of cable B, give the count ⟨12⟩ and the step ⟨1⟩. Before drawing it counts pairs not in the same row and asks whether to join anyway — default No.

Splice on a single fiber

FQSPLICE puts a splice point on a fiber that is not joined to anything. **Middle** (default) uses the middle of the closure or housing; **Point** lets you pick; **Window** does several at once. A fiber that gets a splice is no longer marked as through.

The splice point appears in the bill of materials, but the automatic splice count in **FQBUDGET** comes from the fiber's chain of joined sections, not from splice dots. Add such a splice to the count by hand.

Through fibers

FQTHROUGH marks a fiber as passing through. If the fiber has a splice point, the point is removed and the fiber stays whole. If it has been cut, click the **other leg** anywhere and the two are joined back into one straight fiber; Enter instead only records *through* without joining.

Vertical drop

FQDROP takes a bundle out of a cable at right angles. Click the **first** fiber of the branch near the end being branched, give how many fibers ⟨6⟩, then pick where the branch arrives. Bends are staggered so the vertical runs do not cross, and the fiber order is kept.

Fiber purpose and type

FQPURPOSE (**FQBOJA**) sets what a fiber is used for: answer the purpose, then select the fibers. Colour and line weight follow from it, and the value is what the bill of materials and the legend group by.

purpose	typed as	colour	drawn
Free	Free	black	thin
Working	Working	red	heavy
Planned	Planned	cyan	heavy
Under construction	Construction	yellow	heavy
Accepted	Accepted	blue	heavy
Reserved	Reserved	magenta	heavy
Spare	Unused	dark ochre	thin
Through	Transit	muted teal	thin
Splitter	Splitter	steel blue	thin
Damaged	Damaged	green	heavy

All ten typed words start with a different letter, so one keystroke picks any of them. Two differ from the name shown — `Unused` and `Transit` — so that no word begins another. The ribbon panel offers every value directly and skips the question.

Cutting returns both legs to Free. Joining and splicing leave the purpose alone.

`FQFIBERTYPE` (`FQVRSTA`) records singlemode or multimode, and the choice also applies to the next cable you draw. Colour and weight are not touched. It matters for the budget: 0.24 dB/km at 1490 nm for singlemode against 3.00 dB/km at 850 nm for multimode, so a multimode span costed as singlemode comes out more than ten times better than it is.

Reports and checks

Command	Serbian	What it does
<code>FQBOM</code>	<code>FQBOM</code>	Bill of materials
<code>FQTABLE</code>	<code>FQTABELA</code>	The same list, in the drawing
<code>FQBUDGET</code>	<code>FQBUDZET</code>	Optical loss budget
<code>FQBUDGETTAB</code>	<code>FQBUDZETT</code>	The same budget, in the drawing
<code>FQLEGEND</code>	<code>FQLEGENDA</code>	Symbol legend
<code>FQINFO</code>	<code>FQINFO</code>	All data on one element
<code>FQROUTE</code>	<code>FQTRASA</code>	One fiber's route, highlighted
<code>FQCHECK</code>	<code>FQPROVERA</code>	Drawing check
<code>FQLAYERS</code>	<code>FQSTIL</code>	Layer standard

Reading commands work in any UCS. `FQTABLE` , `FQBUDGETTAB` and `FQLEGEND` place text and refuse outside the world UCS. **What they write is always horizontal**, whatever the current drawing orientation.

Bill of materials

FQBOM scans the whole drawing, not a selection, and merges identical items into one counted row. What splits a row depends on the element:

element	rows split on
ODF, patch panel	model, capacity, ports used
FDH, MDU	model, capacity, ports used, mounting
FAT	model, capacity, connector type, mounting, splitter
closure	type (N / NL), location
OLT	number of PON ports
fiber	cable, label, purpose, cut or through

Designed and as-built never merge. Slack is stored in whole metres, so 12.5 reads 13 everywhere.

FQTABLE writes the same rows as text on layer FQ_TABELA, from a point you pick. It is a snapshot: delete and rerun after changes.

Optical budget

FQBUDGET asks for the source (Fiber / Manual), fiber type, length in km (2.0), splices (8), connector pairs (4), two splitter stages (4) and (8) (0 for a stage that is not there) and the wavelength — **1490 nm by default** for singlemode, **850 nm** for multimode. Fiber reads splices, pairs and type from the fiber you pick.

Term	The number
Fiber	km × dB/km: SM 0.35 at 1310 nm, 0.24 at 1490, 0.22 at 1550; MM 3.00 at 850, 1.00 at 1300
Splices	× 0.10 dB
Connectors	× 0.30 dB per pair
Splitters	1:2 = 3.6 ... 1:64 = 20.5 dB, any other computed
Margin	3.00 dB, always

Class B+ (28 dB) and C+ (32 dB) verdicts both print. FQBUDGETTAB asks everything again, then for an insertion point; its proportional text does not align in columns.

Route, element data, check

FQROUTE prints one fiber's whole route — segments, splices, ends, connector pairs — and highlights it until the next FQROUTE.

FQINFO prints one element twice: the readable fields, then the stored record in brackets. The bracketed keys are never translated — that raw line is what to quote in a support request.

FQCHECK reports on the whole drawing and **never repairs**: duplicate fiber id, duplicate ordinals in a cable, a port used twice, a closure with no splice, a splice with no fiber, working fibers with no termination. Numbering gaps and both legs of a cut fiber are not errors. Latent closures (NL) are skipped by the *closure with no splice* check only.

Legend and layers

`FQLEGEND` asks for a top-left insertion point and draws the symbols, the ten fiber purposes in colour and lineweight, and the active colour code on layer `FQ_LEGENDA`. It is a snapshot: delete and rerun after a change of standard or language.

`FQLAYERS` creates the missing `FQ_` layers and sets colour, lineweight and Description on the rest, fifteen in all. Run it on older drawings and after `FQLANG`. It leaves layer states and non-standard layers untouched. Both commands turn on lineweight display.

Layer	Colour	Lineweight	Description
<code>FQ_VLAKNO</code>	7 white/black	0.13 mm	Fiber
<code>FQ_CEVCIKA</code>	7 white/black	0.13 mm	Cable tube
<code>FQ_OZNAKA</code>	7 white/black	0.13 mm	Fiber labels
<code>FQ_NASTAVAK</code>	1 red	0.25 mm	Closure
<code>FQ_KASETA</code>	7 white/black	0.13 mm	Closure cassettes
<code>FQ_VARENJE</code>	1 red	0.13 mm	Splice
<code>FQ_SPLITTER</code>	5 blue	0.25 mm	Splitter
<code>FQ_ODF</code>	4 cyan	0.25 mm	ODF module
<code>FQ_ODO</code>	3 green	0.25 mm	FDH cabinet
<code>FQ_OLT</code>	6 magenta	0.25 mm	OLT
<code>FQ_ONT</code>	2 yellow	0.25 mm	ONT and drop
<code>FQ_REZERVA</code>	30 orange	0.13 mm	Cable slack
<code>FQ_TABELA</code>	8 grey	0.13 mm	Bill of materials and budget
<code>FQ_LEGENDA</code>	8 grey	0.13 mm	Symbol legend
<code>FQ_ZOK</code>	4 cyan	0.25 mm	FAT box

Layer names never change and are the same in both languages; only the Description follows the interface language. Fibers carry their own colour and lineweight, set from the purpose.

A locked, frozen or off layer takes the geometry but not the data — the element is drawn without its record, and reports do not see it. FiberQ warns once per layer: undo, unlock, repeat.

Your data

File	Holds	Written by	Location
<code>fq-boje.dat</code>	Colour code standards	<code>FQCC</code>	<code><profile>\</code>
<code>fq-kucista.dat</code>	Housing catalogue, connectors	<code>FQCATSAVE</code>	<code><profile>\Support\</code>
<code>fq-kablovi.dat</code>	Cable catalogue	<code>FQCABLESAVE</code>	<code><profile>\Support\</code>

`<profile>` is AutoCAD's `ROAMABLEROOTPREFIX`, usually

`%APPDATA%\Autodesk\AutoCAD <version>\<Rxx.x>\<language>\`. If that folder cannot be written, all three

fall back to `%APPDATA%\FiberQ\`. **Nothing is ever written into** `FiberQ.bundle`. `FQVERSION` prints the exact paths.

Saves are atomic: `.tmp` → `.bak` → real file. A leftover `.bak` means an interrupted save — rename it back by hand. `.dat` files are read line by line and never executed.

Upgrades. New shipped housing and cable types are appended to your file; your own entries are untouched. Run `FQCATSAVE` or `FQCABLESAVE` to keep them, otherwise the merge repeats at every start and a type you deleted keeps coming back. Two limits: the connector list in `fq-kucista.dat` is replaced, not merged; and `fq-boje.dat` carries no version and is never merged, so a saved colour file hides later built-in standards — only TIA-598 is restored.

`fq-boje.dat` is also looked for through AutoCAD's file search, but **only after** both profile locations have been tried, so a stray copy beside the open drawing is used only when you have no colour file of your own. The two catalogues have no such fallback.

What is stored in the drawing

Every element carries a record under the application name `FIBERQ`, on layers named `FQ_*`. That is why the reports work on a drawing you did not make, without your catalogues or settings.

- Drawings from older releases still read; later fields come back empty. There is no migration step.
- `COPY` copies the record too, so `FQCHECK` will report the duplicate fiber ids.
- Connectors and ports are plain geometry with no record — you cannot click one to read data or find one with a filter.

Troubleshooting

Symptom	What to check
No FiberQ line at start-up, <code>FQVERSION</code> unknown	Package not loaded. Confirm the bundle path, then <code>APpload</code> → <code>fq-ribbon.lsp</code> → Startup Suite.
Commands work, ribbon tab missing	See If the FiberQ tab does not appear .
<code>FQFAT</code> , <code>FQPP</code> , <code>FQMDU</code> , <code>FQKUC</code> , <code>FQBRANCH</code> unknown, but cables work	<code>fq-kucista.lsp</code> did not load. <code>FQVERSION</code> shows which parts loaded.
<i>FiberQ draws only in the WCS</i>	Type <code>UCS</code> , then <code>W</code> . Report-only commands work in any UCS.
Element drawn but the reports ignore it	Its layer was locked, frozen or off, so the record was not written. Undo, unlock, repeat.
Object snap dead while dragging	The preview bypasses snap. Press <code>O</code> for one normal pick, or switch the preview off with <code>FQDRAG</code> .
Counts or labels wrong after copying	Run <code>FQCHECK</code> ; duplicate ids come from <code>COPY</code> .
Model list reads (<i>none entered</i>)	No model ships with the catalogue. Type one into the Model field; <code>FQCATSAVE</code> makes it permanent.
Labels mix <code>ODO 1</code> and <code>FDH 1</code>	The language was changed mid-drawing. Keep one language per drawing.
Palette will not open	The <code>TEMPPREFIX</code> folder is not writable.

Uninstalling

Close AutoCAD first — it rewrites loaded customization files on exit.

```
.\install.ps1 -Uninstall
```

Or delete `%APPDATA%\Autodesk\ApplicationPlugins\FiberQ.bundle` by hand (`%ProgramData%\...` for an all-users install). The installer writes no registry entries of its own.

What stays behind: the ribbon tab registration — run `FQRIBBONOFF` **first**, or afterwards `CUILOAD` → unload the `FIBERQ` group; your three `.dat` files, so a reinstall finds them; and the `FQ-JEZIK` profile key.

Drawings are unaffected. The records live in the DWG, and without the add-in the geometry is ordinary polylines and text.

Licence and support

FiberQ 1.0.0, MIT licence. Copyright © 2026 Vladimir Vuković. The full text ships as `LICENSE` in the root of `FiberQ.bundle`, beside `PackageContents.xml`.

FiberQ makes no network connections. It touches only the three data files, the open drawing, the `FQ-JEZIK` profile key, and disposable DCL files in the `TEMPPREFIX` folder.

Support: vukovicvl@fiberq.net · <https://fiberq.net>

When reporting a problem, include the output of `FQVERSION` :

```
FiberQ CAD 1.0.0
fq-sema.lsp      loaded
fq-kucista.lsp   loaded
ribbon          loaded
CAD             24.0s (LMS Tech)   AutoCAD
color code      TIA-598
catalogue       built in, without models
cable catalogue built in
color code file none, only the built-in standards
```

A few more lines follow, giving the full path each catalogue was read from and is written to. Include those too — on their own they answer most *my catalogue disappeared* reports.

The `CAD` line repeats what AutoCAD's own `PRODUCT` variable reports, which is why the AutoCAD-based verticals say `AutoCAD` here even when you are in Map 3D or Civil 3D. What marks a vertical is `(LMS Tech)` in the version string beside it.

For a drawing-specific problem, add the output of `FQCHECK`, and the raw record line from `FQINFO` for the element concerned.