BOVNAR · V1.1

Bovnar — Unit Ambiguities

Bovnar (BVNR) v1.1 documentation · Unit Ambiguities · 2026-07-28

> **Spec version:** 1.1 **Status:** Reference — companion to the unit registry and the cheat sheet **Scope:** Every token that could plausibly mean two things — what Bovnar reads it as, and how to write the other meaning.

Companion to Unit & Currency Reference (the registry) and Unit & Currency Cheat Sheet (the symbol tables). Every row here was checked against the reference parser; where a token is refused, it really is `error_unit_illegal`.

Table of Contents

1. How a token is resolved

A unit component is resolved in this order. Knowing the order explains every table below.

Step	Rule	Consequence
1	A `$` anywhere selects the **currency** table	`$USD`, `k$EUR`, `k~$EUR`. A bare code is never a currency: `USD` is `error_unit_illegal`
2	Otherwise the base symbol is the **longest alias suffix** of the token	`min` matches the alias `min` (minute) before it could match `in` (inch)
3	Whatever precedes the base is the **prefix**, with `~` optional	`k~g` and `kg` are one unit; the writer always emits `k~g`
4	The **prefix policy** of that unit must allow it	`Ki~m` and `Kim` both fail; IEC prefixes are for `b` / `B` only
5	A handful of spellings are **refused by name**	`usb`, `kt` — see §4

Rule 2 is the load-bearing one: **a token that is itself a unit is always that unit**, never a prefixed reading of a shorter one. That is what made the compact form (`kg` for `k~g`) safe to introduce — it can only ever be accepted where the parser previously raised an error, so no document written before it existed changed meaning.

Rule 3's `~` is never wrong to write. When in doubt, or when a reader of your file might be in doubt, use it: `m~in` cannot be misread, `min` can be misremembered.

---

2. A bare unit outranks a prefixed reading

These 19 tokens are units in their own right. The prefixed reading a reader might expect is reachable only with the `~`.

Token	Bovnar reads it as	Might be mistaken for	Write that as
`min`	minute	milli-inch	`m~in`
`cd`	candela	centi-day	`c~d`
`ft`	foot	femto-tonne	`f~t`
`pt`	pint	pico-tonne	`p~t`
`qt`	quart	quecto-tonne	`q~t`
`ct`	carat	centi-tonne	`c~t`
`yd`	yard	yocto-day	`y~d`
`rd`	rod	ronto-day	`r~d`
`ch`	chain	centi-hour	`c~h`
`at`	technical atmosphere	atto-tonne	`a~t`
`au`	astronomical unit	atto-dalton	`a~u`

Token	Bovnar reads it as	Might be mistaken for	Write that as
`kat`	katal	kilo-(technical atmo-sphere)	`k~at`
`nmi`	nautical mile	nano-mile	`n~mi`
`PS`	metric horsepower (Pferdestärke)	peta-siemens	`P~S`
`ph`	phot	pico-hour	`p~h`
`pH`	acidity (pH scale)	pico-henry	`p~H`
`mph`	mile per hour	milli-phot	`m~ph`
`kph`	kilometre per hour	kilo-phot	`k~ph`
`dB`	decibel	deci-byte	(impossible: an SI prefix below kilo is illegal on `B`)

> `pH`, `mph` and `kph` are the reason three quantities became units: while they were not in the table, the compact form resolved them as the picohenry and the milli-/kilophot. Naming the quantity is the durable fix — rule 2 then keeps the prefixed reading out on its own.

---

3. Two prefixed readings compete

When a token can be split in two places, rule 2 decides: the **longer base symbol** wins.

Token	Bovnar reads it as	The other split	Write that as
`dat`	deci-`at` = deci-(technical atmosphere)	deca-tonne	`da~t`
`dau`	deci-`au` = deci-(astronomical unit)	deca-dalton	`da~u` (canonical `da~Da`)

Both are deterministic, not ambiguous — but neither is guessable. Write `d~at` / `da~t` and `d~au` / `da~u` explicitly; the separated form is unambiguous in both directions.

---

4. Refused outright

Two tokens are `error_unit_illegal` on purpose, listed in `.compact_exceptions` in `src/gendata/units.bvnr`. Accepting them would turn a parse error into a quietly wrong unit.

Token	Would have resolved as	Actually means, in the wild	Write
`kt`	kilo-tonne	kilotonne **or** knot, depending on the field	`k~t` (mass) or `kn` (speed)
`usb`	micro-stilb	the bus	`u~sb` (or `µ~sb`) if you really mean microstilb

`kt` is the harder case: both readings are units Bovnar models, so no table lookup can settle it. Reading a speed as a mass is exactly the failure this format exists to prevent, so the author is asked to say which.

---

5. Prefix symbol vs. base unit symbol

Eight symbols are both a prefix and a unit. Bare, they are always the unit; before another unit they are always the prefix.

Symbol	As a bare unit	As a prefix	Example of each
`m`	metre	milli	`m` = metre, `ms` / `m~s` = millisecond
`d`	day	deci	`d` = day, `ds` / `d~s` = decisecond
`h`	hour	hecto	`h` = hour, `hPa` / `h~Pa` = hectopascal
`T`	tesla	tera	`T` = tesla, `TB` / `T~B` = terabyte
`G`	gauss	giga	`G` = gauss, `GHz` / `G~Hz` = gigahertz
`P`	poise	peta	`P` = poise, `PB` / `P~B` = petabyte
`R`	roentgen	ronna	`R` = roentgen, `Rm` / `R~m` = ronnametre
`u`	dalton (ASCII alias of `Da`)	micro (ASCII alias of `µ`)	`u` = dalton, `us` / `u~s` = microsecond, `uu` = micro-dalton

`da` (deca) is a two-character prefix: `dam` = decametre, `dag` = decagram, `das` = decasecond. It is not related to `Da` (dalton) — case separates them (§6).

---

6. Case decides

Symbols are matched byte for byte. These pairs differ only in case and mean unrelated things; several are one keystroke apart in everyday data.

Start with the **bare** symbols, because the prefixed rows below are built from them and they are the three most common symbols in the whole table:

Lower	Means	Upper	Means
`s`	second	`S`	siemens
`h`	hour	`H`	henry
`g`	gram	`G`	gauss
`kn`	knot	`kN`	**kilonewton** — the same trap as `gn` / `GN` and `fn` / `fN` below
`grad`	gradian (also `gradian`, `gon`)	`Grad`	**gigaradian** — and German writes Grad for the degree, so a German-language source is the likely way to type this by accident. Worse than the others: both are angles, so it converts, by a factor of 10^9

Then the prefixed pairs:

Lower	Means	Upper / mixed	Means
`ms`	millisecond	`mS`	millisiemens
`pa` *	—	`Pa` / `pA`	pascal / picoampere
`kg`	kilogram	`kG`	kilogauss
`mt`	millitonne	`Mt` / `MT`	megatonne / megatesla
`gt` *	—	`Gt` / `GT`	gigatonne / gigatesla
`mb`	milli-bit → **illegal** (sub-kilo prefix on `b`)	`MB` / `Mb`	megabyte / megabit
`kb`	kilobit	`kB`	kilobyte
`gib`	—	`Gib` / `GiB`	gibibit / gibibyte
`gal`	gallon	`Gal`	galileo (`mgal` = milligallon, `mGal` = milligal)
`st`	stone	`St`	stokes (`cst` = centistone, `cSt` = centistokes)
`ha`	hectare	`hA`	hectoampere
`hp`	horsepower	`hP`	hectopoise
`gn`	standard gravity	`GN`	giganewton
`gr`	grain	`GR`	gigaroentgen
`dr`	dram	`dR`	deciroentgen
`fn`	fortnight	`fN`	femtonewton
`da`	(prefix deca)	`Da` / `dA`	dalton / deciampere
`np` *	—	`Np` / `nP`	neper / nanopoise
`ev` *	—	`eV` / `EV`	electronvolt / exavolt

Lower	Means	Upper / mixed	Means
`ph`	phot	`pH`	acidity
`ps`	**picosecond** — a valid unit, silently	`PS` / `pS`	metric horsepower / picosiemens
`cv` *	—	`CV` / `cV`	metric horsepower (French chevaux) / centivolt

* not a valid token in that casing — listed because the valid neighbours are easy to mistype.

Common miscasings that are **errors**, not silent variants — `K` is the kelvin, not the kilo prefix:

Written	Result	Intended	Write
`dH`	decihenry — a valid unit, silently	German hardness degree	`°dH`
`cF`	centifarad — a valid unit, silently	hydroponic conductivity factor	`CF`
`fau`	femto-astronomical-unit — a valid unit, silently	formazin attenuation unit	`FAU`
`Kg`, `KG`	`error_unit_illegal`	kilogram	`kg` or `k~g`
`KB`	`error_unit_illegal`	kilobyte	`kB` or `k~B`
`NM`	`error_unit_illegal`	nautical mile	`nmi`
`Cal`	`error_unit_illegal`	kilocalorie (food calorie)	`kcal` or `k~cal`

7. Look-alike characters

Several symbols exist in more than one Unicode code point. Bovnar accepts the variants on input and always emits one canonical form.

Meaning	Accepted spellings	Canonical output
micro prefix	`µ` U+00B5 · `μ` U+03BC (Greek mu) · `u` (ASCII)	`µ` U+00B5
ohm	`Ω` U+2126 · `Ω` U+03A9 (Greek omega) · `ohm` · `Ohm`	`Ω` U+2126
ångström	`Å` U+212B · `Å` U+00C5 · `Å` U+0041 U+030A (A + combining ring) · `angstrom`, `angstroms`	`Å` U+212B
degree	`°` U+00B0 · `deg` · `degr` · `degree` · `degrees`	`°` U+00B0

A document that has been through Unicode normalisation (NFC/NFD/NFKC/NFKD) still parses: every variant above maps to the same unit, which the normalisation test sweep enforces over the whole table.

> **The canonical output is deliberately not NFC.** U+2126 (ohm) and U+212B (ångström) both have singleton NFC decompositions — to U+03A9 and U+00C5 — so a document this library writes is not Unicode-normalised, and running it through NFC **changes its bytes**. It still parses, and re-serialising returns U+2126/U+212B, so the round trip is a semantic fixed point but not a byte fixed point. That matters if you hash, sign or byte-diff bovnar output (the conformance suite's byte-for-byte comparison included): normalise before hashing, or do not normalise at all — but do not assume the two are the same file.

`u` is both the ASCII micro prefix and the bare symbol for the dalton — rule 2 keeps them apart: `u` alone is the dalton, `us` is a microsecond, `uu` is a micro-dalton.

---

8. Temperature scales vs. electrical units

The degree sign is not decoration: without it the token is a different quantity entirely.

With `°`	Means	Without `°`	Means
`°C`	degree Celsius	`C`	coulomb
`°F`	degree Fahrenheit	`F`	farad
`°N`	degree Newton (historical scale)	`N`	newton
`°Ra`	degree Rankine	`Ra`	(accepted alias of `°Ra`)

ASCII aliases avoid the question: `degC`, `degF`, `degRa`, `degN`, `degDe`, `degRe`, `degRo`. `K` (kelvin) is absolute and needs no degree sign.

The water-hardness degrees carry the same warning, and one of them is genuinely dangerous:

With `°`	Means	Without `°`	Means
`°dH`	German hardness degree	`dH`	**decihenry** — a valid unit, so this misreads silently
`°e`	English (Clark) hardness degree	`e`	`error_unit_illegal`
`°fH`	French hardness degree	`fH`	femtohenry
`°rH`	Russian hardness degree	`rH`	rontohenry

With `°`	Means	Without `°`	Means
`°aH`	American hardness degree	`aH`	attohenry

Their ASCII long forms avoid the question entirely: `german_hardness`, `english_hardness` (or `clark_degree`), `french_hardness`, `russian_hardness`, `american_hardness`.

---

9. Same dimension, different quantity

The SI dimension vector cannot tell these apart — they are all dimensionally equal — yet they are not interchangeable in meaning. Bovnar's compatibility check is dimension-based, so it **accepts** conversions in the first group; the second group is protected by an explicit quantity-kind rule.

Dimensionally equal, and Bovnar will convert between them (factor 1) — the distinction is yours to keep:

Units	Shared dimension	The distinction Bovnar cannot see
`J` and `N·m`	kg·m²·s⁻²	energy vs. torque
`Gy` and `Sv`	m²·s⁻²	absorbed dose vs. equivalent dose
`W`, `VA`, `var`	kg·m²·s⁻³	active, apparent and reactive power
`Hz`, `Bq`, `Bd`	s⁻¹	frequency, radioactivity, symbol rate
`mph`, `kn`, `m/s`	m·s⁻¹	genuinely interchangeable — just different scales

> Offering the conversion is not the same as rewriting the annotation. `BVN_UNIT_REDUCE` leaves every unit in this table alone: a `Sv` serialises as `Sv`, never as `Gy`. The collapse folds a compound into the named unit it spells out (`A·s` → `C`); it never substitutes one named unit for another.

Dimensionless but not interchangeable — Bovnar refuses these conversions:

Unit	Kind	Refused against
`b`	information (bit)	`B` — 8 bits are not silently a byte
`B`	information (byte)	`b`
`rad`, `°`, `grad`, `rev`, `arcmin`, `arcsec`	angle (one shared kind, so `°` → `rad` works)	plain numbers, `Hz`, `%`
`sr`	angle, weight 2 (`sr` = `rad²`)	`rad` at exponent 1

Unit	Kind	Refused against
`lm`, `lx`, `ph`	angle, weight 2 — each carries one steradian (`lm` = `cd·sr`)	`cd`, `cd/m²`, `sb`: luminous flux is not luminous intensity, illuminance is not luminance
`dB`	logarithmic	`Np`, `%`, plain numbers
`Np`	logarithmic	`dB`, `%`, plain numbers
`pH`	logarithmic	`%`, `ppm`, plain numbers
`NTU`	turbidity, white light 90°	every other turbidity scale, `%`, plain numbers
`FNU`	turbidity, near-infrared 90°	ditto
`FTU`	turbidity, geometry unstated	ditto
`FAU`	turbidity, attenuation 0°	ditto
`JTU`	turbidity, visual candle	ditto
`PSU`	practical salinity	`‰`, `g/kg`, plain numbers
`%`, `‰`, `‱`, `pcm`, `ppm`, `ppb`	pure ratios	(freely interconvertible, and with a plain number: 1 % → 0.01)

Water chemistry calls the American hardness scale "ppm" (milligrams of $CaCO_3$ per litre). Bovnar's `ppm` is the **dimensionless** 10^{-6} and the hardness scale is `°aH`, an amount concentration — the two carry different dimensions and do not convert into one another. If your source says "ppm hardness", write `°aH`.

The logarithmic kinds are separate because no factor can relate them: 20 dB is a ratio of 100, not twice 10 dB, and a pH one unit lower is a tenfold concentration. `rpm` is a cycle rate and `rev/min` an angular rate — they differ by 2π and do not convert into one another either.

10. An affine scale inside a compound has no value

`°C`, `°F`, `°De`, `°N`, `°Re` and `°Ro` are offset scales: 0 is not nothing. That is fine for a lone temperature and meaningless in a product.

Written	Parses	Converts	Why
`°C`	yes	yes — `20 °C` → `293.15 K`	the offset has somewhere to go
`°C/h`	yes	**no** — `error_unit_mismatch`	the offset is a number of kelvin; `K·s⁻¹` has nowhere to put it
`°C·m`	yes	**no**	ditto

Written	Parses	Converts	Why
`°C²`, `°C·°F`	yes	**no**	an offset scale is meaningful at exponent 1 only

The annotation stays legal, because it is a reasonable thing to write — people do record a rate of change in °C/h. What it means is a temperature **difference** per hour, and the difference is what Bovnar cannot infer from the unit: `20 °C/h` as a rate is `20 K/h`, while `20 °C` as a reading is `293.15 K`. Rather than pick one, the library refuses to produce a number and leaves the components for a consumer that knows which was meant. Write `K/h` when you mean the rate — the kelvin has no offset and the ambiguity disappears.

---

11. A named unit is not its compound

A named unit is a single component; the compound that means the same thing is several. Both are valid and both convert to the same SI value, but annotation-vs-inline reconciliation is **structural**, so mixing the two spellings in one value is `error_unit_mismatch`.

Named	Equivalent compound	Same SI factor?	`<float:64,named> 1.0 compound;`
`mph`	`mi/h`	yes (0.44704 m·s⁻¹)	`error_unit_mismatch`
`kph`	`k~m/h`	yes (5/18 m·s⁻¹)	`error_unit_mismatch`
`kn`	`nmi/h`	yes	`error_unit_mismatch`
`Hz`	`s^-1`	yes	`error_unit_mismatch`
`Pa`	`k~g/(m·s²)`	yes	`error_unit_mismatch`

Pick one spelling per document and stay with it. `bvn_units_compatible` and `bvn_unit_convert_factor` relate them, so a consumer can still convert freely.

---

12. Unit vs. currency

A currency is recognised **only** with the `$` sigil, so the two namespaces cannot collide. The classic cases:

Token	Bare (no `$`)	With `$`
`cup` / `CUP`	`cup` = US cup; `CUP` = `error_unit_illegal`	`$CUP` = Cuban Peso
`BTU`	British thermal unit (`Btu`, `btu` too)	(not an ISO 4217 code)
`SOL`	`error_unit_illegal`	`$SOL` = Solana
`BAR`	`error_unit_illegal` (use lowercase `bar`)	(not an ISO 4217 code)

Token	Bare (no `$`)	With `$`
`USD`	`error_unit_illegal`	`$USD`

A prefix goes before the sigil, `~` optional: `k~$EUR` and `k$EUR` are both thousands of euro. `kUSD` is an error — the sigil is not optional.

---

13. Looks like a unit, is not one

Common abbreviations that are **not** in the table. All are `error_unit_illegal`, which is the point: a wrong unit is worse than a refused one.

Written	Intended	Write instead
`kWh`	kilowatt-hour	`kW·h` or `k~W·h`
`mAh`	milliampere-hour	`mA·h` or `m~A·h`
`mcg`	microgram	`µg`, `ug` or `µ~g`
`cc`	cubic centimetre	`cm³`, `cm^3` or `c~m³`
`sqft`	square foot	`ft²` or `ft^2`
`ksi`	kilopound per square inch	`kpsi` or `k~psi`
`kts`	knots	`kn`
`hr`, `hrs`	hour	`h`
`Cal`	food calorie	`kcal` or `k~cal`
`NM`	nautical mile	`nmi`
`KB`, `Kg`	kilobyte, kilogram	`kB`, `kg`
`kt`	kilotonne or knot	`k~t` or `kn` (§4)

Two that are accepted but rarely mean what the writer intended:

Written	Bovnar reads	If you meant…
`fl`	femtolitre	fluid ounce → `fl_oz`
`mt`	millitonne (= 1 kg)	metric ton → `t`; megatonne → `Mt` or `M~t`

---

14. Water hardness: six scales, one quantity

All six measure the concentration of dissolved alkaline-earth ions, and Bovnar converts freely between them and `m~mol/L`. What differs is the reference compound each scale

counts, which is why their mass-per-litre definitions are not comparable and the amount concentration is.

Scale	Symbol	Defined as	1 mmol/L equals
German	`°dH`	10 mg CaO / L	5.6077 °dH
English / Clark	`°e`, `°Clark`	1 grain $CaCO_3$ / imperial gallon	7.0217 °e
French	`°fH`	10 mg $CaCO_3$ / L	10.0086 °fH
Russian	`°rH`	1 mg Ca / L	40.078 °rH
American	`°aH`	1 mg $CaCO_3$ / L (called "ppm")	100.086 °aH
US grains	`gpg`	1 grain $CaCO_3$ / US gallon	5.8468 gpg
Equivalents	`m~val/L`, `m~eq/L`	½ mmol/L (divalent ions)	2.000 mval/L
Amount	`m~mol/L`	—	1

The traps, in order of how easily they bite:

Trap	Why	Do this
`dH` instead of `°dH`	`dH` is the decihenry — an accepted unit, so nothing errors	keep the `°`, or write `german_hardness`
"ppm" for hardness	Bovnar's `ppm` is the dimensionless 10^{-6}	write `°aH`
`mval/L` for a monovalent species	`val` here is the water-analysis equivalent: 1 val = ½ mol	write `m~mol/L` for the amount directly
Expecting an exact conversion	the degrees are derived from molar masses, so they carry `.exact = false`	ordinary conversion works; a lossless one reports `error_unit_inexact`
`mmol/l` as a "unit"	it is the compound `m~mol/L`, not a base unit	write `mmol/L`, `m~mol/L` or `mmol/l` — all the same
`gr/gal` for `gpg`	`gr/gal` is a mass concentration, `gpg` an amount concentration; they do not convert	write `gpg` for the hardness scale

Turbidity and salinity are instrument scales, and the trap is that they look interchangeable.

Token	Means	Not	Because
`NTU`	white-light turbidity, 90° (EPA 180.1)	`FNU`, `FTU`, `FAU`, `JTU`	equal on a formazin standard, different on real water — the optics respond differently to particle size and colour
`FNU`	near-infrared turbidity, 90° (ISO 7027)	the other four	same reason, from the other side

Token	Means	Not	Because
`FTU`	formazin turbidity, **geometry unstated**	the other four	it does not say which optics were used; that is its entire content
`FAU`	formazin attenuation, 0° (ISO 7027)	the nephelometric three	measures light removed from the beam, not scattered sideways — a different optical quantity, used above ~40 FNU
`JTU`	Jackson candle turbidimeter (historical)	the formazin four	"1 JTU ≈ 1 NTU" holds near 40 units and nowhere else — nonlinear and sample-dependent
`PSU`	practical salinity (PSS-78, a conductivity ratio)	`‰`, `g/kg`	S_P 35 is ≈ 35.165 g/kg — equating them is wrong by ~0.5 %; write `g/k~g` for absolute salinity
`CF`	hydroponic conductivity factor (EC in mS/cm × 10)	`cF` — that is the **centifarad**	uppercase only

`JTU` and `PSU` take no prefix (the candle method cannot resolve below ~25 JTU; practical salinity is bounded by construction). The other turbidity scales do — `m~NTU` is real in ultrapure-water work.

Bovnar refuses every conversion in the "Not" column rather than implying a factor exists. `CF` is the exception in the other direction: it really is a conductivity, so `1 CF = 0.1 mS/cm = 100 µS/cm` converts exactly.

Conductivity and dissolved solids need no units of their own, and that is the trap: people look for one.

Quantity	Write	Not
Electrical conductivity	`µS/cm`, `mS/cm`, `dS/m`, `S/m`	a base unit — there isn't one, and none is needed
… in the pre-SI spelling	`µmho/cm` (`mho`, `mhos`, `℧` are the siemens)	`mho` was unknown before this; it is an alias now
Total dissolved solids	`mg/L`	`ppm` — that is the dimensionless 10^{-6}
Resistivity	`MΩ·cm`	—

For TDS the `ppm` question is softer than for hardness: at 1 kg/L, 1 mg/L is 1 ppm by mass, so `ppm` states a real (dimensionless) quantity. It simply is not the same dimension as `mg/L`, so the two never convert into one another. A meter's "TDS ppm" reading is EC times an instrument factor — record the EC and the factor, or the TDS as `mg/L`.

---

15. Same name, different definition

Not parser ambiguities — real-world ones. Bovnar keeps these distinct, and the distinction is usually where the money is.

Family	Members and exact values
Ton	`t` = 1000 kg · `tn_sh` (short ton) = 907.18474 kg · `tn_l` (long ton) = 1016.0469088 kg
Ounce	`oz` (mass) = 28.349523125 g · `oz_t` (troy) = 31.1034768 g · `fl_oz` (volume) = 29.5735295625 mL
Gallon	`gal` (US) = 3.785411784 L · `gal_uk` = 4.54609 L (US is 0.8327× the imperial)
Pint / quart	`pt`, `qt` (US) vs. `pt_uk`, `qt_uk` — same 0.8327× ratio
Pressure ≈ 1 atm	`atm` = 101325 Pa · `at` (technical) = 98066.5 Pa · `bar` = 100000 Pa
Millimetre of mercury	`Torr` = 101325/760 Pa ≈ 133.322368 Pa · `mmHg` = 133.322387415 Pa — **not** the same unit (they differ by 1.4×10^{-7} relative)
Horsepower	`hp` (mechanical) = 745.6998715822702 W · `PS` (metric) = 735.49875 W
Calorie	`cal` = 4.184 J (thermochemical); the food "Calorie" is `kcal`
Thou	`thou` = `mil` = 25.4 µm — `mil` is **not** a milliradian, which is `mrad` / `m~rad`

16. Quick index: if you mean X, write Y

If you mean	Write	Not
millisecond	`ms`, `m~s`	`mS` (millisiemens)
millisiemens	`mS`, `m~S`	`ms`
kilogram	`kg`, `k~g`	`Kg`, `KG`
kilobyte / kilobit	`kB` / `kb`	`KB`
kilobyte (binary)	`KiB`, `Ki~B`	`kB` (decimal, 1000 B)
megatonne	`Mt`, `M~t`	`MT` (megatesla), `mt` (millitonne)
kilotonne	`k~t`	`kt` (refused)
knot	`kn`	`kt`, `kts`
milli-inch	`m~in`	`min` (minute)
milliradian	`mrad`, `m~rad`	`mil` (thou)
picohenry	`p~H`	`pH` (acidity)
acidity	`pH`	`p~H` (picohenry)
phot	`ph`	`pH`

If you mean	Write	Not
millilitre	`mL`, `m~L`	`ml` is fine too; `fl` is a femtolitre
microgram	`µg`, `ug`, `µ~g`	`mcg`
kilowatt-hour	`kW·h`, `k~W·h`	`kWh`
cubic centimetre	`cm³`, `cm^3`	`cc`
miles per hour	`mph` (or `mi/h`)	mixing both in one value
kilometres per hour	`kph`, `kmh` (or `k~m/h`)	mixing both in one value
degree Celsius	`°C`, `degC`	`C` (coulomb)
thousand US dollars	`k$USD`, `k~$USD`	`kUSD`
a plain count	omit the unit, or `no_unit`	`%` (that is 10^{-2})
German water hardness	`°dH`, `german_hardness`	`dH` (decihenry)
hardness "in ppm"	`°aH`	`ppm` (dimensionless 10^{-6})
hardness in equivalents	`m~val/L`, `mval/l`	`m~mol/L` (that is twice the value)
hardness in millimoles	`m~mol/L`, `mmol/L`	— it needs no unit of its own
hardness in US grains	`gpg`	`gr/gal` (a mass concentration)
conductivity	`µS/cm`, `mS/cm`, `dS/m`	— no base unit exists or is needed
conductivity, older US data	`µmho/cm`	— `mho` is the siemens
dissolved solids	`mg/L`	`ppm` (dimensionless 10^{-6})
turbidity, white light 90°	`NTU`	`FNU`, `FTU`
turbidity, near-IR 90°	`FNU`	`NTU`, `FTU`
turbidity, method unstated	`FTU`	`NTU` or `FNU` — those claim a geometry
turbidity, attenuation	`FAU`	`FNU` (a different optical quantity)
turbidity, candle method	`JTU`	any formazin scale
practical salinity	`PSU`	`‰` or `g/kg`
absolute salinity	`g/k~g`	`PSU`
hydroponic EC scale	`CF`	`cF` (centifarad)

See also

- Unit & Currency Reference — the registry these readings come from
- Unit & Currency Cheat Sheet — every symbol in one place
- Specification §11 — Units System — the normative unit rules
- FAQ §4 — Units — common unit questions

Every claim in this document was verified against the reference implementation. If you find a token whose behaviour surprises you and it is not listed here, that is a documentation bug — please report it.

End of Bovnar — Unit Ambiguities (Bovnar spec 1.1).