ReliaEn 1.0.0 User Manual · Chapter 6.14

Rebuild derived links and Consistency check — keeping the links between records whole

Menu: CMMS ▸ Rebuild derived links… and CMMS ▸ Consistency check…

---

# Part 1 — What the function does

**What it does.** A CMMS file holds more than assets: preventive maintenance (PM) schedules, failure mode and effects analysis (FMEA) studies, reliability block diagram (RBD) blocks, parts and more. ReliaEn records how they belong together as links. Two entries of the CMMS menu look after the links as a whole:

| Entry | What ReliaEn does |
|---|---|
| Consistency check… | Examines every valid link and shows, in a box titled Links, one line with the counts of dangling links, loops and duplicates, then one line for each problem found |
| Rebuild derived links… | Regenerates the derived links from their sources: the asset hierarchy and the PM schedules' assets. It adds any that are missing, retires any the sources no longer imply, and shows the counts in a box |

A link is **derived** when ReliaEn writes it from another record, **native** when it is written once as a link, and **bridge:** followed by a system's name when it came from an enterprise maintenance system. A **dangling** link is a valid link whose record at one end no longer exists. Links are never deleted, only retired, so the rebuild leaves the history in the file.

**From what data.** The open CMMS file. Neither entry asks for anything.

**When an engineer uses it.** After a large import or a restore, to confirm that the links are whole.

**Do not use it when**

| Situation | Use instead |
|---|---|
| You want to see what hangs off one asset, or retire one link | The tab Related of CMMS ▸ Assets… (chapter 6.1) |
| You want to move an asset to another parent | Move in CMMS ▸ Assets…; the hierarchy link follows |

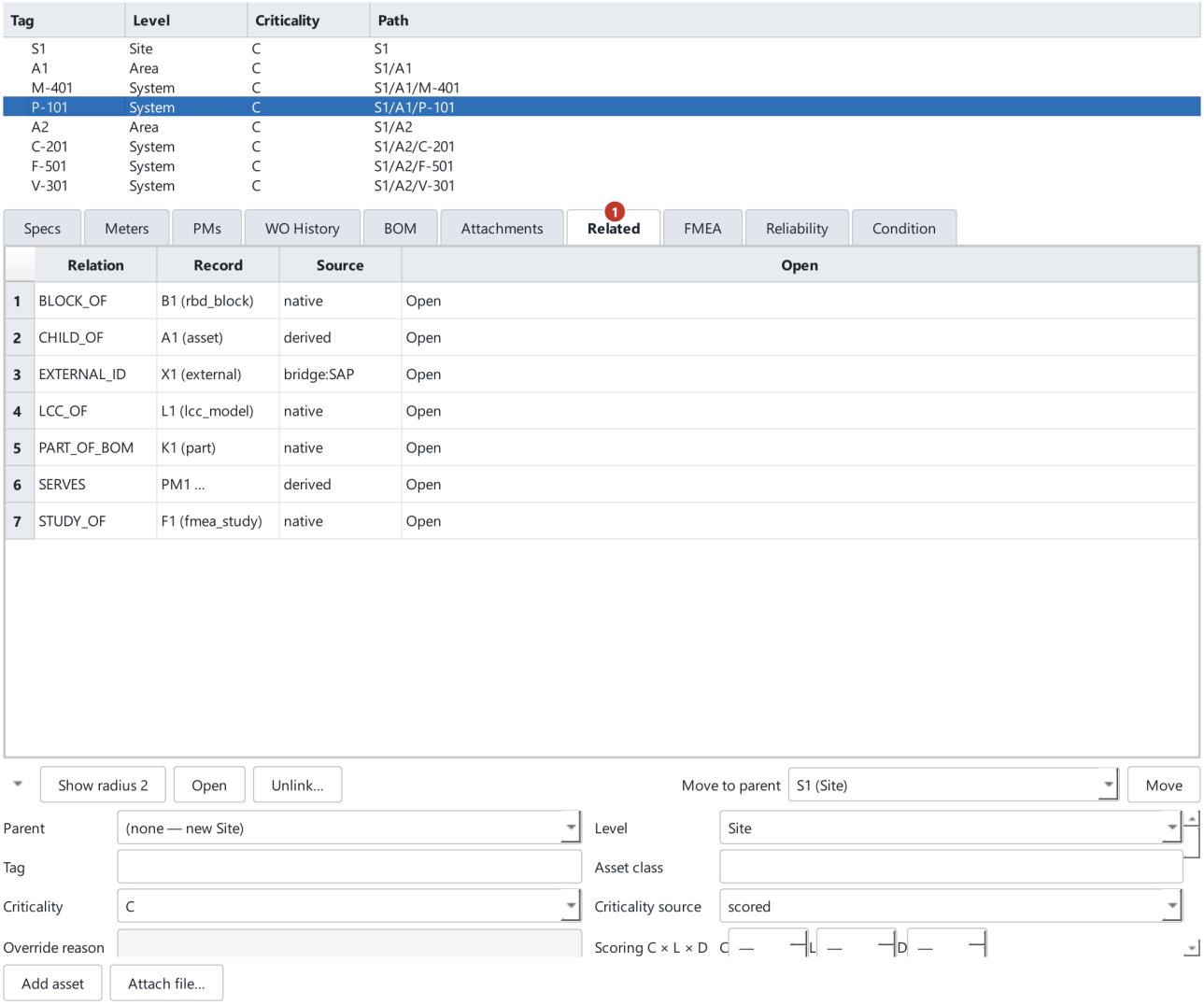
---

# Part 2 — The example's input data

*Setting.* Site S1 holds area A1 with P-101 and M-401, and area A2 with C-201, V-301 and F-501. Besides the eight assets the file holds eight further records, joined by seventeen valid links. The administrator checks the file after loading it, rebuilds the derived links, then checks a second file saved after a restore, in which a link from block B7 points to an asset that no longer exists.

| Files in the worked-example zip | |
|---|---|
| 6-Links/links_plant.reliaen | The sound file |
| 6-Links/links_plant_broken.reliaen | The same plant with one broken link |

The links of one asset, P-101, on the tab Related of CMMS ▸ Assets…:



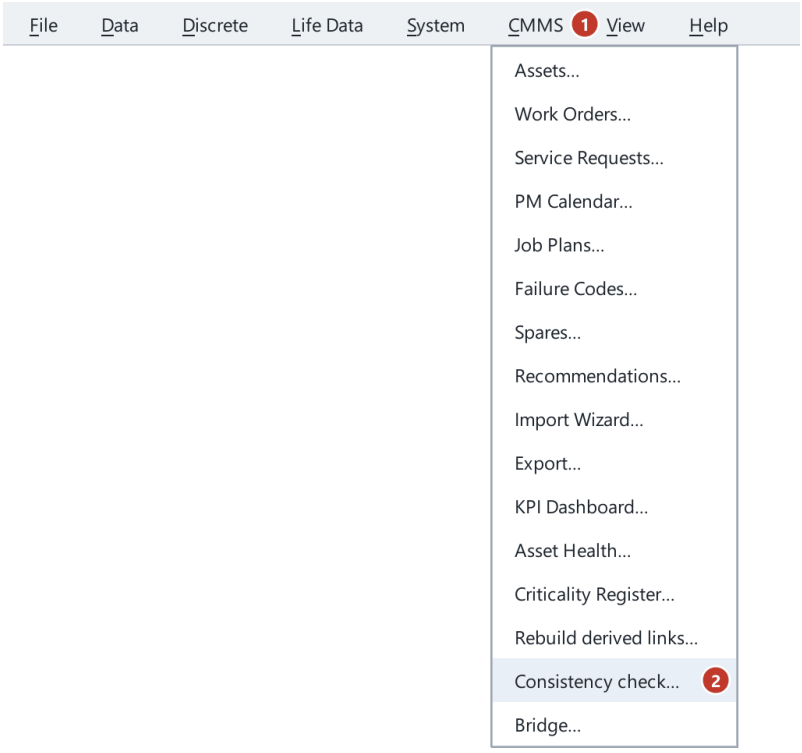

P-101's links

- ① The tab Related: seven links, each with its Relation, its Record and its Source.

---

# Part 3 — How to use it, step by step

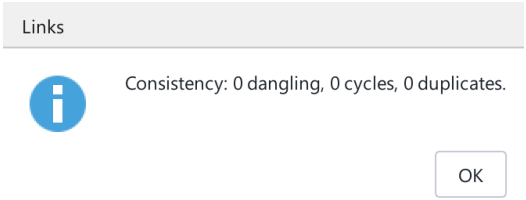
1. **Open a copy of links_plant.reliaen, then choose CMMS ▸ Consistency check…** in the menu bar.




Step 1 — the menu path

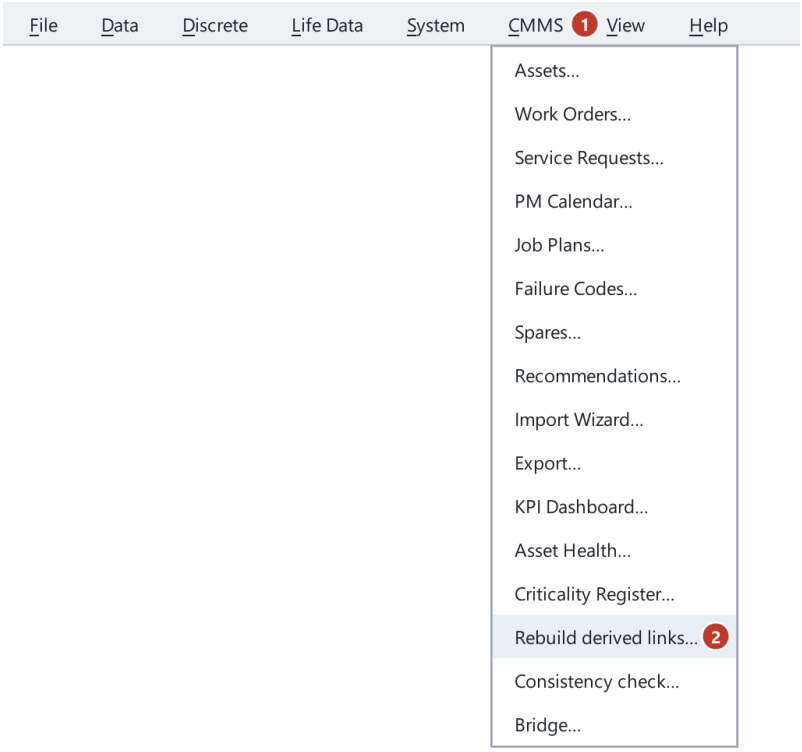
- ① CMMS: the menu.
- ② Consistency check…: the entry to click.

2. **Read the box, then click OK:**




Step 2 — the box

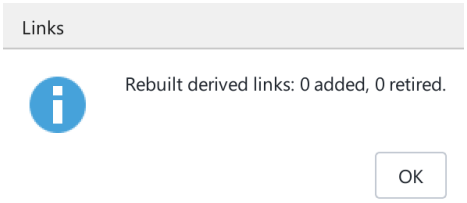
3. **Choose CMMS ▸ Rebuild derived links….**




Step 3 — the menu path

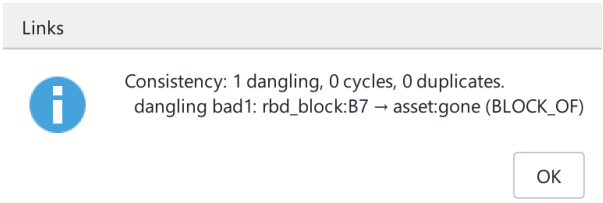
- ① CMMS: the menu.
- ② Rebuild derived links…: the entry to click.

4. **Read the box, then click OK:**




Step 4 — the box

5. **Open a copy of `links_plant_broken.reliaen` and choose `CMMS` ▸ `Consistency check…` again:**

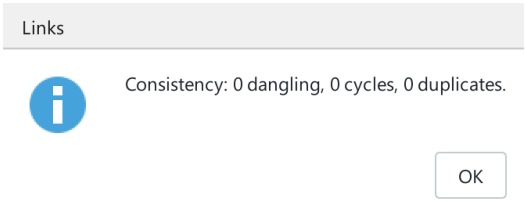



Step 5 — the box

---

# Part 4 — The result, picture by picture

The check on the sound file:

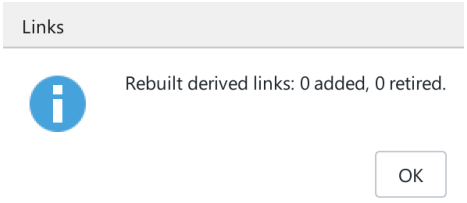



The check on the sound file

The box `Links` reads `Consistency: 0 dangling, 0 cycles, 0 duplicates.` Of the seventeen links, none points to a record that is gone, none closes a loop, and none is written twice.

**What to decide.** Nothing to repair. Run the check again after every restore or bulk import.

The rebuild:

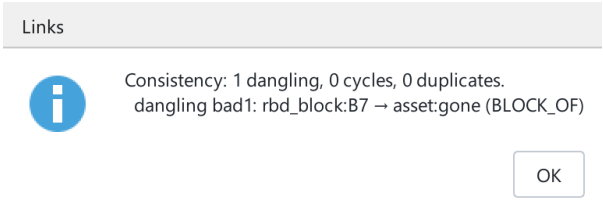



The rebuild

The box reads `Rebuilt derived links: 0 added, 0 retired.` Every derived link already stands as its source implies.

**What to decide.** A rebuild that reports anything else means the links had drifted from their sources: check what changed.

The check on the broken file:




The check on the broken file

The first line reads `Consistency: 1 dangling, 0 cycles, 0 duplicates.` The second names the problem: `dangling bad1: rbd_block:B7 → asset:gone (BLOCK_OF)`. That is the link's identifier `bad1`, its two ends, block B7 and an asset that is gone, and its relation.

**What to decide.** Find out which asset block B7 stood for before the restore. Until the block is mapped again, the system model holds a block with no asset behind it.

---

# Part 5 — How to interpret the results

**A line with any count above zero needs action.** Each problem line names the link and both of its ends.

**The three counts.** Dangling: a link to a record that no longer exists. Cycles: a loop in a relation that must not loop, such as an asset above itself in the hierarchy. Duplicates: the same link written twice.

**The check changes nothing.** It reads and reports. The rebuild writes, and only derived links: it does not repair a native link such as the one in the broken file.

**Derived links follow their sources.** To change one, change the asset or the schedule behind it; ReliaEn then writes the link.

**Links from an enterprise maintenance system stay as they came.** A link whose source begins `bridge:` is kept as that system delivered it.