

AR-880-JET — CPS-D Crash & Talkgroup Fix

A community fix for Abbree's CPS-D programming software (AR-880 DMR radio)

- **By:** N6JET (with Claude AI)
 - **Date:** May 28, 2026
 - **Target:** CPS-D **v1.0.45.0** (Abbree Customer Programming Software — Digital)
 - **Status:** Provided free, as-is, with no warranty.
-

TL;DR

Stock CPS-D v1.0.45.0 has two serious problems for AR-880 owners:

1. **It crashes** with an `ArgumentOutOfRangeException` whenever you try to open a saved codeplug file, which makes the radio very difficult to program.
2. **It won't accept DMR talkgroup IDs above 1024**, so common talkgroups — including 7-digit IDs like Brandmeister/TGIF groups — could not be programmed at all. This was a dealbreaker for amateur DMR use.

Both stem from the same underlying cause: UI spinner controls whose value ranges are far too narrow. This project documents the bugs, the root cause, and a working fix — and provides a **binary patch** you apply to *your own* copy of CPS-D so you never have to download a modified executable from anyone.

After the fix, codeplugs load without crashing **and** full-size DMR talkgroup IDs can be entered, saved, and used on-air. (Verified: TG 3100 programmed, written to the radio, and confirmed TX/RX through a hotspot.)

Important: This repository does **not** distribute Abbree's software. You download the official CPS-D from Abbree, then apply the patch below to the copy on your own machine. The patch file contains only the *changes*, not Abbree's code.

Who this is for

Owners of the Abbree AR-880 DMR radio who either can't load/save codeplugs because CPS-D crashes, or can't program DMR talkgroup IDs above 1024 (including 7-digit talkgroups). If you bought an AR-880 and found the programming software unusable out of the box, this is the fix.

The problems

Problem 1 — Crash on codeplug load

When you open a previously saved codeplug in CPS-D v1.0.45.0, the program throws an `ArgumentOutOfRangeException` and fails to load the file. Common error messages include:

- Value of '248' is not valid for 'Value'
- Value of '2026' is not valid for 'Value'
- Value of '27' is not valid for 'Value'
- Value of '1' is not valid for 'Value'
- Value of '-7' is not valid for 'Value'

How to reproduce:

1. Install stock CPS-D v1.0.45.0.
2. Create or open a codeplug and save it.
3. Close and reopen the saved codeplug file.
4. CPS-D throws the exception above and aborts the load.

Problem 2 — Talkgroup IDs capped at 1024

Stock CPS-D refuses to accept a DMR talkgroup ID greater than 1024. Because most real-world talkgroups use larger IDs — including 7-digit Brandmeister/TGIF talkgroups — they simply could not be programmed. For amateur DMR operators this was the single biggest reason to avoid the radio.

How to reproduce:

1. In stock CPS-D, open the digital contact / talkgroup list.
2. Try to enter a talkgroup ID above 1024 (e.g. `3100` , or a 7-digit ID).
3. The field rejects it / will not hold the value.

Root cause

CPS-D is a .NET Framework 4.5 WinForms application. Its UI uses many `NumericUpDown` spinner controls whose `Minimum` / `Maximum` properties are set to ranges that are **narrower than the values actually stored in codeplug files**. When the load routine assigns a stored value to a spinner, .NET rejects it because it falls outside the control's allowed range.

Specifically:

1. **Sentinel values** (e.g. `248`) stored in unused/reserved codeplug fields exceed spinner `Maximum` values that were set as low as 100, 255, etc.
2. **The current year** (e.g. `2026`) can exceed a year spinner's `Maximum` depending on load order.
3. **Range-setting order:** in places `set_Value` runs *before* `set_Maximum` / `set_Minimum` , so even a valid value is rejected because the range hasn't been widened yet.
4. **Negative-range spinners** (dBm power fields, roughly -120 to -80) need their negative `Minimum` preserved, while other minimums need to be relaxed.

5. **The talkgroup ID spinner** has a `Maximum` of 1024, far below the real DMR talkgroup ID range. This is the same class of bug as the others — an over-restrictive spinner `Maximum` — which is why raising the spinner maximums (below) lifts the 1024 cap as well as fixing the load crash.
-

The proper fix (for Abbree / source level)

If you're an Abbree developer reading this: the clean fix lives in your source, not in a patched binary. Any of the following resolves the load crash (ideally combine 1 and 2):

1. Set each `NumericUpDown`'s `Minimum` and `Maximum` **before** assigning its `Value` during codeplug load.
2. **Clamp** each incoming value into `[Minimum, Maximum]` before assignment (or guard the assignment).
3. **Widen** each spinner's range to match the true domain of its field, including reserved/sentinel values.

For the negative dBm spinners, **keep** the existing negative minimums (≈ -12 and -120); those are correct.

For the **talkgroup ID field specifically**, set the `Maximum` to at least 16,777,215 so the full DMR talkgroup ID range (including 7-digit Brandmeister/ TGIF IDs) can be entered. Also confirm the underlying codeplug field and the radio firmware accept the full range — see the verification note below, which indicates they do.

This patch has been emailed to Abbree (info@abbree.com). If they ship a fixed CPS-D, use theirs and you can ignore this entirely.

The community fix (binary patch to your own copy)

Because most of us don't have Abbree's source, the fix here is applied at the .NET CIL (bytecode) level inside `CPS-D.exe`. **No other files are modified** and the file size is unchanged (842,544 bytes).

What the patch changes

#	Change	Sites	Detail
1	Static <code>set_Maximum</code> raised to 99,999,999	23	Hardcoded maximums (orig. 495 / 510 / 1000) widened
2	Dynamic <code>set_Maximum</code> raised to 99,999,999	28	Runtime-computed maximums (orig. 255-16,777,215) widened; only element[0] of the <code>Decimal(int32[])</code> array changed, preserving the flags field
3	dBm spinner <code>set_Maximum</code> fixed	1	Offset <code>0x50540</code> : rebuilt negative <code>Decimal(-80)</code> into positive <code>Decimal(99,999,999)</code>
4	Positive dynamic <code>set_Minimum</code> removed	22	Replaced with <code>pop / pop / nop</code> (orig. minimums: 1, 5, 10, 15, 250, 500, 750, 1970)
5	Negative dynamic <code>set_Minimum</code> preserved	2	<code>0x3F236</code> (Min -12), <code>0x50575</code> (Min -120) intentionally kept
6	<code>numDay</code> day-of-month range branches removed	3	<code>0x3E3E9-0x3E3FA</code> , <code>0x3E3FC-0x3E40D</code> , <code>0x3E40F-0x3E425</code> NOPed (were Max 28/29/31 via <code>DaysInMonth()</code>)
7	Static <code>set_Minimum</code> redirects removed	5	<code>0x329ED</code> , <code>0x32A18</code> , <code>0x32A97</code> , <code>0x32AE9</code> , <code>0x32AFD</code> NOPed (orig. minimums 14/15)

Net effect: every spinner now accepts `Minimum` = -120 (dBm spinners) or 0 (all others), and `Maximum` = 99,999,999. This:

- eliminates the `ArgumentOutOfRangeException` crashes on codeplug load, regardless of sentinel or unexpected values; **and**
- lifts the talkgroup ID cap from 1024 to 99,999,999, so full-size DMR talkgroup IDs — including 7-digit Brandmeister/TGIF groups — can be entered.

Verified on-air: TG **3100** (a value impossible to enter in stock CPS-D) was programmed with the patched CPS, written to an AR-880, and confirmed working — successful **TX and RX through a DMR hotspot** on that talkgroup. So the fix is not just "the field accepts the number"; the value survives the write to the radio and keys up correctly on-air.

Note: raising all maximums to 99,999,999 and stripping the positive minimums is a deliberately blunt workaround that makes the software *usable*. It is not the "correct" fix Abbree should ship (see the source-level fix above). It does not introduce data corruption — the radio still validates values on write — but the spinners will no longer stop you from typing nonsense into a field, so program with care.

How to apply the fix

You need the **official** CPS-D v1.0.45.0 from Abbree, then you apply the patch to *your* copy.

Step 1 — Get the official CPS-D

Download CPS-D v1.0.45.0 from Abbree's website (<https://download.abbree.com/AR-880.rar> , also linked from the AR-880 product page). Extract it.

Step 2 — Verify you have the right base file

Confirm your unmodified `CPS-D.exe` matches the expected version before patching.

```
# Windows (PowerShell)
Get-FileHash .\CPS-D.exe -Algorithm SHA256

# Linux/macOS
sha256sum CPS-D.exe
```

Expected size: **842,544 bytes**. Expected SHA-256 of the **original** file: `<FILL IN: sha256 of stock CPS-D.exe v1.0.45.0>`

If your hash doesn't match, you have a different CPS-D version — these offsets won't apply and you should not patch blindly.

Step 3 — Apply the binary patch

This repo ships a binary diff (`cps-d-fix.xdelta`) that contains only the changes above — not Abbree's code. Apply it to your own `CPS-D.exe` :

```
# Using xdelta3
xdelta3 -d -s CPS-D.exe cps-d-fix.xdelta CPS-D-patched.exe

# (Windows builds of xdelta3 are available; or use bspatch with a .bsdifff patch)
```

Then verify the result:

```
sha256sum CPS-D-patched.exe
```

Expected SHA-256 of the **patched** file: `<FILL IN: sha256 of patched CPS-D.exe>`

Step 4 — Install

Back up your original `CPS-D.exe` , then replace it with `CPS-D-patched.exe` (rename to `CPS-D.exe`) in your CPS-D program folder, typically:

```
C:\Program Files\CPS-D\
```

The rest of the install is unchanged:

CPS-D.exe	(patched)
CPS-D.exe.config	(original, unchanged)
WinUSBNet.dll	(original, unchanged)
DotNetZip.dll	(original, unchanged)

Prefer to do it by hand? The change table above lists the affected metadata tokens, offsets, and original values so you can apply the edits yourself with dnSpy, a CIL editor, or a hex editor against your own copy. The binary diff is simply the convenient, fully-complete version of the same changes.

Technical notes

- CPS-D is a .NET Framework 4.5 WinForms application written in C#.
- The binary is obfuscated (class/method names replaced with Unicode).
- All patches are in-place CIL bytecode edits within the .NET assembly.
- No changes to metadata tables, the string heap, or assembly structure.
- File size unchanged (842,544 bytes).

CIL opcodes used in the patches

00	nop	no operation
20 xx	ldc.i4	load 32-bit integer constant
25	dup	duplicate top of stack
26	pop	discard top of stack
6F xx	callvirt	virtual method call
73 xx	newobj	create new object
9E	stelem.i4	store array element
2A	ret	return from method

Key .NET metadata tokens

0x0A00039C	NumericUpDown.set_Maximum
0x0A00038B	NumericUpDown.set_Minimum
0x0A00038C	NumericUpDown.set_Value
0x0A00038A	Decimal..ctor(int32[])
0x0A00031E	Decimal..ctor(int32)

Disclaimer & licensing

- This is an **unofficial, community-made fix**, provided **free of charge, as-is**, with **no warranty** of any kind. Use at your own risk. Always keep a backup of your original CPS-D installation.

- This project distributes **only a description of the bug and a binary diff of the changes**. It does **not** distribute Abbree's CPS-D software. You must obtain CPS-D from Abbree and apply the patch to your own legally-downloaded copy.
- CPS-D and the AR-880 are products of Abbree (Xiamen Aorui Electronic Co., Ltd.). This project is not affiliated with or endorsed by Abbree. All trademarks belong to their respective owners.
- If Abbree releases an official CPS-D that fixes this crash, use the official software instead.

A note from N6JET

The Abbree AR-880 is a cool radio that I and many other DMR amateurs were excited to own. Now the radio we bought and like so much actually works the way it should. Everyone wins.

73 de N6JET