DIADROM

UN R156 · READINESS CHECKLIST

UN R156 software-update readiness — the checklist

Fourteen questions that separate the teams with an easy type-approval conversation from the rest — for homologation-facing diagnostics and software managers.

Diadrom checklist · July 2026 · from the article by Jonas Hellberg, Head of Product Development

SOFTWARE UPDATE MANAGEMENT

UN R156 software-update readiness — the checklist

UN R156 requires a Software Update Management System for type approval. The update mechanism is the easy half. The record — which software ran on which unit, when it changed, on whose approval — is where readiness is decided.

Why a checklist, not a feature list

UN Regulation No. 156 requires a Software Update Management System (SUMS) for vehicle type approval: a process that knows which software versions are compatible with which vehicles, protects the integrity of updates, records every update carried out, and identifies software configurations — the RXSWIN identifier exists for exactly this. ISO 24089 describes the corresponding engineering practice.

The uncomfortable discovery for many organisations is that the hard part is not the update mechanism — it is the record. The fourteen questions below separate the teams that will have an easy conversation with their approval authority from those that will not. Answer each honestly; every ‘no’ is a work package.

A · The record — what SUMS actually stands or falls on

- Can you state, per vehicle, which software versions it currently runs — and prove it?
- Can you show which combinations of logical blocks (application, calibration, data sets) are approved to run together?
- Is every update recorded: what was installed, when, on whose approval?
- Are software configurations identified and traceable to type approval (RXSWIN)?

B · The mechanism — the reprogramming sequence itself

- Does your reprogramming follow the full UDS (ISO 14229) choreography — session control, security access, erase, RequestDownload (0x34), TransferData (0x36), RequestTransferExit (0x37), verification routines, reset?
- Does an interrupted flash — power loss mid-erase, a cable pulled mid-transfer — always leave the ECU recoverable, with the resident bootloader intact and the application validated at every boot?
- Is the flash driver downloaded to RAM per programming session, so no persistent write path to flash exists in normal operation?
- Are block dependencies enforced so mismatched combinations never run?

C · Security — inside the sequence, not around it

- Are software images signed, and verified by the bootloader before the application is ever marked valid?
- Are the verification keys anchored in hardware (an HSM) rather than in software?

- Do you have a path from classic 0x27 seed-and-key towards certificate-based Authentication (0x29, ISO 14229-1:2020)?

D · The process — one discipline, not three tools

- Are end-of-line programming, workshop reprogramming and OTA one managed process with one record — not three separate tools?
- Does your engineering practice follow ISO 24089, and can your suppliers hand you the evidence your approval authority will ask for?
- Is reflashing treated as a managed process with named ownership — not a tool someone runs?

The framing that keeps you honest

A bootloader and its tooling are built to support the R156 process — supporting a regulation is not a claim of certification, and any supplier who says otherwise should worry you. What matters is whether the record holds when the approval authority asks.

IF YOU KEEP FOUR THINGS

- R156 makes the update record — which software, which vehicle, whose approval — as decisive as the update mechanism itself.
- Recovery from an interrupted flash is designed in, not handled ad hoc: the bootloader never erases itself and validates the application at every boot.
- Authenticity beats access control: signed images verified before anything is marked valid, keys anchored in hardware.
- One managed reflashing process across production, workshop and OTA — with one record.

Building, replacing or qualifying an ECU flash bootloader — or bringing reflashing into serial production? Diadrom builds and maintains flash bootloaders in serial production (Autotech Bootloader), with the diagnostics discipline the R156 record depends on. Read the full article at **diadrom.com/insights/ecu-flash-programming-uds** or talk to us: **info@diadrom.se**.