# Formát ECID

## Pojmenování CAN zpráv a dekódování signálů v PP2CAN / LogAnalyzer – referenční manuál

*Popisuje chování parseru ECID (CANMsgDatabase.cpp) ve verzi s opravami z listopadu 2026.*

1. Úvod ..... 3
1.1 Kde se ECID používá ..... 3
2. Struktura souboru ..... 3
2.1 Zápis čísel ..... 3
2.2 Verze formátu (^ECID) ..... 4
3. Přehled řídicích znaků ..... 4
4. Pravidla pojmenování (>) ..... 5
4.1 Položky pravidla a šablony ..... 5
4.2 Jak se porovnává ID ..... 5
4.3 Jak se porovnávají data ..... 6
4.4 Pořadí: první shoda vyhrává ..... 6
5. Šablony a zásobník ..... 6
5.1 Skládání názvu ..... 6
5.2 Sekce { } ..... 6
5.3 + a - ..... 7
5.4 -- a ! ..... 7
6. Indikace velikosti (S) ..... 7
7. Signály (@ a :) ..... 8
7.1 Společné položky ..... 8
7.2 Číslování bitů ..... 8
7.3 Typy signálů ..... 9
7.4 Jaké signály (EYE) vzniknou ..... 10
7.5 Složené zprávy: T:ISOTP, T:ISOTPX a T:J1939 ..... 10
7.6 Signály nezávisí na pořadí pravidel ..... 12
7.7 Signály šablony (:) ..... 12
7.8 Speciální hodnoty (!) ..... 12
8. Seznamy hodnot [ ] ..... 12
9. Vkládání souborů (&) ..... 13
10. Vyhodnocení a výkon ..... 13
11. Chyby a log ..... 13
12. Kompletní příklady ..... 15
12.1 J1939 / FMS ..... 15
12.2 OBD-II přes ISO-TP ..... 15
12.3 Připravené knihovny ..... 16
13. Časté chyby a doporučení ..... 17
14. Převod DBC do ECID (dbc2ecid) ..... 17
14.1 Použití ..... 17

14.2 Co se převede ..... 17
14.3 Co se nepřevede (report) ..... 18
14.4 Příklad výstupu ..... 18
14.5 Doporučený postup a překlad ..... 18
Příloha A – rychlá reference ..... 19

# 1. Úvod

ECID (Extended CAN IDentification) je textový formát, kterým se popisuje, jak se mají pojmenovat přijaté CAN zprávy a které hodnoty (signály) se z nich mají dekódovat. Jeden soubor ECID tak nahrazuje dvě věci: tabulku názvů zpráv a definici signálů.

Formát nahrazuje starší CID, který uměl jen jedno pevné nastavení pojmenování pro všechny zprávy. ECID naproti tomu umí:

- jednotlivá ID, rozsahy ID, seznamy ID a masky ID (11bit i 29bit),
- podmínky na hodnoty datových bajtů DB0–DB7, včetně masek a rozsahů,
- šablony se zásobníkem, takže se společné podmínky píšou jen jednou,
- indikaci velikosti (např. délka ISO-TP rámce) přímo v názvu zprávy,
- signály typu unsigned, signed, hex, float, bit a text s měřítkem, offsetem, jednotkou a seznamem hodnot,
- vkládání dalších souborů (`&`), takže lze stavět knihovny (OBD-II, UDS, …).

## 1.1 Kde se ECID používá

- **Data view** a **filtr logu** – je-li zvoleno dekódování ID „ECID“, zobrazí se u každé zprávy název, který vrátí ECID.
- **Složené zprávy** – pravidla a signály s `T:ISOTP` nebo `T:J1939` vyhodnocuje Data view nad složenými daty vícerámcového přenosu (kapitola 7.5). Ostatní okna (Signal receiver, analyzér) pracují jen s jednotlivými CAN zprávami.
- **Signály (EYE)** – z každého řádku `@/:` vznikne jeden EYE signál. Signály jsou k dispozici v seznamu předdefinovaných signálů („Templates / ECID signals“) a zobrazují se v Data view a Signal receiveru.

> Pojmenování zprávy a signály jsou dvě nezávislé věci. Název určuje první vyhovující pravidlo `>`. Signály se ale vytvoří ze všech pravidel, i těch, která při pojmenování nikdy „nevyhrají“ (viz kapitola 7.6).

# 2. Struktura souboru

- Prostý text, doporučená přípona `.ecid`. Kódování ANSI/cp1250 (texty s diakritikou, `°C`). Konce řádků CRLF i LF.
- Každý řádek se zpracovává samostatně. Mezery a tabulátory na začátku a na konci řádku se ignorují, odsazovat lze libovolně.
- O významu řádku rozhoduje **první znak** (řídicí znak). Řádek, který nezačíná řídicím znakem, se tiše ignoruje.
- Řídicí znak musí být **na začátku nového řádku**. Na jednom řádku nelze mít dva řídicí znaky (např. `%1:x22,%2:xF1` je chyba).
- Položky na řádku se oddělují čárkou `,`, parametry položky dvojtečkou `:`. Desetinný oddělovač je tečka.
- Délka řádku není omezena.
- Komentář je řádek začínající `;`. Komentář na konci řádku za položkami není podporován.

## 2.1 Zápis čísel

| Zápis | Příklad | Význam |
|---|---|---|
| desítkově | `123` | Číslo 123. |
| hex s předponou x | `x7E8`, `X7e8` | Šestnáctkově 0x7E8 (doporučený zápis). |
| hex s předponou 0x | `0x7E8` | Také šestnáctkově. |
| obsahuje A–F | `7E8` | Bez předpony, ale obsahuje písmeno A–F → bere se jako hex. Pozor: `100` je desítkově! |

| Zápis | Příklad | Význam |
|---|---|---|
| rozsah | `xFD:FE` | Je-li první mez hex, druhá se čte také jako hex (`FE` = 0xFE). |

> Doporučení: ID, data a masky pište vždy s předponou `x`. Váhy, offsety a pozice bitů pište desítkově.

### 2.2 Verze formátu (^ECID)

Hlavní soubor může na začátku (za úvodními komentáři) uvést verzi formátu, pro kterou je napsán. Parser v této verzi programu podporuje verzi 1.5.

```
; UDS (ISO 14229-1) ...
^ECID:1.5
[UDS_NRC
```

Bez řádku ^ECID se hlavní soubor bere jako verze parseru (1.5). Vkládaný soubor bez ^ECID se bere jako verze hlavního souboru – nic se nehlásí.

Zapíše se do logu (kapitola 11), načítání ale pokračuje: soubor vyžaduje vyšší verzi, než parser podporuje; vkládaný soubor uvádí jinou verzi než hlavní soubor; ^ECID až za prvním pravidlem, seznamem nebo vložením; druhý ^ECID ve stejném souboru (platí první); chybný zápis nebo neznámá direktiva ^ (řádek se přeskočí).

Verze se porovnává číselně (major.minor, 1.10 je vyšší než 1.9). Starší verze programu řádek ^ECID přeskočí bez chyby. Všechny soubory knihovny ECID_knihovna.zip obsahují ^ECID:1.5.

## 3. Přehled řídicích znaků

| Znak | Název | Funkce |
|---|---|---|
| `;` | komentář | Řádek se ignoruje. |
| `>` | pravidlo | Vloží pravidlo pojmenování: kopie aktuální šablony + položky řádku + název `#`. |
| `%` | šablona | Doplní/změní položky aktuální šablony; `#text` připojí text k názvu šablony. |
| `+` | šablona se zálohou | Uloží šablonu na zásobník a pak funguje jako `%`. |
| `{` | začátek sekce | Uloží šablonu na zásobník. |
| `}` | konec sekce | Obnoví šablonu ze zásobníku. |
| `-` | obnovení | Obnoví šablonu ze zásobníku (jako `}`). |
| `--` | reset | Vyprázdní zásobník a vynuluje šablonu. |
| `!` | nulování části šablony | `!` celá šablona, `!N` název, `!S` velikost, `!L` seznam ID, `!I` ID/rozsah, `!M` maska ID, `!m` masky všech bajtů, `!0`–`!7` jeden bajt i s maskou. |
| `=` | název šablony | Mimo seznam hodnot: nastaví (přepíše) název šablony. |
| `[ ]` | seznam hodnot | Začátek `[Název` a konec `]` seznamu hodnot. Uvnitř jsou jen řádky `=`. |
| `=` | položka seznamu | Uvnitř `[ ]`: `=hodnota:text`, více položek oddělených čárkou. |
| `@` | signál | Signál (EYE) k bezprostředně předchozímu bezchybnému pravidlu `>`. |
| `:` | signál šablony | Signál k aktuální šabloně (ID a podmínky šablony, bez pravidla). |
| `&` | vložení souboru | Načte jiný soubor ECID na tomto místě. |

| Znak | Název | Funkce |
|---|---|---|
| ^ | verze formátu | ^ECID:major.minor – verze formátu souboru (kapitola 2.2). Píše se na začátek souboru. |
| ? | popis signálu | Popis bezprostředně předchozího signálu (@ nebo :), více řádků se spojí (kapitola 7.1). |

# 4. Pravidla pojmenování (>)

Pravidlo popisuje, kterým zprávám patří daný název. Skládá se z položek oddělených čárkou. **Název `#` musí být poslední položka** a je povinný. Pravidlo bez `#` se zahodí a ohlásí se chyba.

```
>I:x123,#Motor status
>I:x18FEF100,E,M:x3FFFF00,#CCVS
>R:x100:x10F,#Group 100-10F
>L:x200:x201:x205,#List 200/201/205
>I:x7E8,1:x41,2:x0C,#OBD engine speed response
>I:x300,m0:xF0,0:x20,#ISO-TP consecutive frame
>I:x400,0:x10:x1F,#DB0 range 10-1F
>I:x400,#Other 400
```

## 4.1 Položky pravidla a šablony

| Položka | Zápis | Význam |
|---|---|---|
| I | `I:id`<br>`I:*` | Jedno ID. `I:*` zruší ID (pravidlo platí pro každé ID). Zruší i seznam `L`. |
| R | `R:od:do` | Rozsah ID včetně mezí. Zruší seznam `L`. Chyba, je-li od > do. |
| L | `L:id1:id2:…` | Seznam ID. Má přednost před `I`/`R` ze šablony. |
| M | `M:maska`<br>`M:*` | Maska ID. Porovnávají se jen bity masky s hodnotou 1. Platí pro `I`, `R` i `L`. `M:*` masku zruší. |
| E | `E` | 29bitové (extended) ID. Bez `E` jde o 11bitové ID. Zrušit lze jen `!`. |
| 0 … 7 | `N:hodnota`<br>`N:od:do`<br>`N:*` | Podmínka na datový bajt DBN: přesná hodnota nebo rozsah (0–255). `N:*` podmínku zruší. |
| m0 … m7 | `mN:maska`<br>`mN:*` | Maska bajtu DBN. Použije se jen tehdy, když má bajt i hodnotu. Samotná maska nic nekontroluje. |
| S | `S:bit:délka[:B]` | Indikace velikosti – viz kapitola 6. |
| T | `T:ISOTP`<br>`T:ISOTPX`<br>`T:J1939`<br>`T:*` | Transportní vrstva – pravidlo a jeho signály platí pro složená data vícerámcového přenosu (kapitola 7.5). `T:ISOTPX` = ISO-TP s rozšířenou adresací. `T:*` vrací jednotlivé CAN zprávy. |
| # | `#text` | Název. Poslední položka, smí obsahovat čárky. Prázdné `#` je povoleno (zůstane jen název šablony). |

## 4.2 Jak se porovnává ID

- Nejprve musí sedět typ ID: pravidlo s `E` platí jen pro 29bitové zprávy, bez `E` jen pro 11bitové.
- Bez masky: `I` znamená shodu, `R` znamená od ≤ ID ≤ do, `L` znamená shodu s některým prvkem.
- S maskou se obě strany nejprve maskují: `(ID & M) == (I & M)`, u rozsahu `(od & M) ≤ (ID & M) ≤ (do & M)`.

- Pravidlo bez jakéhokoli ID (ani v šabloně) platí pro všechna ID daného typu.

Příklady masek:

| Zápis | Vyhovující ID |
|---|---|
| `I:x7E8,M:x7F8` | 7E8 – 7EF |
| `I:x18DA00F1,E,M:x1FFF00FF` | 18DAxxF1 (UDS fyzický požadavek, libovolná cílová adresa) |
| `I:x00FEF100,E,M:x3FFFF00` | J1939 PGN FEF1 s libovolnou prioritou a zdrojovou adresou (např. 18FEF100, 0CFEF117) |

### 4.3 Jak se porovnávají data

Kontrolují se jen bajty, které mají zadanou hodnotu nebo rozsah. Pro bajt N s maskou m platí: `(od & m) ≤ (DBN & m) ≤ (do & m)`. Bez masky je m = xFF.

- `m0:xF0,0:x20` – horní nibble DB0 musí být 2 (ISO-TP consecutive frame 0x20–0x2F).
- `m2:x01,2:x01` – bit 0 bajtu DB2 musí být 1 (např. lichý sub-function u SecurityAccess).
- `2:xFD:FE,3:x00:xFF` – DB2 v rozsahu 0xFD–0xFE a DB3 libovolný.

> Pravidla vidí jen bajty DB0–DB7. U zpráv CAN FD delších než 8 bajtů se podmínky na další bajty zadat nedají.

### 4.4 Pořadí: první shoda vyhrává

Pravidla se procházejí v pořadí, v jakém jsou v souboru (po rozbalení všech vložených souborů). Zpráva dostane název **prvního** pravidla, které vyhoví ID i datům. Konkrétní pravidla proto patří **před** obecná. Obecné pravidlo („vše ostatní“) se píše jako poslední, typicky se zástupným znakem `*`.

```
>I:x400,0:x10:x1F,#DB0 range 10-1F     ; 400 [15 ..] -> "DB0 range 10-1F"
>I:x400,#Other 400                     ; 400 [25 ..] -> "Other 400"
```

> Komentáře za `;` v ukázkách jsou jen pro čtenáře manuálu. V souboru ECID musí být komentář na samostatném řádku.

Když nevyhoví žádné pravidlo, název je prázdný.

## 5. Šablony a zásobník

Šablona je „rozpracované pravidlo“: soubor položek (ID, maska, E, bajty, velikost, název), které se automaticky přidají ke každému dalšímu pravidlu `>`. Pravidlo vznikne jako kopie aktuální šablony a jeho vlastní položky šablonu doplní nebo přepíší. Na začátku souboru je šablona prázdná.

### 5.1 Skládání názvu

Každé `#text` na řádku `%` nebo `+` připojí text k názvu šablony (s mezerou). Název pravidla je pak `název šablony + mezera + vlastní název`. Prázdné vlastní `#` mezeru nepřidá. Řádek `=text` název šablony přepíše a `!N` ho smaže.

### 5.2 Sekce { }

`{` uloží šablonu na zásobník, `}` ji obnoví. Vše, co se uvnitř sekce do šablony přidá, za `}` zmizí. Sekce lze libovolně vnořovat.

```
--
%I:x500
%m0:xF0,0:x00,#SF
%S:0:4
{
%1:x10,#Session
>2:x01,#default
>2:x03,#extended
>2:*,#other
}
>1:x3E,#TesterPresent
```

| Zpráva 0x500 | Název |
|---|---|
| `02 10 03` | `SF Session extended size:2` |
| `02 10 05` | `SF Session other size:2` |
| `01 3E` | `SF TesterPresent size:1` |
| `12 10 …` | (prázdný – DB0 & F0 ≠ 0) |

### 5.3 + a -

`+` je zkratka pro `{` následované `%`. Uložený stav se obnoví pomocí `-` (nebo `}`).

```
--
%I:x700,#Base
+1:x01,#A
>2:x01,#x
-
>2:x02,#y
```

Zpráva 700 `00 01 01` dostane název `Base A x`, zpráva 700 `00 xx 02` název `Base y`.

### 5.4 -- a !

`--` vyprázdní zásobník i šablonu a začíná se „od nuly“. Hodí se jako oddělovač nezávislých bloků (např. jednotlivých ID v hlavním souboru). `!` nuluje jen vybranou část šablony:

```
--
=Fixed name
%I:x600
>0:x01,#one          ; 600 [01] -> "Fixed name one"
!N
>0:x02,#two          ; 600 [02] -> "two"
```

> Neuzavřená sekce `{` na konci souboru i `}` nebo `-` s prázdným zásobníkem se hlásí jako chyba.

## 6. Indikace velikosti (S)

`S:bit:délka[:B]` řekne, kde je ve zprávě uložena délka přenosu. Hodnota se přečte stejně jako unsigned signál (bit = pozice LSB, délka v bitech, `B` = Motorola, jinak Intel) a k názvu zprávy se připojí `size:N`.

| Použití | Zápis | Příklad názvu |
|---|---|---|
| ISO-TP Single Frame | `S:0:4` | `02 10 03` → `… size:2` |
| ISO-TP First Frame | `S:8:12:B` | `10 14 62 …` → `… size:20` |

Platné hodnoty: bit 0–63, délka 1–64; u Intelu musí pole skončit do bitu 63 (bit + délka ≤ 64). Chybná položka `S:` v šabloně šablonu nezmění (platí předchozí hodnota). `!S` indikaci v šabloně vypne.

# 7. Signály (@ a :)

Řádek `@` definuje signál – hodnotu, která se ze zprávy dekóduje a zobrazí. Patří k **bezprostředně předchozímu bezchybnému pravidlu** `>` a přebírá jeho ID, masku, typ ID a podmínky na data. Má-li předchozí pravidlo chybu, signál se odmítne. Signál se tak nikdy omylem nepřipojí k jiné zprávě.

Řádek `:` má stejnou syntaxi, ale patří k **aktuální šabloně**. Hodí se pro hodnoty společné celé skupině pravidel (např. SID a NRC v negativní odpovědi). Šablona musí mít ID.

```
[OnOff
=0:Off,=1:On
=2:Error,=3:N/A
]
>I:x00FEF100,E,M:x3FFFF00,#CCVS
@U:8:16:L:0.00390625:0,*km/h,#Wheel speed
@H:24:2:L,$OnOff,#CC active
@B:7:N,#Bit7 negated
```

## 7.1 Společné položky

| Položka | Význam |
|---|---|
| `#text` | Název signálu. Bez ? se popisem signálu stane název posledního pravidla (zobrazí se v tooltipu Data view). |
| `##` | Název signálu = název posledního pravidla. |
| `*text` | Jednotka (např. `*km/h`, `*°C`). |
| `?text` | Popis signálu (tooltip v Data view, okno EYE). Musí být poslední položka a smí obsahovat čárky. Delší popis pište na další řádky ?. |
| `$jméno` | Seznam hodnot, který musí být definován dřív (kapitola 8). |
| `!hodnota:text`<br>`!od..do:text`<br>`!J` | Speciální surová hodnota (např. N/A, error) – zobrazí se jen text a hodnota se nepočítá do min/max/průměru (kapitola 7.8). Položek `!` může být víc. |
| typ | Právě jedna z položek `U S H F B T` (viz 7.3). Bez typu je chyba „Unknown data type“. |

Popis lze psát i na samostatné řádky začínající znakem ? hned za signálem (@ nebo :). Několik řádků ? se spojí mezerou, případně za popis z řádku signálu. Komentáře (;) mezi nimi nevadí; jakýkoli jiný řádek popis signálu ukončí. Řádek ? bez předchozího signálu se zapíše do logu a přeskočí. U signálu s rozsahem nebo seznamem ID dostanou popis všechny jeho kopie.

```
@U:24:16:L:0.125:0,*rpm,#Engine speed,?SPN 190
?Actual engine speed, calculated over
?a minimum crankshaft angle of 720 degrees.
```

Délka popisu není v ECID omezena. Tooltip Data view zobrazí asi 480 znaků; při uložení do .sye se popis zkrátí na 255 znaků.

Na rozdíl od pravidla `>` nezáleží u `@` na pořadí položek.

## 7.2 Číslování bitů

Pozice bitu se počítá jako `bajt × 8 + bit v bajtu`, kde bit 0 je nejméně významný bit bajtu. `bit` u signálu vždy označuje **nejméně významný bit (LSB)** signálu, a to i u Motoroly.

| Bajt | b7 | b6 | b5 | b4 | b3 | b2 | b1 … b0 |
|---|---|---|---|---|---|---|---|
| DB0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 … 0 |
| DB1 | 15 | 14 | 13 | 12 | 11 | 10 | 9 … 8 |
| DB2 | 23 | 22 | 21 | 20 | 19 | 18 | 17 … 16 |
| DB3 | 31 | 30 | 29 | 28 | 27 | 26 | 25 … 24 |
| DB4 | 39 | 38 | 37 | 36 | 35 | 34 | 33 … 32 |

| Bajt | b7 | b6 | b5 | b4 | b3 | b2 | b1 … b0 |
|---|---|---|---|---|---|---|---|
| DB5 | 47 | 46 | 45 | 44 | 43 | 42 | 41 … 40 |
| DB6 | 55 | 54 | 53 | 52 | 51 | 50 | 49 … 48 |
| DB7 | 63 | 62 | 61 | 60 | 59 | 58 | 57 … 56 |

### Intel (L, little endian)

Od LSB se bity berou nahoru; po bitu 7 bajtu se pokračuje bitem 0 **následujícího** bajtu. 16bitová hodnota v DB1 (LSB) a DB2 (MSB) má zápis `U:8:16:L`.

### Motorola (B, big endian)

Od LSB se bity berou nahoru; po bitu 7 bajtu se pokračuje bitem 0 **předchozího** bajtu. Pro hodnotu zarovnanou na bajty v DBa (MSB) … DBb (LSB) platí `bit = b × 8`.

| Hodnota | Zápis |
|---|---|
| 16 bitů, DB3 = MSB, DB4 = LSB (OBD otáčky) | `@U:32:16:B:0.25:0` |
| 32 bitů, DB3 … DB6 | `@H:48:32:B` |
| 12 bitů: DB1 bity 0–3 (MSB) + celý DB2 (LSB) | `@S:16:12:B:1:0` |
| ISO-TP délka FF: DB0 bity 0–3 + DB1 | `S:8:12:B` |

> Pozor na rozdíl oproti DBC: v DBC je u Motoroly startovní bit MSB. V ECID (i v EYE) je to vždy LSB.

## 7.3 Typy signálů

| Typ | Zápis | Popis |
|---|---|---|
| U | `U:bit:délka:L\|B:váha:offset` | Unsigned celé číslo. Všech 5 parametrů je povinných. |
| S | `S:bit:délka:L\|B:váha:offset` | Signed (dvojkový doplněk, znaménko = nejvyšší bit signálu). |
| H | `H:bit:délka:L\|B` | Zobrazení v hex (stavová slova, DTC, ID). Váha a offset se ignorují, i když jsou uvedeny. |
| F | `F:bit:32\|64:L\|B:váha:offset` | IEEE 754 float (32) nebo double (64). |
| F (x) | `F:bit:x:exponent:mantisa:L\|B:váha:offset` | Uživatelský float se zadaným počtem bitů exponentu a mantisy. |
| B | `B:bit[:N]` | Jeden bit. `N` = negace (stav „zapnuto“ odpovídá hodnotě 0). |
| T | `T:bajt:počet[:R]` | Text (ASCII). Počítá se v **bajtech** od DB0. `R` = obrácené pořadí bajtů. |
| T (*) | `T:*:index` | Pole textu odděleného `*` (J1939 VI, DI, CI, ECUID: `pole0*pole1*`), index od 0. S `T:J1939` funguje pro jednu zprávu i složený přenos (BAM). |

| Typ | Zápis | Popis |
|---|---|---|
| J | `J:bit`<br>`J:bit:verze` | J1939 DTC – 32 bitů Intel (SPN, FMI, CM, OC) → `SPN 100 FMI 3 OC 1`. Bit CM = 0: SPN verze 4. Bit CM = 1: SPN ve verzi 1, 2 nebo 3 (J1939-73 z roku 1996), kterou z dat poznat nelze – zadejte ji jako `verze` (1, 2, 3). Bez verze se zobrazí SPN podle verze 1 a v závorce SPN pro verze 2 a 3: `SPN 100 FMI 1 OC 3 (CM1 v1; v2: SPN 24580, v3: SPN 265216)`; podle známého SPN se pozná, která verze platí. |
| P | `P:bit` | Reference na PGN – 24 bitů Intel → `PGN FECA00`. |
| O | `O:bit` | OBD DTC (SAE J2012) – 2 bajty Motorola, `bit` = LSB → `P0301`. |
| D | `D:bit` | UDS DTC (ISO 14229) – 3 bajty Motorola, `bit` = LSB → `P0301-17` (poslední bajt = typ chyby). |

Fyzická hodnota se počítá jako:

```
hodnota = raw × váha − offset
```

Offset se **odečítá**. Teplota s offsetem –40 °C (raw 0 = –40 °C) se tedy píše `U:24:8:L:1:40`. Rok J1939 TD (raw 0 = 1985) se píše `U:40:8:L:1:-1985`.

> Rozdíl oproti DBC: v DBC platí `fyz = raw × factor + offset`. Při převodu z DBC je proto potřeba změnit znaménko offsetu.

## 7.4 Jaké signály (EYE) vzniknou

- Pravidlo s jedním ID: jeden EYE se stejným ID, maskou a typem ID.
- Pravidlo se seznamem `L` nebo rozsahem `R`: jeden EYE pro **každé** ID. ID, která jsou po maskování shodná, se vytvoří jen jednou. Nejvýše 4096 ID na signál, zbytek se ohlásí a ignoruje.
- Podmínky na data se přenesou do EYE (kontrola bajtu s maskou), takže se signál zobrazí jen u zpráv, které podmínkám vyhoví.
- **Rozsah bajtu** (`N:od:do`) EYE neumí. Bajt se pak nekontroluje vůbec a parser vypíše varování.
- Bez ID (ani v šabloně) nelze signál vytvořit – chyba.

## 7.5 Složené zprávy: T:ISOTP, T:ISOTPX a T:J1939

Odpověď delší než jedna CAN zpráva se posílá po částech. Položka `T:` v pravidle nebo šabloně říká, že pravidlo a jeho signály platí pro **složená data** celého přenosu. Skládání dělá Data view.

| Zápis | Význam |
|---|---|
| `T:ISOTP` | ISO-TP (ISO 15765-2): Single Frame, First Frame + Consecutive Frames, klasický CAN i CAN FD. Bajt 0 = první bajt za PCI (u diagnostiky SID). Pozice signálů se počítají od něj a mohou ležet i daleko za 8. bajtem (max. 8192 bajtů). |
| `T:ISOTPX` | ISO-TP s rozšířenou / smíšenou adresací (ISO 15765-2 extended / mixed addressing): bajt 0 CAN zprávy je adresa (N_TA cílové ECU, u odpovědi adresa testeru), PCI a data začínají bajtem 1. Jedna klasická zpráva tak nese 6 bajtů dat místo 7. Přenosy na stejném CAN ID s různou adresou se skládají zvlášť. Bajt 0 složených dat = SID jako u `T:ISOTP`. Používá např. BMW (tester 6F1, ECU 6xx). |
| `T:J1939` | J1939 TP (BAM, RTS/CTS). Pozice platí pro data PGN – u jedné zprávy (8 B) i u složeného přenosu stejně. PGN se automaticky přidá do seznamu skládaných PGN. |

| Zápis | Význam |
|---|---|
| `T:*` | Zpět na jednotlivé CAN zprávy (výchozí stav). |

```
--
%I:x7E8,M:x7F8,#UDS Resp
{
%T:ISOTP
>0:x62,1:xF1,2:x90,#VIN
@T:3:17,#VIN
>0:x59,1:x02,#DTC by status
@D:40,#DTC 1
@D:72,#DTC 2
>0:x49,1:x17,#OBFCM
@U:144:32:B:0.01:0,!xFFFFFFFF:N/A,*l,#Lifetime fuel
}
--
>I:x00FECA00,E,M:x3FFFF00,T:J1939,#DM1
@J:16,#DTC 1
@J:48,#DTC 2
>I:x00FE6B00,E,M:x3FFFF00,T:J1939,#DI Driver Identification
@T:*:0,#Driver 1 identification
@T:*:1,#Driver 2 identification
--
{
%I:x6F1,#Diag Req
%T:ISOTPX
>0:x22,#ReadDataByIdentifier
}
{
%I:x600,M:x700,#Diag Resp
%T:ISOTPX
>0:x62,1:xF1,2:x90,#VIN
@T:3:17,#VIN
}
```

Jak se to chová v Data view:

- ISO-TP se skládá jen pro ID, ke kterému existuje signál `T:ISOTP` / `T:ISOTPX` nebo (při dekódování názvů ECID) pravidlo `T:ISOTP` / `T:ISOTPX`. Signály EYE ani DBC k tomu nejsou potřeba – pro samotné názvy a data stačí ECID. Hodnoty signálů se zobrazí, když jsou signály načtené jako EYE. O způsobu adresace rozhoduje první signál, resp. první pravidlo pro dané ID.
- Totéž platí pro J1939 TP: PGN pravidel `T:J1939` se při dekódování názvů ECID přidají do seznamu skládaných PGN automaticky (i bez EYE) a po načtení jiného ECID se seznam obnoví. Řádek `MULTI PGN` pak má název, délku a složená data.
- Chyby přenosu: vypadlá, prohozená nebo zdvojená Consecutive Frame přenos zahodí; Consecutive Frame bez First Frame se ignoruje; nový Single Frame nebo First Frame rozpracovaný přenos nahradí; mezera mezi zprávami přenosu delší než 2 s (nebo čas jdoucí zpět – nový log) přenos zahodí. Neúplný přenos nic neaktualizuje.
- Složená zpráva má vlastní řádek `ISO-TP 7E8` (u `T:ISOTPX` s adresou, např. `ISO-TP 612 [F1]`) s délkou a daty. Signály ISO-TP se zobrazují jen pod tímto řádkem. Jednotlivé CAN zprávy (SF/FF/CF/FC) mají dál své řádky.
- Při dekódování názvů ECID má každé pravidlo (služba, DID) vlastní řádek: čtení VIN, sériového čísla, DTC i negativní odpověď `7F … 78` (ResponsePending) zůstanou vidět vedle sebe a pozdější odpověď nepřepíše dřívější. Zprávy bez pravidla mají řádek podle prvního bajtu (SID). Bez ECID je jeden řádek na ID.
- Pravidla `T:ISOTP` se při pojmenování jednotlivých CAN zpráv přeskakují a naopak – obě sady se navzájem neovlivní.
- Podmínky na bajty `0`–`7` a `m0`–`m7` se u `T:` vztahují k prvním 8 bajtům složených dat. U ISO-TP se uplatní i maska standardního ID (např. `I:x7E8,M:x7F8` = 7E8–7EF).
- Signál, který leží za koncem přijatých dat (např. DTC 5 v odpovědi se 3 chybami), se neaktualizuje.

- Signál pro maskované ID (7E8–7EF, 18DAF1xx) má pro každou ECU (skutečné ID, u `T:ISOTPX` i adresu) a každý řádek vlastní kopii. Odpovídá-li více ECU, hodnoty se nemíchají a každá je pod řádkem své ECU – v normálním režimu i v režimu replikace.

> Bez `T:` se signál dekóduje z jedné CAN zprávy. Pro diagnostiku přes ISO-TP bez `T:ISOTP` mají signály smysl jen v Single Frame (pozice včetně bajtu PCI).

## 7.6 Signály nezávisí na pořadí pravidel

Pravidlo, které při pojmenování nikdy nevyhraje (protože před ním je obecnější pravidlo se stejným ID), přesto vytvoří své signály. Toho se dá využít: názvy lze definovat jednou a signály zvlášť. Zároveň je ale třeba hlídat, aby stejný signál nevznikl dvakrát.

```
>I:x00FEF100,E,M:x3FFFF00,#CCVS
>I:x00FEF100,E,M:x3FFFF00,#CCVS signals
@U:8:16:L:0.00390625:0,*km/h,#Wheel speed
```

Zpráva se jmenuje `CCVS`, signál „Wheel speed“ přesto existuje.

## 7.7 Signály šablony (:)

```
{
%1:x7F,#Negative response
:H:16:8:L,$UDS_SID,#Rejected SID
:H:24:8:L,$UDS_NRC,#NRC
>3:x78,#ResponsePending
>3:*,#
}
```

Oba signály vzniknou jednou pro celou skupinu, ne pro každé pravidlo zvlášť.

## 7.8 Speciální hodnoty (!)

Mnoho protokolů vyhrazuje některé surové hodnoty pro „nedostupné“ nebo „chyba“ (J1939 FE/FF, OBFCM FFFFFFFF). Položka `!` na řádku signálu je označí: Data view zobrazí jen text a hodnotu nezapočítá do min/max/průměru ani nepřepočte váhou.

```
@U:144:32:B:0.01:0,!xFFFFFFFF:N/A,*l,#Lifetime fuel
@U:0:8:L:1:40,!xFE:error,!xFF:not available,*°C,#Coolant
@U:8:16:L:0.00390625:0,!J,*km/h,#Wheel based speed
```

- `!hodnota:text` nebo `!od..do:text`. Bez textu se zobrazí `N/A`. Porovnává se **surová** hodnota (před váhou a offsetem), u signed bitový vzor signálu.
- `!J` doplní pravidla J1939-71 podle délky signálu: 2 bity 2/3, 4 bity E/F, 8/16/24/32 bitů FE…/FF… = error / not available.
- Speciální hodnoty se ukládají i do souboru .eye.

# 8. Seznamy hodnot [ ]

Seznam hodnot přiřazuje surovým hodnotám signálu texty. Signál se na seznam odkazuje položkou `$jméno`.

```
[Switch
=0:switched off,=1:switched on
=x2:error: really
=3:not available
]
```

- `[jméno` začíná seznam a `]` ho končí. Uvnitř jsou jen řádky `=`. Jiný řídicí znak v seznamu je chyba.
- Položka má tvar `hodnota:text`. Hodnota může být desítkově i hex (`x2`). Text se odděluje **první** dvojtečkou, takže sám dvojtečky obsahovat smí (`error: really`).
- Na jednom řádku může být víc položek oddělených čárkou. Další položky mohou, ale nemusí začínat `=`. Text proto **nesmí obsahovat čárku**.
- Seznam musí být definován **před** prvním použitím. Seznamy jsou globální a platí i ve vložených souborech.

- Hodnota, která v seznamu není, se zobrazí jen jako číslo.
- **Rozsah** `=od..do:text` (např. `=x80..xFF:manufacturer specific`). Rozsah do 256 hodnot se rozepíše na jednotlivé hodnoty (funguje všude), větší rozsah se uloží jako rozsah a zobrazuje ho Data view. Přesná hodnota má přednost před rozsahem a rozsah do 256 hodnot před větším rozsahem, z větších rozsahů vyhrává užší – nezávisle na pořadí v seznamu. U překrývajících se rozsahů do 256 hodnot vyhrává ten zapsaný dřív.

# 9. Vkládání souborů (&)

```
&OBD2_resp.ecid
&C:\ECID\common\lists.ecid
```

- Relativní cesta se počítá od složky souboru, ve kterém je `&` napsáno. Oddělovač může být `\` i `/`.
- Absolutní cesta (`C:\…`, `\\server\…`, `/…`) se použije beze změny.
- Vložený soubor se zpracuje přesně na místě `&`, se **stejnou šablonou a zásobníkem**. Může tedy šablonu využívat (např. tělo služeb vložené do bloku s ID), ale také ji měnit. Proto by měl mít vyvážené `{ }` a neměl by obsahovat `--`.
- Stejný soubor lze vložit vícekrát (např. tělo UDS pro 11bit i 29bit ID).
- Vnoření nejvýše 16 úrovní. Hlubší vnoření (typicky cyklus) se ohlásí jako chyba.

Typické použití – jedno tělo pro více ID:

```
--
%I:x7E0,M:x7F8,#UDS Req
&UDS_req.ecid
--
%I:x7E8,M:x7F8,#UDS Resp
&UDS_resp.ecid
--
%I:x18DAF100,E,M:x1FFFFF00,#UDS Resp
&UDS_resp.ecid
--
```

# 10. Vyhodnocení a výkon

Při zobrazení zprávy se název hledá takto:

1. Hledání v cache podle ID a jen těch bitů dat, podle kterých se pro toto ID skutečně rozhoduje.
2. Pokud zpráva v cache není, pravidla se projdou v pořadí ze souboru. Vyhraje první, které vyhoví typu ID, ID a datům.
3. Má-li pravidlo `S`, přečte se velikost a k názvu se připojí `size:N`.
4. Výsledek (i „nic nenalezeno“) se uloží do cache.

Doporučení pro rychlé a přehledné soubory:

- Rozdělujte podle ID co nejdřív (bloky `--` / `%I:` v hlavním souboru). Pravidla jiného ID se pak při rozhodování vůbec nezkoumají.
- Podmínky na data omezte jen na bajty, které opravdu rozlišují zprávy. Méně bajtů znamená menší cache.
- Velké rozsahy ID (`R:0:xFFFFFFFF`) u signálů nepoužívejte. Vytvořily by tisíce EYE. Použijte masku.

# 11. Chyby a log

Chyby při načítání se zapisují do souboru `<soubor>.ecid.log` vedle hlavního souboru. Jsou-li nějaké chyby, log se po načtení automaticky otevře v Poznámkovém bloku. Každý záznam obsahuje soubor, číslo řádku, text řádku a popis chyby. Chybné pravidlo nebo signál se přeskočí, zbytek souboru se načte.

```
Filename: C:\ECID\err.ecid Line: 1
>I:x100
Name (#) not found, the rule is ignored
---------------
Filename: C:\ECID\err.ecid Line: 2
@U:0:8:L:1:0,#Orphan
An identifier, list of identifiers, or range of identifiers was not specified (or the previous
> line has an error)
---------------
```

| Hlášení | Příčina / řešení |
|---|---|
| Name (#) not found, the rule is ignored | Pravidlo `>` bez `#`. Doplňte název, může být i prázdný `#`. |
| An identifier … was not specified (or the previous > line has an error) | `@` bez platného předchozího pravidla, nebo `:` v šabloně bez ID. |
| Unknown parameter: … (control character must be at the beginning of a new line) | Neznámá položka, nebo druhý řídicí znak na stejném řádku. |
| Unknown parameter (ignored) | Neznámá položka na řádku `@`. Položka se přeskočí. |
| Unknown data type | Na řádku `@` chybí typ `U S H F B T`. |
| Wrong number of parameters for … | Chybí parametry typu (např. U potřebuje bit:délka:endian:váha:offset). |
| Wrong FLOAT type paramater (32, 64 or x) | U typu F je délka jiná než 32, 64 nebo x. |
| Data byte: value out of range 0..255 or from > to | Hodnota bajtu mimo 0–255 nebo obrácený rozsah. |
| Data mask: value out of range 0..255 | Maska bajtu mimo 0–255. |
| Range: from > to | `R` s obrácenými mezemi. |
| Size: wrong first bit or length | `S` mimo rozsah (bit 0–63, délka 1–64). |
| Not found name section. | `$jméno` odkazuje na seznam, který ještě nebyl definován. |
| Naming section - … | Chyba v seznamu hodnot: chybí `]`, `]` bez `[`, prázdný seznam, nebo nepovolený znak uvnitř. |
| Stack is empty | `}` nebo `-` bez odpovídajícího `{` / `+`. |
| Section { not closed with } at the end of file | Neuzavřená sekce. |
| Unknown command | Neznámá varianta `!` nebo `-`. |
| File not found / File include error | Soubor nebo vložený soubor nejde otevřít. Zkontrolujte cestu. |
| Include nested too deep (cyclic include?) | Více než 16 úrovní `&`, obvykle cyklus. |
| Too many identifiers for one signal (max 4096) | Signál u příliš velkého rozsahu nebo seznamu ID. |
| Warning: data byte range is not supported for signals | Signál u pravidla s rozsahem bajtu. Bajt se v EYE nekontroluje. |
| STRING field index out of range (T:*:0..127) | Index pole u `@T:*:index` je větší než 127. |
| Unknown transport (T:ISOTP, T:ISOTPX, T:J1939 or T:*) | Neznámá hodnota položky `T:`. |
| J1939 DTC SPN version for CM=1 must be 1, 2 or 3 | U `@J:bit:verze` je verze mimo 1–3. |
| Wrong number of parameters for DTC/PGN | U `@J`, `@P`, `@O`, `@D` chybí pozice bitu. |
| Naming section - range from > to | Obrácený rozsah v seznamu hodnot (`=od..do:text`). |

| Hlášení | Příčina / řešení |
|---|---|
| Special value: … | Chybná položka `!` (chybí hodnota, obrácený rozsah, `!J` u nepodporované délky). |
| File requires ECID x.y, parser supports 1.5 - some items may not be loaded correctly | Soubor je pro novější verzi formátu. Aktualizujte program, nebo počítejte s tím, že některé položky se nenačtou. |
| Included file has ECID version x.y, main file a.b | Vkládaný soubor uvádí jinou verzi než hlavní soubor. Sjednoťte ^ECID, nebo ho ve vkládaném souboru vynechte. |
| ECID version should be at the beginning of the file (before rules, lists and includes) | ^ECID je až za pravidlem, seznamem nebo &. Verze přesto platí; přesuňte řádek na začátek. |
| ECID version already set in this file, ignored | Druhý ^ECID ve stejném souboru. Platí první. |
| Wrong ECID version format (^ECID:major.minor, e.g. ^ECID:1.5), ignored | Chybný zápis verze, např. ^ECID:1 nebo ^ECID:1.x. |
| Unknown directive (^ECID:major.minor expected), ignored | Řádek začíná ^, ale nejde o ^ECID:. |
| Description (?) without preceding signal (@ or :), ignored | Řádek ? není hned za signálem (např. za pravidlem > nebo na začátku souboru). |

# 12. Kompletní příklady

## 12.1 J1939 / FMS

```
; J1939 - priorita a SA se ignoruji
[Pedal
=0:released,=1:pressed,=2:error,=3:not available
]
--
%E,M:x3FFFF00
>I:x00FEF100,#CCVS Cruise Control/Vehicle Speed
@U:8:16:L:0.00390625:0,*km/h,#Wheel based vehicle speed
@H:28:2:L,$Pedal,#Brake switch
>I:x00F00400,#EEC1 Electronic Engine Controller 1
@U:16:8:L:1:125,*%,#Actual engine percent torque
@U:24:16:L:0.125:0,*rpm,#Engine speed
>I:x00FEEE00,#ET1 Engine Temperature 1
@U:0:8:L:1:40,*°C,#Engine coolant temperature
--
```

## 12.2 OBD-II přes ISO-TP

```
--
%I:x7E8,M:x7F8,#OBD Resp
{
%m0:xF0,0:x00,#SF
%S:0:4
{
%1:x41,#Mode01
>2:x0C,#Engine speed
@U:32:16:B:0.25:0,*rpm,#Engine speed
>2:x05,#Coolant temperature
@U:24:8:L:1:40,*°C,#Coolant temperature
>2:*,#other PID
}
}
>m0:xF0,0:x10,S:8:12:B,#FF
>m0:xF0,0:x20,#CF
--
```

| Zpráva 0x7E8 | Výsledek |
|---|---|
| `04 41 0C 1A F8` | `OBD Resp SF Mode01 Engine speed size:4`, Engine speed = 1726 rpm |
| `03 41 05 7B` | `… Coolant temperature size:3`, 83 °C |
| `10 14 49 02 …` | `OBD Resp FF size:20` |
| `21 …` | `OBD Resp CF` |

## 12.3 Připravené knihovny

| Soubor | Obsah |
|---|---|
| UDS_ISO14229.ecid | Hlavní soubor UDS (ISO 14229) + kompletní OBD-II (~180 PID, Mode 01–0A, DID F4xx). 11bit i 29bit ID podle ISO 15765-4. Vkládá UDS_req/resp, UDS_DID_SF/FF a OBD2_req/resp. |
| KWP2000_CAN.ecid | Hlavní soubor KWP 2000 (ISO 14230-3) over CAN + OBD-II. Vkládá KWP_req/resp a OBD2_req/resp. |
| FMS.ecid | FMS-Standard / J1939: CCVS, EEC1/2, LFC, HRLFC, DD, ET1, HOURS, VDHR, SERV, TCO1, AMB, DC1/2, DM1 (DTC 1–10), VI, DI, CI, ECUID (BAM) a další. |
| OBD2_J1979.ecid | Samostatný hlavní soubor jen OBD-II (SAE J1979) bez UDS: 11bit i 29bit ID podle ISO 15765-4, jednotlivé zprávy SF/FF/CF/FC i složené ISO-TP se signály všech PID, negativní odpověď 7F. Vkládá OBD2_frames_req/resp a OBD2_req/resp(_TP). |
| CANopen_CiA301.ecid | CANopen CiA 301 (předdefinovaná ID): NMT, SYNC, TIME, EMCY (chybové kódy, chybový registr), SDO (příkaz, index/subindex, data, abort kódy), heartbeat/boot-up, LSS a TPDO/RPDO 1–4. Názvy s číslem uzlu 1–127, signály jsou společné pro všechny uzly (maska ID) – odpovídá-li více uzlů, hodnoty se střídají. PDO jen názvy (obsah podle EDS). Vkládá CANopen_SDO_*.ecid. |
| CHAdeMO.ecid | CHAdeMO (11bit, 500 kbit/s): vozidlo 100h–102h, nabíječka 108h–109h (napětí, proudy, SOC, časy, stavové a chybové bity), V2H 200h, 208h, 209h. Jen samostatná sběrnice konektoru. |
| GBT27930.ecid | GB/T 27930-2015 (čínské DC nabíjení, J1939 29bit): CHM, BHM, CRM, BRM, BCP, CTS, CML, BRO, CRO, BCL, BCS, CCS, BSM, BMV (256 článků), BMT, BST, CST, BSD, CSD, BEM, CEM, TP.CM/TP.DT. Víceramcové zprávy přes T:J1939. |
| XCP_CAN.ecid<br>CCP.ecid | XCP on CAN a CCP 2.1: názvy příkazů, odpovědí, chyb a událostí, parametry hlavních příkazů (MTA, DAQ). ID nejsou standardní – vzorová ID v souboru upravte podle A2L. Odpovědi a DAQ jen názvy (obsah závisí na příkazu / A2L). Vkládá XCP_cmd/XCP_res. |
| UDS_VW.ecid | UDS + OBD-II + koncern VW: identifikační DID, motor Simos18, MEB (ID.3/ID.4/Enyaq) – baterie, články, teploty, DC-DC, klimatizace. |
| UDS_Ford_MachE.ecid<br>UDS_GM_Bolt.ecid<br>UDS_Jaguar_IPace.ecid<br>UDS_MG_ZSEV.ecid<br>UDS_Renault_Zoe.ecid | UDS + OBD-II + výrobní DID elektromobilů dané značky (SOC, napětí, proud, teploty baterie…). Pro každou značku samostatný soubor, protože značky používají stejná ID s jinými DID. |
| priklady/ | Komentované příklady k tomuto manuálu: 01 základ, 02 šablony, 03 signály a speciální hodnoty, 04 ISO-TP pro UDS a OBD-II, 05 J1939 TP. |

| Soubor | Obsah |
|---|---|
| X_TP.ecid | Ke každému tělu (UDS_req/resp, KWP_req/resp, OBD2_req/resp, UDS_DID, UDS_OBD_F4) patří verze _TP se složenými zprávami ISO-TP: názvy a všechny signály – i delší odpovědi (VIN, OBFCM, DTC 1–10, všechny OBD PID). Soubory bez _TP pojmenují jednotlivé CAN zprávy SF/FF/CF/FC. |

# 13. Časté chyby a doporučení

- **Obecné pravidlo před konkrétním** – konkrétní pravidlo pak nikdy nevyhraje. Řešení: `*` a „ostatní“ vždy na konec bloku.
- `100` **místo** `x100` – bez předpony je to desítkové 100 (0x64).
- **Offset se znaménkem jako v DBC** – v ECID se offset odečítá.
- **Motorola start bit jako v DBC (MSB)** – v ECID je to LSB.
- **Čárka v názvu signálu nebo v textu seznamu** – čárka položky odděluje. V názvu pravidla `#` (poslední položka) čárka smí být.
- **Seznam hodnot až za signálem** – seznam musí být definován dřív. Seznamy dávejte na začátek hlavního souboru.
- `--` **ve vloženém souboru** – smaže šablonu (ID) volajícího souboru.
- **Podmínka na data bez masky** – `m0:xF0` samotné nic nekontroluje; přidejte i hodnotu `0:x20`.
- **Signály ve First/Consecutive Frame** – data se neskládají, dekódování dává smysl jen v Single Frame.
- **Rozsah bajtu u pravidla se signály** – EYE bajt pak nekontroluje. Pro signál použijte přesnou hodnotu nebo masku.

# 14. Převod DBC do ECID (dbc2ecid)

dbc2ecid.exe je program pro příkazovou řádku Windows, který převede soubor Vector DBC do ECID 1.5. Nepotřebuje instalaci ani další knihovny. Výsledek je soubor ECID (cp1250, CRLF, ^ECID:1.5) a report, který vypíše, co se nepřevedlo.

## 14.1 Použití

```
dbc2ecid vstup.dbc [vystup.ecid] [/j1939] [/prefix] [/en] [/report:soubor.txt]

dbc2ecid motor.dbc
dbc2ecid C:\DBC\truck.dbc C:\ECID\truck.ecid /j1939 /report:truck.txt
```

- vystup.ecid – bez uvedení se použije název vstupu s příponou .ecid.
- /j1939 – všechny 29bit zprávy se převedou jako J1939 (maska PGN). Bez přepínače jen zprávy s VFrameFormat = J1939PG nebo při ProtocolType = J1939.
- /prefix – název signálu bude Zpráva.Signál (jinak jen Signál). Hodí se, když se názvy signálů v různých zprávách opakují.
- /en – komentáře v ECID a report anglicky.
- /report:soubor – report se kromě obrazovky uloží i do souboru.
- Návratový kód: 0 = OK, 1 = chybí parametry, 2 = vstup nejde otevřít, 4 = výstup nejde zapsat.

## 14.2 Co se převede

- Zprávy BO_ → pravidlo >I:x…,[E],#název. Komentář zprávy (CM_ BO_) → komentář ; nad pravidlem. Dlouhý název (SystemMessageLongSymbol) má přednost.
- J1939 → maska PGN: PDU2 (PF ≥ F0h) M:x3FFFF00, PDU1 M:x3FF0000 (bez priority, SA i DA). Zpráva J1939 delší než 8 B dostane T:J1939 (skládání TP v Data view).

- Signály SG_ → @U / @S: start bit Motoroly se přepočítá z MSB (DBC) na LSB (ECID), offset se otočí (DBC –40 → ECID 40). SIG_VALTYPE_ float/double → @F:…:32 / 64. Signál delší než 64 bitů zarovnaný na bajty (Intel) → text @T (např. VIN).
- Tabulky hodnot VAL_ → s□z□amy [VT_Zp□ava_Sig□al]; st□j□é tab□lky s□zapíší j□□j□d□o□.
- Komentář signálu (CM_ SG_) → řádky ?popis (víceřádkový komentář = více řádků ?).
- Multiplex (M / mN) → pravidlo s podmínkou na bity multiplexoru (hodnota a maska, libovolné bity v DB0–DB7) a název „Zpráva [Mux=N]“. Rozšířený multiplex (SG_MUL_VAL_, vnořené multiplexory) → podmínka na všechny multiplexory v cestě, rozsahy se rozepíšou (max. 64 hodnot). Signály bez multiplexu jsou u obecného pravidla zprávy, které je až za pravidly s podmínkou.
- Kódování: vstup v UTF-8 se převede do cp1250. Čárka v názvu signálu, jednotce nebo textu tabulky se nahradí středníkem (čárka v ECID odděluje položky) – je uvedeno v reportu.

## 14.3 Co se nepřevede (report)

- min/max signálu, uzly (BU_, vysílač/příjemci), atributy (GenMsgCycleTime, GenSigStartValue, …), skupiny signálů (SIG_GROUP_), proměnné prostředí (EV_), samostatné VAL_TABLE_.
- Multiplexor za DB7 (podmínky ECID jsou jen na DB0–DB7), multiplex s více než 64 hodnotami a signál delší než 64 bitů, který není zarovnaný na bajty – signál se převede bez podmínky nebo vynechá a je uveden v reportu.
- Signály bez zprávy (VECTOR__INDEPENDENT_SIG_MSG) se vynechají.

## 14.4 Příklad výstupu

```
; ---------- Mux (11 bit x200, DLC 8) ----------
>I:x200,0:x01,#Mux [Sel=1]
@U:8:16:L:0.1:0,*V,#A
?Voltage when Sel=1
>I:x200,0:x02,#Mux [Sel=2]
@S:8:16:L:1:0,*A,#B
>I:x200,#Mux
@U:0:8:L:1:0,$VT_Mux_Sel,#Sel
```

## 14.5 Doporučený postup a překlad

- Převeďte DBC, přečtěte report a otevřete ECID v programu. Chyby načtení jsou v <soubor>.ecid.log (kapitola 11).
- Vygenerovaný soubor je běžný ECID – lze ho dál upravovat (šablony, vlastní názvy, seznamy, ?popisy) nebo vložit (&) do vlastního hlavního souboru.
- Převod je jednosměrný; po změně DBC soubor vygenerujte znovu (vlastní úpravy si držte v samostatném souboru, který vygenerovaný soubor vkládá).
- Přeložení ze zdrojového kódu: Visual Studio (Developer Command Prompt) cl /EHsc /O2 /std:c++17 dbc2ecid.cpp, nebo MinGW g++ -O2 -std=c++17 -static -o dbc2ecid.exe dbc2ecid.cpp.

# Příloha A – rychlá reference

```
;  komentar
>  pravidlo     >I:x123,E,M:x1FFFFF00,0:x01,m1:xF0,1:x20,S:0:4,#Nazev
%  sablona      %I:x7E8,M:x7F8,#Prefix
+  push+sablona +1:x22,#RDBI
{  push         }  pop          -  pop          --  reset vseho
!  reset        !N !S !L !I !M !m !0..!7
=  nazev sab.   =Pevny nazev
[  seznam       [Jmeno  /  =0:Off,=1:On,=x80..xFF:Rozsah  /  ]
@  signal       @U:bit:delka:L|B:vaha:offset,*jednotka,$seznam,#Nazev
:  signal sab.  :H:16:8:L,#Nazev
!  spec. hodn.  @U:0:8:L:1:0,!xFF:N/A,!xF0..xFE:error,!J
&  vlozeni      &soubor.ecid
^  verze        ^ECID:1.5
?  popis        ?Delsi popis predchoziho signalu

Polozky:  I:id  I:*  R:od:do  L:a:b:c  M:maska  M:*  E
          N:hod  N:od:do  N:*  mN:maska  mN:*  S:bit:delka[:B]  #nazev
          T:ISOTP  T:ISOTPX  T:J1939  T:*
Typy:     U S H F B T J P O D   (T:*:n = pole textu oddeleneho *)      hodnota = raw * vaha -
offset
Bity:     bit = bajt*8 + bit_v_bajtu, vzdy LSB signalu
```