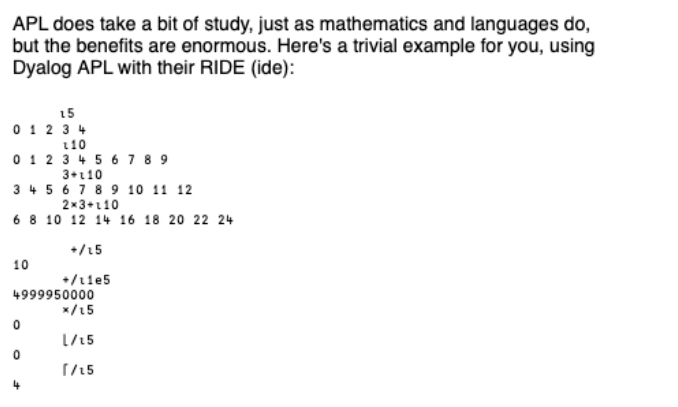
PLANELY WINNING THE APL FORGE

Holden Hoover
holdenhoover@bkaw.ca

WHO AM I?

- High School Student
- Using APL since 2020
- Summer of 2021, worked on APEX



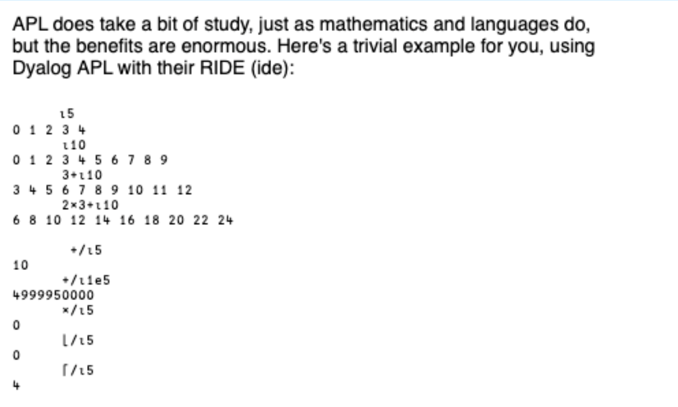
APL does take a bit of study, just as mathematics and languages do, but the benefits are enormous. Here's a trivial example for you, using Dyalog APL with their RIDE (ide):

```
      ⍳5
0 1 2 3 4
      ⍳10
0 1 2 3 4 5 6 7 8 9
      3+⍳10
3 4 5 6 7 8 9 10 11 12
      2×3+⍳10
6 8 10 12 14 16 18 20 22 24

       +/⍳5
10
      +/⍳1e5
4999950000
      ×/⍳5
0
      ⌊/⍳5
0
      ⌈/⍳5
4
```

WHO AM I?

- High School Student
- Using APL since 2020
- Summer of 2021, worked on APEX
- Aviation Enthusiast



APL does take a bit of study, just as mathematics and languages do, but the benefits are enormous. Here's a trivial example for you, using Dyalog APL with their RIDE (ide):

```
      ⍳5
0 1 2 3 4
      ⍳10
0 1 2 3 4 5 6 7 8 9
      3+⍳10
3 4 5 6 7 8 9 10 11 12
      2×3+⍳10
6 8 10 12 14 16 18 20 22 24

       +/⍳5
10
      +/⍳1e5
4999950000
      ×/⍳5
0
      ⌊/⍳5
0
      ⌈/⍳5
4
```

MY PROJECT

Radar Ingest System is an application that aggregates, processes, and stores ADS-B messages from a network of individual and independent feeder antennas, creating a unified database of real-time data from multiple aircraft across the network.

ADS-B?

WHAT IS ADS-B?

Automatic **D**ependent **S**urveillance-**B**roadcast

WHAT IS ADS-B?

Automatic - No request is made

Dependent - Systems are on the plane

Surveillance - For surveillance of planes

Broadcast - Available to all

HOW DOES ADS-B WORK?

In a nutshell

Plane has a computer

Computer broadcasts a message

Message is received by an antenna

HOW DOES RADAR INGEST SYSTEM WORK?

Radar (antenna) connects to RIS

Radar receives message

Radar sends the message to RIS

RIS **ingests** the message

RIS appends (or modifies) a plane entry into the database

WHAT EXACTLY IS “INGESTION”

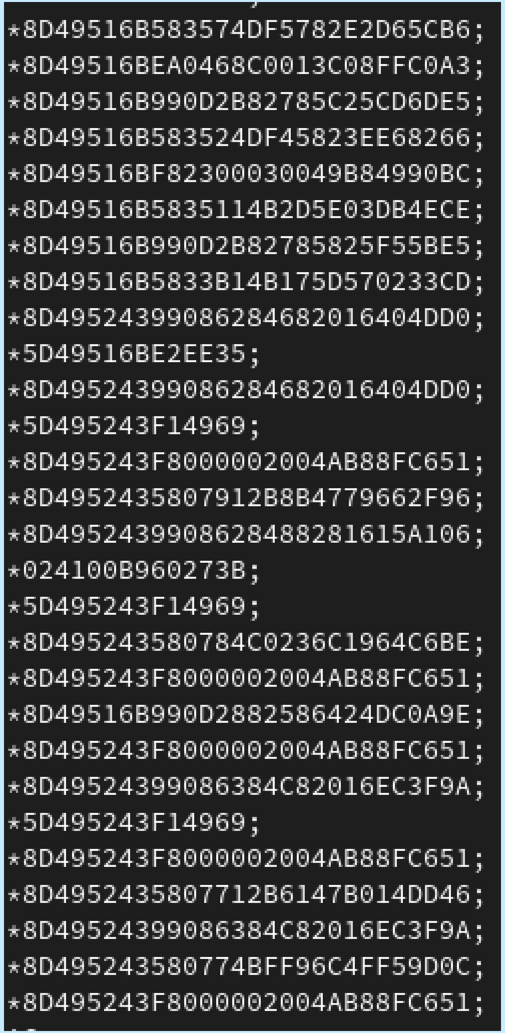
Message gets sent to RIS from Radar

The message gets cleaned up

The message is sent to adsbProcess

The aircraft record is returned to the RIS instance

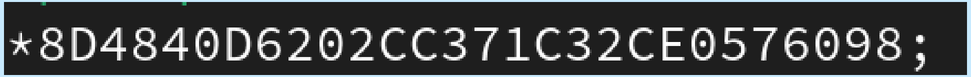
The record is added to the database



```
*8D49516B583574DF5782E2D65CB6;
*8D49516BEA0468C0013C08FFC0A3;
*8D49516B990D2B82785C25CD6DE5;
*8D49516B583524DF45823EE68266;
*8D49516BF82300030049B84990BC;
*8D49516B5835114B2D5E03DB4ECE;
*8D49516B990D2B82785825F55BE5;
*8D49516B5833B14B175D570233CD;
*8D495243990862846820164O4DD0;
*5D49516BE2EE35;
*8D495243990862846820164O4DD0;
*5D495243F14969;
*8D495243F8000002004AB88FC651;
*8D4952435807912B8B4779662F96;
*8D49524399086284882816154A106;
*024100B960273B;
*5D495243F14969;
*8D495243580784C0236C1964C6BE;
*8D495243F8000002004AB88FC651;
*8D49516B990D2882586424DC0A9E;
*8D495243F8000002004AB88FC651;
*8D49524399086384C82016EC3F9A;
*5D495243F14969;
*8D495243F8000002004AB88FC651;
*8D4952435807712B6147B014DD46;
*8D49524399086384C82016EC3F9A;
*8D495243580774BFF96C4FF59D0C;
*8D495243F8000002004AB88FC651;
```

WHAT EXACTLY IS “INGESTION”

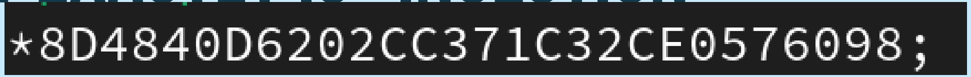
CONTINUED



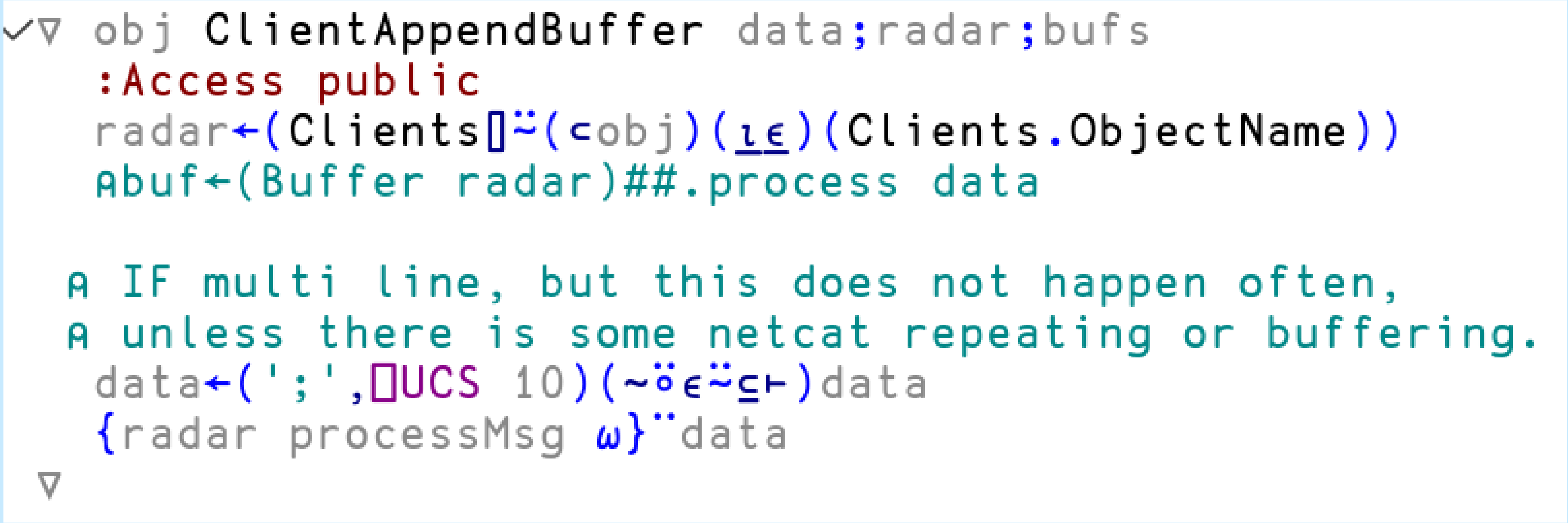
```
*8D4840D6202CC371C32CE0576098;
```

WHAT EXACTLY IS "INGESTION"

CONTINUED



```
*8D4840D6202CC371C32CE0576098;
```



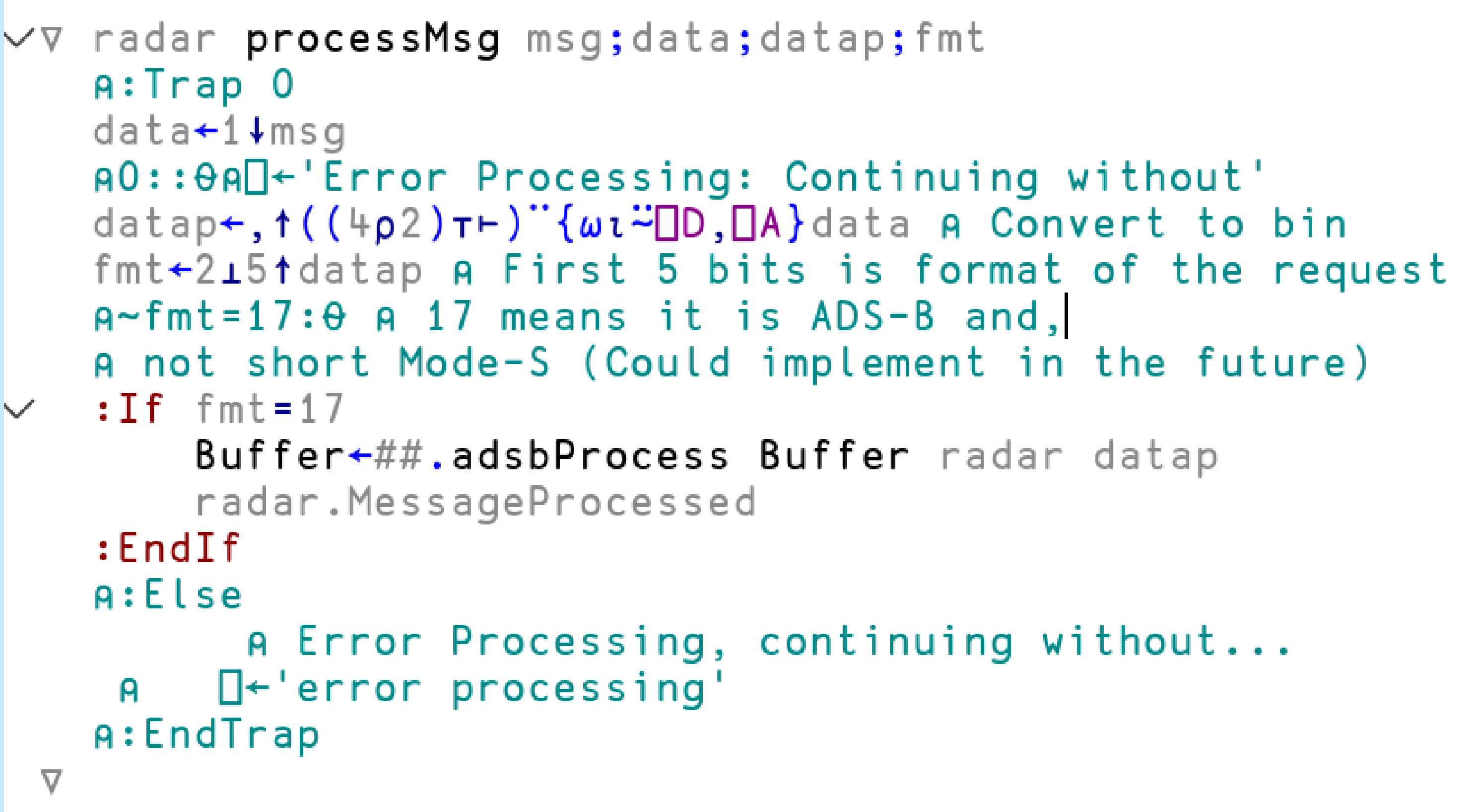
```
✓∇ obj ClientAppendBuffer data;radar;bufs
   :Access public
   radar←(Clients⌷⍨(⊂obj)(⍳∊)(Clients.ObjectName))
   ⍝buf←(Buffer radar)##.process data

  ⍝ IF multi line, but this does not happen often,
  ⍝ unless there is some netcat repeating or buffering.
   data←(';',⎕UCS 10)(~⍤∊⍨⊆⊢)data
   {radar processMsg ⍵}¨data
 ∇
```

WHAT EXACTLY IS “INGESTION”

CONTINUED

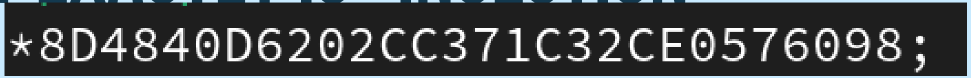
```
*8D4840D6202CC371C32CE0576098;
```



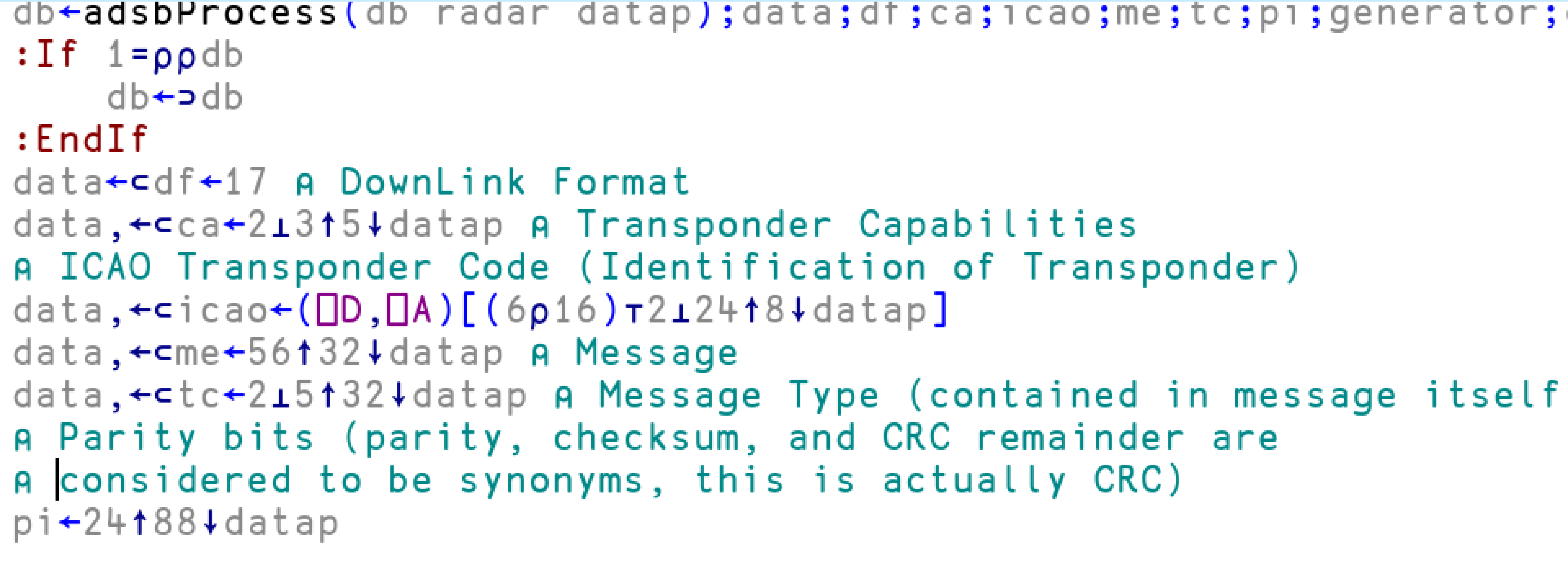
```
∨∇ radar processMsg msg;data;datap;fmt
     ⍝:Trap 0
     data←1↓msg
     ⍝0::θ⍝⎕←'Error Processing: Continuing without'
     datap←,↑((4⍴2)⊤⊢)¨{⍵⍳⍨⎕D,⎕A}data ⍝ Convert to bin
     fmt←2⊥5↑datap ⍝ First 5 bits is format of the request
     ⍝~fmt=17:θ ⍝ 17 means it is ADS-B and,
     ⍝ not short Mode-S (Could implement in the future)
∨    :If fmt=17
         Buffer←##.adsbProcess Buffer radar datap
         radar.MessageProcessed
     :EndIf
     ⍝:Else
          ⍝ Error Processing, continuing without...
      ⍝    ⎕←'error processing'
     ⍝:EndTrap
 ∇
```

WHAT EXACTLY IS “INGESTION”

CONTINUED



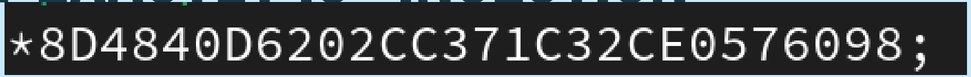
```
*8D4840D6202CC371C32CE0576098;
```



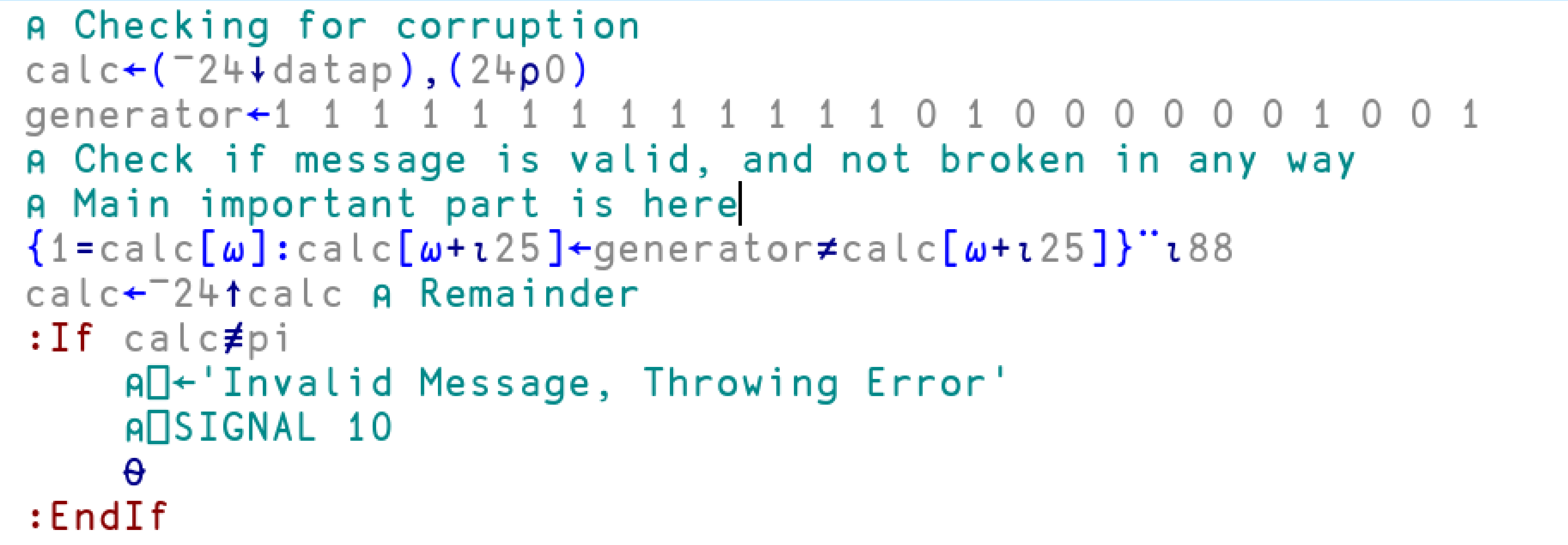
```
db←adsbProcess(db radar datap);data;df;ca;icao;me;tc;pi;generator;
:If 1=⍴⍴db
    db←⊃db
:EndIf
data←⊂df←17 ⍝ DownLink Format
data,←⊂ca←2⊥3↑5↓datap ⍝ Transponder Capabilities
⍝ ICAO Transponder Code (Identification of Transponder)
data,←⊂icao←(⎕D,⎕A)[(6⍴16)⊤2⊥24↑8↓datap]
data,←⊂me←56↑32↓datap ⍝ Message
data,←⊂tc←2⊥5↑32↓datap ⍝ Message Type (contained in message itself
⍝ Parity bits (parity, checksum, and CRC remainder are
⍝ considered to be synonyms, this is actually CRC)
pi←24↑88↓datap
```

WHAT EXACTLY IS “INGESTION”

CONTINUED



```
*8D4840D6202CC371C32CE0576098;
```

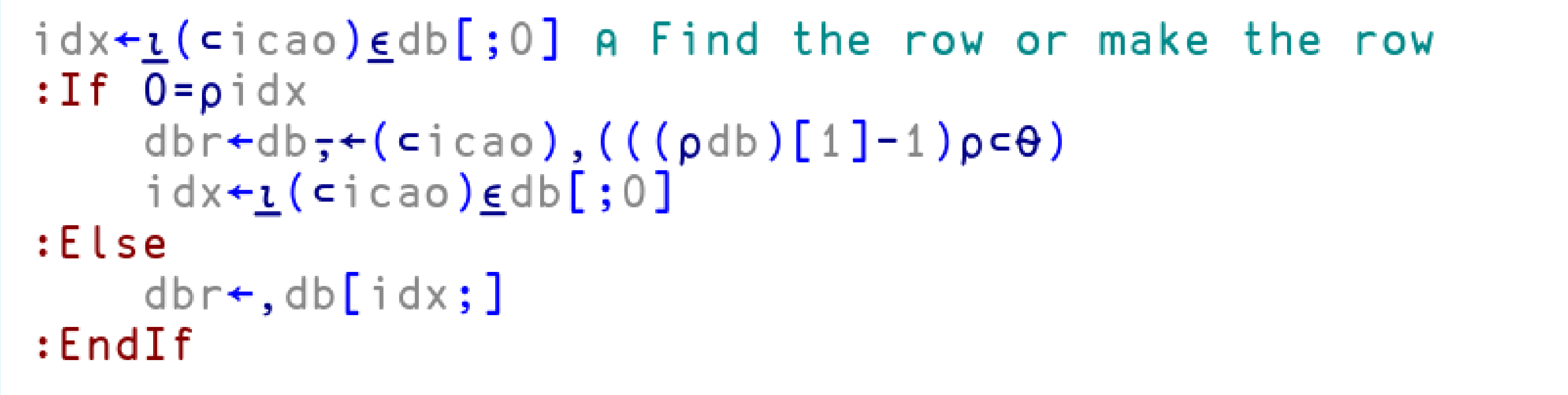


```
⍝ Checking for corruption
calc←(¯24↓datap),(24⍴0)
generator←1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 1 0 0 1
⍝ Check if message is valid, and not broken in any way
⍝ Main important part is here
{1=calc[⍵]:calc[⍵+⍳25]←generator≠calc[⍵+⍳25]}¨⍳88
calc←¯24↑calc ⍝ Remainder
:If calc≢pi
    ⍝⎕←'Invalid Message, Throwing Error'
    ⍝⎕SIGNAL 10
    ⍬
:EndIf
```

WHAT EXACTLY IS "INGESTION"

CONTINUED

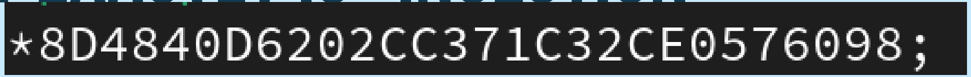
```
*8D4840D6202CC371C32CE0576098;
```



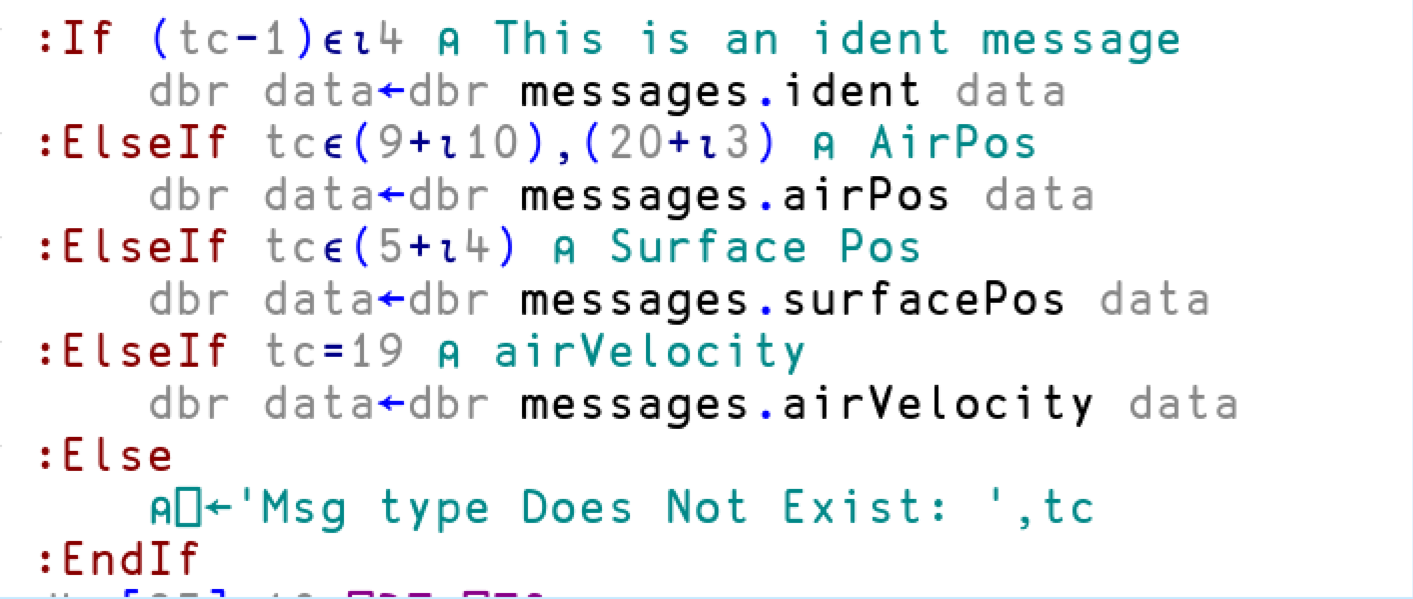
```
idx←⍸(⊂icao)⍷db[;0] ⍝ Find the row or make the row
:If 0=⍴idx
    dbr←db⍪←(⊂icao),(((⍴db)[1]-1)⍴⊂⍬)
    idx←⍸(⊂icao)⍷db[;0]
:Else
    dbr←,db[idx;]
:EndIf
```

WHAT EXACTLY IS “INGESTION”

CONTINUED



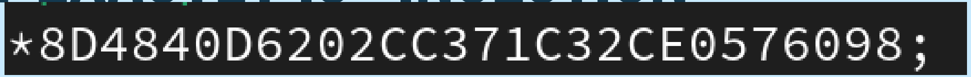
```
*8D4840D6202CC371C32CE0576098;
```



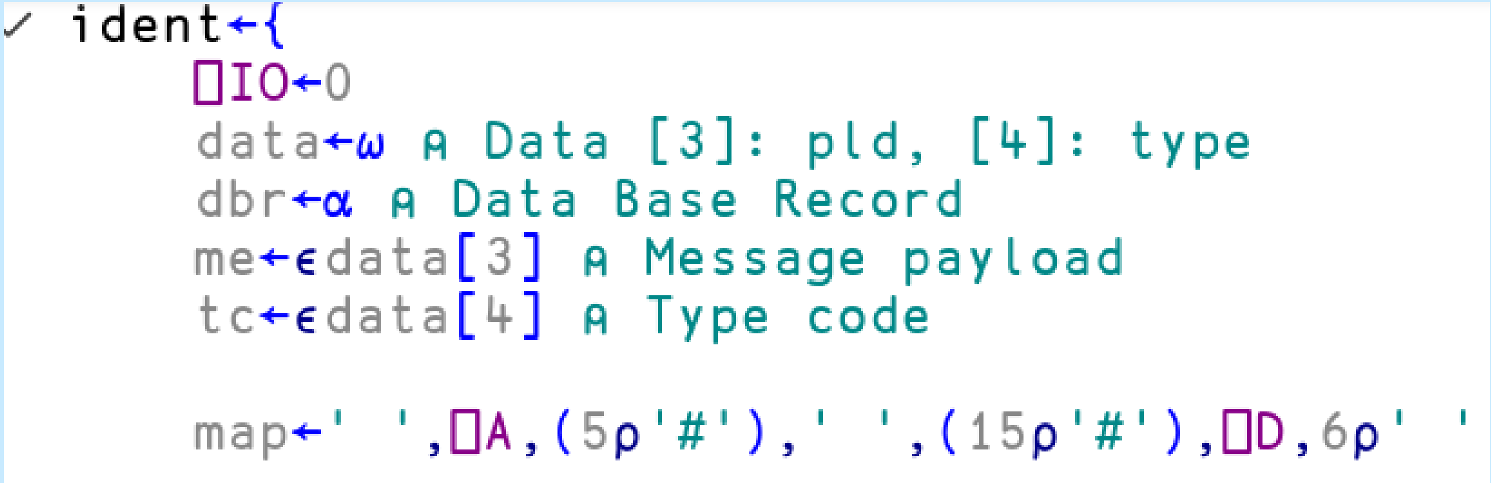
```
:If (tc-1)∊⍳4 ⍝ This is an ident message
    dbr data←dbr messages.ident data
:ElseIf tc∊(9+⍳10),(20+⍳3) ⍝ AirPos
    dbr data←dbr messages.airPos data
:ElseIf tc∊(5+⍳4) ⍝ Surface Pos
    dbr data←dbr messages.surfacePos data
:ElseIf tc=19 ⍝ airVelocity
    dbr data←dbr messages.airVelocity data
:Else
    ⍝⎕←'Msg type Does Not Exist: ',tc
:EndIf
```

WHAT EXACTLY IS "INGESTION"

CONTINUED

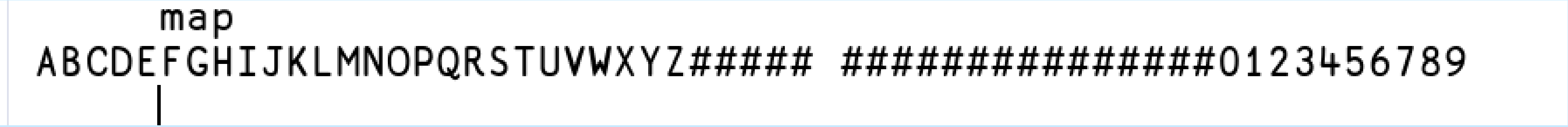


```
*8D4840D6202CC371C32CE0576098;
```



```
✓ ident←{
        ⎕IO←0
        data←⍵ ⍝ Data [3]: pld, [4]: type
        dbr←⍺ ⍝ Data Base Record
        me←∊data[3] ⍝ Message payload
        tc←∊data[4] ⍝ Type code

        map←' ',⎕A,(5⍴'#'),' ',(15⍴'#'),⎕D,6⍴' '
```

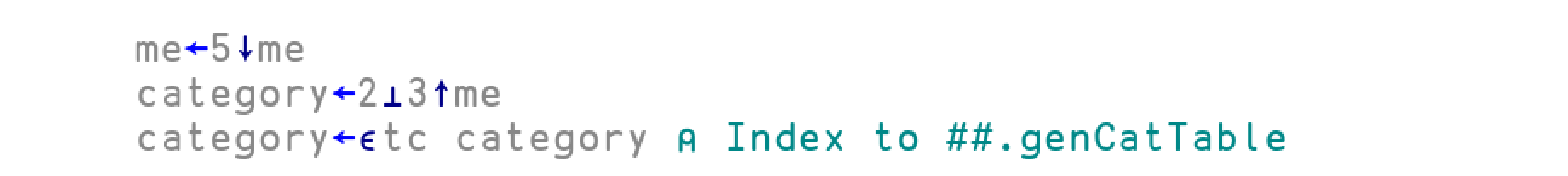


```
      map
ABCDEFGHIJKLMNOPQRSTUVWXYZ##### ###############0123456789
```

WHAT EXACTLY IS “INGESTION”

CONTINUED

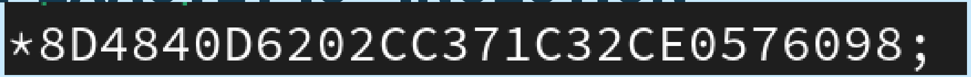
*8D4840D6202CC371C32CE0576098;



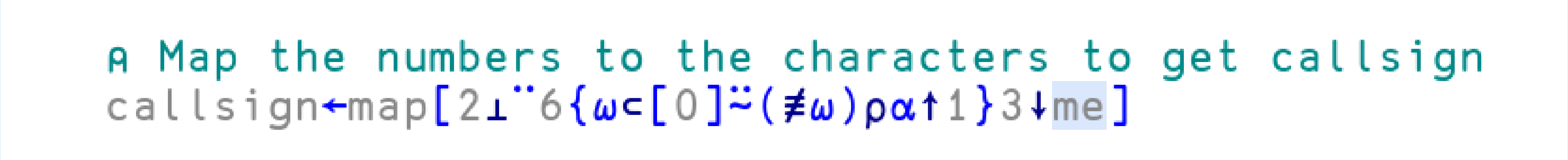
```
me←5↓me
category←2⊥3↑me
category←∊tc category ⍝ Index to ##.genCatTable
```

WHAT EXACTLY IS “INGESTION”

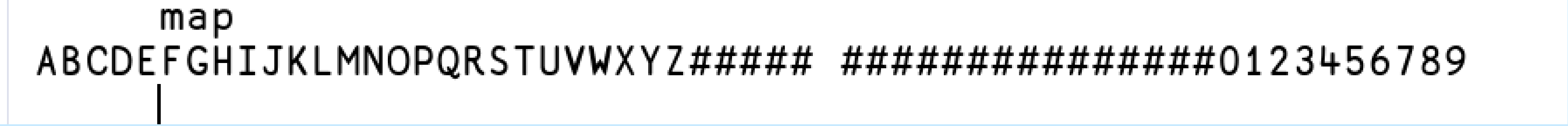
CONTINUED



```
*8D4840D6202CC371C32CE0576098;
```



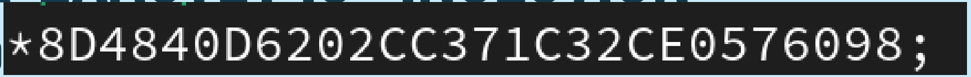
```
⍝ Map the numbers to the characters to get callsign
callsign←map[2⊥¨6{ω⊂[0]⍨(≢ω)ρα↑1}3↓me]
```



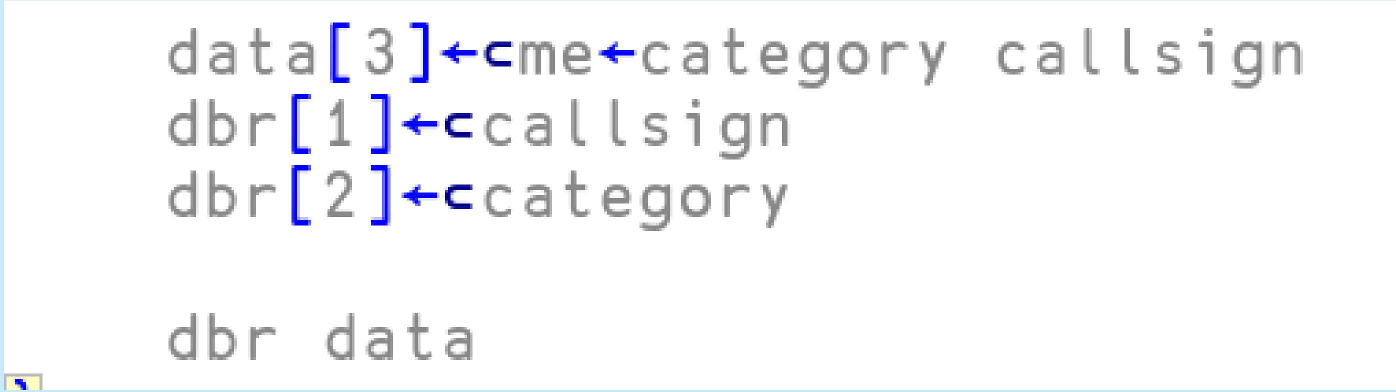
```
      map
ABCDEFGHIJKLMNOPQRSTUVWXYZ##### ###############0123456789
      
```

WHAT EXACTLY IS “INGESTION”

CONTINUED



```
*8D4840D6202CC371C32CE0576098;
```



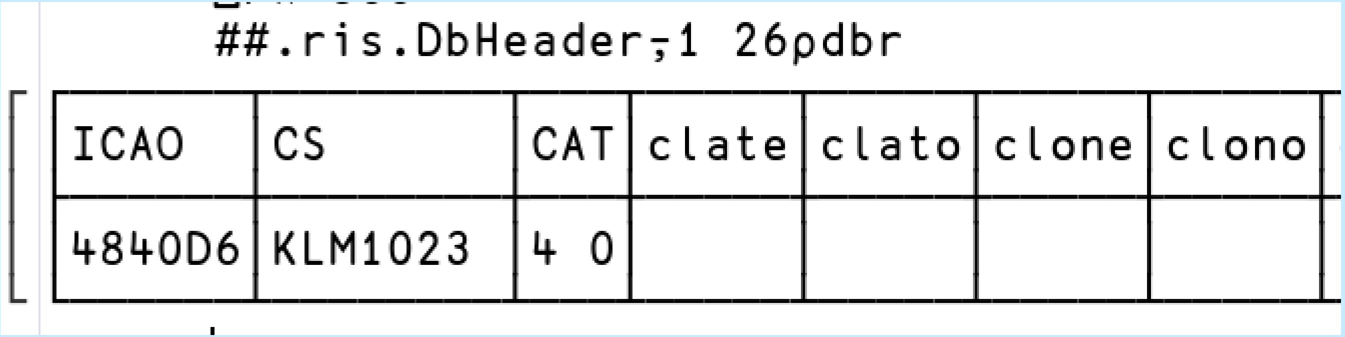
```
    data[3]←⊂me←category callsign
    dbr[1]←⊂callsign
    dbr[2]←⊂category

    dbr data
```

WHAT EXACTLY IS “INGESTION”

CONTINUED

```
*8D4840D6202CC371C32CE0576098;
```



```
      ##.ris.DbHeader⍪1 26⍴dbr
```

ICAO	CS	CAT	clate	clato	clone	clono
4840D6	KLM1023	4 0				

RIS ARCHITECTURE

Cider

Tatin

Dyalog Class

Dyalog Conga

Dyalog Jarvis

FUTURE

Proper **U**I

Needs some major optimisation

Needs to be made more robust

New radar communication protocol?

THANK YOU