# Error Handling

## Adám Brudzewsky

# Error Handling

## Adám Brudzewsky

# Error Handling – Part 1

*Number for given message*
aplcart.info?q=:: message

Tradfn structure `:Trap` errnos (0 means all)

Dfn error guard errnos`::`code

Error Number of last error `⎕EN`

Error Message for number `⎕EM`

# Error Handling – Part 2

*Construct error message*
aplcart.info?q=dmx 1

| | |
|---|---|
| Execute there on error | `⎕TRAP←errnos 'E' code` |
| Cut back and execute | `⎕TRAP←errnos 'C' code` |
| Ext'd Diagnostic Message | `⎕DMX` |

# Error Handling – Part 3

Signal an error: `⎕SIGNAL errno`

Signal a custom error: `⎕SIGNAL⊂name-value pairs`

Assert: aplcart.info?q=assert

Resignal: aplcart.info?q=resignal

# Error Handling – Part 4

Debugging mode: `debug←0`

Adjustable trapping: `∊debug↓0(1 22)`

Collecting info: `error←⎕DMX`

Be mindful of dfn error scopes!

# Error Handling – Part 4

dyalog.tv

Debugging mode: `debug←0`

Adjustable trapping: `∊debug↓0(1 22)`

Collecting info: `error←⎕DMX`

Be mindful of dfn error scopes!

# Error Handling – Part 4

dyalog.tv

Debugging mode: `debug←0`

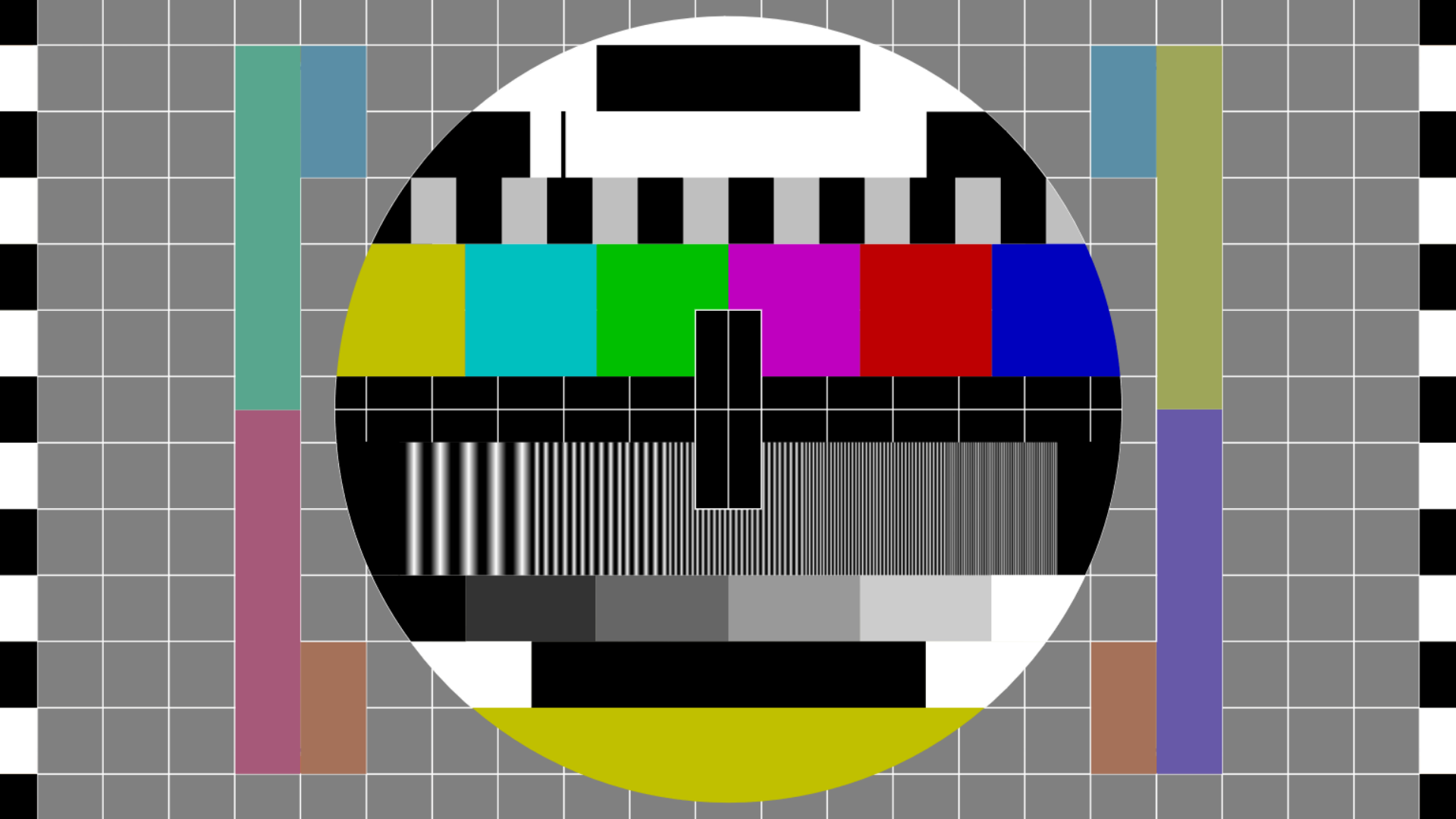
Adjustable trapping: `∈debug↓0(1 22)`

Collecting info: `error←⎕DMX`

Be mindful of dfn error scopes!

# Capsules

## what ⎕SIGNAL leaves



# Tradfn Capsules

## what ⎕SIGNAL leaves

# Capsules

```
:Namespace test
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

# Capsules

```
:Namespace test
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
test.inner
```

# Capsules

```
:Namespace test
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
test.inner
```

# Capsules

```
:Namespace test
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
ERROR 256
      test.inner
      ^
```

# Capsules

```
:Namespace test

    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
ERROR 256
      test.inner
      ^
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
ERROR 256
      test.inner
      ^
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
ERROR 256
      test.inner
      ^
      test.outer
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
ERROR 256
      test.inner
      ^
      test.outer
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
ERROR 256
      test.inner
      ^
      test.outer
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
ERROR 256
      test.inner
      ^
      test.outer
trapped outside!
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          ⎕SIGNAL 256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
test.inner
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
test.inner
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
trapped inside!
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
trapped inside!
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
trapped inside!
      test.outer
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
trapped inside!
      test.outer
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
trapped inside!
      test.outer
```

# Capsules

```
:Namespace test
    ∇ outer
      :Trap 0
          inner
      :Else
          ⎕←'trapped outside!'
      :EndTrap
    ∇
    ∇ inner
      :Trap 0
          {⎕SIGNAL ⍵}256
      :Else
          ⎕←'trapped inside!'
      :EndTrap
    ∇
:EndNamespace
```

```
      test.inner
trapped inside!
      test.outer
trapped inside!
```

# Overview

# Overview

- Special events: errors, interrupts, exceptions

# Overview

- Special events: errors, interrupts, exceptions
- Resetting system constants

# Overview

- Special events: errors, interrupts, exceptions
- Resetting system constants
- Poll: event message to number

# Special Events

## exceptions and interrupts

# Special events

## Errors

⎕EN is in range 1...999

Catch all with :Trap 0

# Special events

## Errors

⎕EN is in range 1...999

Catch all with `:Trap 0`

## Interrupts

⎕EN is in range 1001...1008

Catch all with `:Trap 1000`

# Special events

## Errors

⎕EN is in range 1...999

Catch all with `:Trap 0`

## Exceptions

⎕EN is 90

## Interrupts

⎕EN is in range 1001...1008

Catch all with `:Trap 1000`

# Special events: interrupts

## Interrupts

⎕EN is in range 1001...1008

Catch all with :Trap 1000

# Special events: interrupts

## Interrupts

aplcart.info

⎕EN is in range 1001...1008

Catch all with :Trap 1000

⍸ Tell me about:

X,Y,Z:any M,N:num I,J:int A,B:Bool C,D:char f,g,h:fn ax:axis s:scal v:vec m:mat

| | |
|---|---|
| ⍬ | Empty Numeric Vector |
| ⊢Y | Same: Y |
| X dop Y⊢Z | Separate dyadic operator's right operand from its right argument (same as (X dop Y)Z ) |
| X⊢Y | Right: Y |
| X⊢Y | Church Boolean false (X if false, else Y) |
| ⊣Y | Same: Y |
| X⊣Y | Left: X |
| X⊣Y | Church Boolean true (X if true, else Y) |
| +Y | Conjugate ('Identity' if Y not complex) |
| +N | Mirror complex N across x-axis |
| M+N | Adding N to M |

Tell me about: :: interrupt

X,Y,Z:any M,N:num I,J:int A,B:Bool C,D:char f,g,h:fn ax:axis s:scal v:vec m:mat

| | |
|---|---|
| 1000::any exception | Covers all exceptions (errors 1001-1009) |
| 1001::STOP VECTOR | Execution stopped due to ⎕STOP or stop bit set from editor |
| 1002::WEAK INTERRUPT | Execution stopped by weak interrupt |
| 1003::INTERRUPT | Execution stopped by strong interrupt |
| 1004::INPUT INTERRUPT | EOF to ⍞ or soft interrupt to ⍞ |
| 1005::EOF INTERRUPT | EOF when reading input from a file (when an input to APL is from a file) |
| 1006::TIMEOUT | Time limit specified by ⎕RTL exceeded while awaiting input through ⍞ or ⎕SR |
| 1007::RESIZE | User resizes the ⎕SM window |
| 1008::DEADLOCK | Two threads acquiring a hold of two different tokens, and then each asks to hold the other token |

Showing 9 of 2589

Quiz

Tell me about: :: interrupt

X,Y,Z:any M,N:num I,J:int A,B:Bool C,D:char f,g,h:fn ax:axis s:scal v:vec m:mat

| | |
|---|---|
| 1000::any exception | Covers all exceptions (errors 1001-1009) |
| 1001::STOP VECTOR | Execution stopped due to ⎕STOP or stop bit set from editor |
| 1002::WEAK INTERRUPT | Execution stopped by weak interrupt |
| 1003::INTERRUPT | Execution stopped by strong interrupt |
| 1004::INPUT INTERRUPT | EOF to ⍞ or soft interrupt to ⍞ |
| 1005::EOF INTERRUPT | EOF when reading input from a file (when an input to APL is from a file) |
| 1006::TIMEOUT | Time limit specified by ⎕RTL exceeded while awaiting input through ⍞ or ⎕SR |
| 1007::RESIZE | User resizes the ⎕SM window |
| 1008::DEADLOCK | Two threads acquiring a hold of two different tokens, and then each asks to hold the other token |

Showing 9 of 2589

Quiz

# Special events: interrupts

```
F←{ }
```

# Special events: interrupts

```
F←{⎕DL 1}
```

# Special events: interrupts

```
F←{ ⊣ ⎕DL 1}
```

# Special events: interrupts

```
F←{1+⍵ ⊣ ⎕DL 1}
```

# Special events: interrupts

```
F←{⎕←1+⍵ ⊣ ⎕DL 1}
```

# Special events: interrupts

```
F←{∇ ⎕←1+⍵ ⊣ ⎕DL 1}
```

# Special events: interrupts

```
F←{::  ⋄  ∇ ⎕←1+⍵  ⊣  ⎕DL  1}
```

# Special events: interrupts

```
F←{1000:: ⋄ ∇ ⎕←1+⍵ ⊣ ⎕DL 1}
```

# Special events: interrupts

```
F←{1000::'got to',⍵ ⋄ ∇ ⎕←1+⍵ ⊣ ⎕DL 1}
```

# Special events: interrupts

```
F←{1000::'got to',⍵ ⋄ ∇ ⎕←1+⍵ ⊣ ⎕DL 1}
```

# Special events: interrupts

```
F←{1000::'got to',⍵ ⋄ ∇ ⎕←1+⍵ ⊣ ⎕DL 1}
F 0
```

# Special events: interrupts

```
F←{1000::'got to',⍵ ⋄ ∇ ⎕←1+⍵ ⊣ ⎕DL 1}
'I ',F 0
```

# Special events: interrupts

```
F←{1000::'got to',⍵ ⋄ ∇ ⎕←1+⍵ ⊣ ⎕DL 1}
'I ',F 0
```

# Special events: interrupts

```
F←{1000::'got to',⍵ ⋄ ∇ ⎕←1+⍵ ⊣ ⎕DL 1}
'I ',F 0
```

# Special events: interrupts

```
      F←{1000::'got to',⍵ ⋄ ∇ ⎕←1+⍵ ⊣ ⎕DL 1}
      'I ',F 0
1
2
3
4
5
6
I got to 6
```

Action ▸ Interrupt

# Special events: interrupts

```
Drill←{
      ⍞←'17×27?'
      ⍞≡⍎17×27
}
```

# Special events: interrupts

```
      Drill 0
```

```
Drill←{
      ⎕←'17×27?'
      ⍞≡⍎17×27
}
```

# Special events: interrupts

```
      Drill 0
17×27?
459
1
```

```
Drill←{
      ⎕←'17×27?'
      ⍞≡⍎17×27
}
```

# Special events: interrupts

```
      Drill 0
17×27?
459
1
```

```
Drill←{
      ⎕←'17×27?'
      ⎕RTL←⍵ ◇ ⍞≡⍎17×27
}
```

# Special events: interrupts

```
      Drill 0
17×27?
459
1
```

```
Drill←{
      ⎕←'17×27?'
      1006::'Too slow!'
      ⎕RTL←⍵ ◇ ⍞≡⍕17×27
}
```

# Special events: interrupts

```
      Drill 0
17×27?
459
1
```

```
Drill←{
    ⎕←'17×27?'
    1006::'Too slow!'
    ⎕RTL←⍵ ⋄ ⍞≡⍕17×27
}
```

# Special events: interrupts

```
      Drill 0
17×27?
459
1
      Drill 5
```

```
Drill←{
    ⎕←'17×27?'
    1006::'Too slow!'
    ⎕RTL←⍵ ◇ ⍞≡⍕17×27
}
```

# Special events: interrupts

```
      Drill 0
17×27?
459
1
      Drill 5
```

```
Drill←{
    ⎕←'17×27?'
    1006::'Too slow!'
    ⎕RTL←⍵ ⋄ ⍞≡⍕17×27
}
```

# Special events: interrupts

```
      Drill 0
17×27?
459
1
      Drill 5
```

```
Drill←{
    ⎕←'17×27?'
    1006::'Too slow!'
    ⎕RTL←⍵ ◇ ⍞≡⍕17×27
}
```

# Special events: interrupts

```
      Drill 0
17×27?
459
1
      Drill 5
17×23?
Too slow!
```

```
Drill←{
    ⎕←'17×27?'
    1006::'Too slow!'
    ⎕RTL←⍵ ◇ ⍞≡⍕17×27
}
```

# Special events

## Errors

⎕EN is in range 1...999

Catch all with `:Trap 0`

## Exceptions

⎕EN is 90

## Interrupts

⎕EN is in range 1001...1008

Catch all with `:Trap 1000`

# Special events: exceptions

**Exceptions**

⎕EN is 90

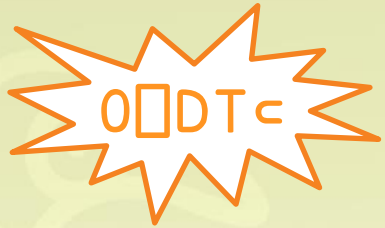
# Special events: exceptions

```
Check←{
}
```

# Special events: exceptions


```
Check←{
}
```

# Special events: exceptions




```
Check←{
      ¯1⎕DT⊂⍵
}
```

# Special events: exceptions

```
Check←{
      ¯1⎕DT⊂⍵
}
```

# Special events: exceptions

```
Check←{
      'Valid'⊣⍠1⎕DT⊂⍵
}
```

# Special events: exceptions

```
Check←{
      0::⎕DMX.Message
      'Valid'⊣¯1⎕DT⊂⍵
}
```

# Special events: exceptions

```
Check←{
      0::⎕DMX.Message
      'Valid'⊣¯1⎕DT⊂⍵
}
```

```
Check 2020 2 29
```

# Special events: exceptions

```
      Check 2020 2 29
Valid
```

```
Check←{
      0::⎕DMX.Message
      'Valid'⊣¯1⎕DT⊂⍵
}
```

# Special events: exceptions

```
      Check 2020 2 29
Valid

      Check 2021 2 29
```

```
Check←{
      0::⎕DMX.Message
      'Valid'⊣¯1⎕DT⊂⍵
}
```

# Special events: exceptions

```
Check←{
      0::⎕DMX.Message
      'Valid'⊣¯1⎕DT⊂⍵
}
```

```
      Check 2020 2 29
Valid

      Check 2021 2 29
Invalid date-time
```

# Special events: exceptions


```
      Check 2020 2 29
Valid

      Check 2021 2 29
Invalid date-time
```

```
Check←{
    0::⎕DMX.Message
    'Valid'⊣¯1⎕DT⊂⍵
}
```

# Special events: exceptions

.net

```
      Check 2020 2 29
Valid
```

```
Check←{
    ⎕USING←'System'
    0::⎕DMX.Message
    'Valid'⊣¯1⎕DT⊂⍵
}
```

# Special events: exceptions

```
      Check 2020 2 29
Valid
```

```
Check←{
    ⎕USING←'System'
    0::⎕DMX.Message
    'Valid'⊣⎕NEW DateTime ⍵
}
```

# Special events: exceptions

```
      Check 2020 2 29
Valid

      Check 2021 2 29
```

```
Check←{
    ⎕USING←'System'
    0::⎕DMX.Message
    'Valid'⊣⎕NEW DateTime ⍵
}
```

# Special events: exceptions


```
      Check 2020 2 29
Valid

      Check 2021 2 29
A .NET exception has occurred.
Check ⎕EXCEPTION for details
```

```
Check←{
    ⎕USING←'System'
    0::⎕DMX.Message
    'Valid'⊣⎕NEW DateTime ⍵
}
```

# Special events: exceptions

```
      Check 2020 2 29
Valid

      Check 2021 2 29
```

```
Check←{
    ⎕USING←'System'
    0::⎕EXCEPTION.Message
    'Valid'⊣⎕NEW DateTime ⍵
}
```

# Special events: exceptions

.net

```
      Check 2020 2 29
Valid

      Check 2021 2 29
Year, Month, and Day parameters
describe an un-representable
DateTime.
```

```
Check←{
    ⎕USING←'System'
    0::⎕EXCEPTION.Message
    'Valid'⊣⎕NEW DateTime ⍵
}
```

# Special events: exceptions

# Special events: exceptions

```
⎕EXCEPTION.⎕NL-⍳9
```

# Special events: exceptions

.net

```
      ⎕EXCEPTION.⎕NL-⍳9
 ActualValue  Data  Equals  GetBaseException  GetHashCode
GetObjectData  GetType  HResult  HelpLink  InnerException
Message  ParamName  ReferenceEquals  Source  StackTrace
TargetSite  ToString
```

# Special events: exceptions

.net

```
      ⎕EXCEPTION.⎕NL-⍳9
 ActualValue  Data  Equals  GetBaseException  GetHashCode
GetObjectData  GetType  HResult  HelpLink  InnerException
Message  ParamName  ReferenceEquals  Source  StackTrace
TargetSite  ToString
      ⎕EXCEPTION.Source
mscorlib
```

# Special events: exceptions

.net

```
      ⎕EXCEPTION.⎕NL-⍳9
 ActualValue  Data  Equals  GetBaseException  GetHashCode
GetObjectData  GetType  HResult  HelpLink  InnerException
Message  ParamName  ReferenceEquals  Source  StackTrace
TargetSite  ToString
      ⎕EXCEPTION.Source
mscorlib
      ⎕EXCEPTION.TargetSite
Int64 DateToTicks(Int32, Int32, Int32)
```

# Resetting System Constants

## cleaning up your mess

# Resetting system constants

# Resetting system constants

```
÷0
```

# Resetting system constants

```
      ÷0
DOMAIN ERROR: Divide by zero
      ÷0
      ^
```

# Resetting system constants

```
      ÷0
DOMAIN ERROR: Divide by zero
      ÷0
      ^
      ⎕EN,⎕DMX.ENX
```

# Resetting system constants

```
      ÷0
DOMAIN ERROR: Divide by zero
      ÷0
      ^
      ⎕EN,⎕DMX.ENX
11 1
```

# Resetting system constants

```
      ÷0
DOMAIN ERROR: Divide by zero
      ÷0
      ^
      ⎕EN,⎕DMX.ENX
11 1
      ⎕SIGNAL 0
```

# Resetting system constants

```
      ÷0
DOMAIN ERROR: Divide by zero
      ÷0
      ^
      ⎕EN,⎕DMX.ENX
11 1
      ⎕SIGNAL 0
      ⎕DMX
```

# Resetting system constants

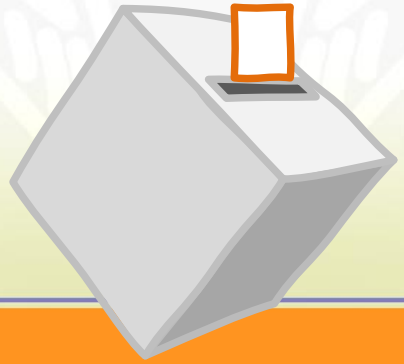
```
      ÷0
DOMAIN ERROR: Divide by zero
      ÷0
      ^
      ⎕EN,⎕DMX.ENX
11 1
      ⎕SIGNAL 0
      ⎕DMX
      ⎕EN,⎕DMX.ENX
```

# Resetting system constants

```
      ÷0
DOMAIN ERROR: Divide by zero
      ÷0
      ^
      ⎕EN,⎕DMX.ENX
11 1
      ⎕SIGNAL 0
      ⎕DMX
      ⎕EN,⎕DMX.ENX
0 0
```

# Poll: Event Message to Number

how would you like this to work?



# Messages instead of Numbers

# Messages instead of Numbers: function

```
ERROR←(' ERROR'⎕R''⎕EM⍳99)⍳⊆
```

# Messages instead of Numbers: function

```
ERROR←(' ERROR'⎕R''⎕EM⍳99)⍳⊆

:Trap ERROR 'RANK' 'LENGTH'
```

# Poll: event message to number

# Poll: event message to number

```
⍞←ems←⎕EM 4 5
```

# Poll: event message to number

```
⎕←ems←⎕EM 4 5
RANK ERROR  LENGTH ERROR
```

# Poll: event message to number

```
      ⎕←ems←⎕EM 4 5
 RANK ERROR   LENGTH ERROR
      ⎕EM⍕⍣¯1⊢ems
4 5
```

# Poll: event message to number

```
      ⎕←ems←⎕EM 4 5
 RANK ERROR  LENGTH ERROR
      ⎕EM⍕⍣¯1⊢ems
```

A

```
4 5
```

```
      ⎕EM ems
```

B

```
4 5
```

# Poll: event message to number

```
      ⎕←ems←⎕EM 4 5
 RANK ERROR  LENGTH ERROR
      ⎕EM⍕⍣¯1⊢ems
```

A

```
4 5
```

```
      ⎕EM ems
```

B

```
4 5
```

```
      ⎕SOMETHINGELSE ems
```

C

```
4 5
```

# Poll: event message to number

```
      ⎕←ems←⎕EM 4 5
 RANK ERROR  LENGTH ERROR
      ⎕EM⍣¯1⊢ems
```

A
```
4 5
```

```
      ⎕EM ems
```

B
```
4 5
```

```
      ⎕SOMETHINGELSE ems
```

C
```
4 5
```

D
```
:Trap 'RANK ERROR' 'LENGTH ERROR'
```

# Poll: event message to number

```
      ⎕←ems←⎕EM 4 5
 RANK ERROR  LENGTH ERROR
      ⎕EM⍣¯1⊢ems
```

A
```
4 5
```

```
      ⎕EM ems
```

B
```
4 5
```

```
      ⎕SOMETHINGELSE ems
```

C
```
4 5
```

D
```
:Trap 'RANK ERROR' 'LENGTH ERROR'
```

adam@dyalog.com

# Error Handling – Part 5

Internal tradfn signal `{⎕SIGNAL ⍵}`

Interrupts `1000::` or `1001...1008`

Exceptions `90::⎕EXCEPTION`

Resetting `⎕SIGNAL 0`

# Error Handling – Part 5

*Number for given message*
aplcart.info?q=:: message

Internal tradfn signal `{⎕SIGNAL ⍵}`

Interrupts `1000::` or `1001...1008`

Exceptions `90::⎕EXCEPTION`

Resetting `⎕SIGNAL 0`

# Upcoming webinars

| | | | | |
|---|---|---|---|---|
| Oct | 7 | BAA | open session | britishaplassociation.org |
| Oct | 21 | BAA | open session | |
| Nov | 8 | Dyalog '21 | dyalog.com/user-meetings/dyalog21.htm | |
| Nov | 9 | | | |
| Nov | 18 | BAA | open session | |
| Dec | 2 | BAA | open session | |
| Dec | 16 | BAA | open session | |
| Jan | 20 | Dyalog | webinar (topic TBA) | |