DEBUGGING

Coenraad Marinissen

c.marinissen@nikhef.nl

June 24, 2026

Amsterdam

DEBUGGING STANDARDS

1 — 2 — 3 — 4 — 5

1	2	3	4	5
What exactly failed?	**Isolate**	**Find root cause**	**Fix and test**	**What else?**
Is it reproducible?	Shrink the input: Find a minimal Form program.	Find the code path and state that make it fail.	Can the old behaviour never return?	Regression tests, related cases, docs, and PR cleanup.

ISSUE #828

1. What exactly failed?
2. Isolate
3. Find root cause
4. Fix/test
5. What else?

form-dev / form

Type / to search

Code | Issues 238 | Pull requests 16 | Agents | Discussions | Actions | Projects | Wiki | Security and quality | Insights | Settings

PushHide and PopHide together with regular Hide statement #828

New issue

Open

cbmarini opened on May 13 — Collaborator

What should the intended behaviour of the following program be?

```
Symbols x;
Local F1 = x+1;
.sort

PushHide;
Local F2 = x+2;
.sort

PushHide;
Local F3 = x+3;
.sort

Hide;
.sort

PopHide;
Print;
.end
```

At the moment, this results in

```
F1 =
   1 + x;

F2 =
   2 + x;

F3 =
   3 + x;
```

so all three expressions become active again.

The manual states the following about `PopHide`:

Undoes the action of the most recent pushhide statement (see 7.121). If there is no matching pushhide statement an error will result.

Based on this description, I would expect the final `PopHide;` to reactivate only `F2`.

Assignees
No one - Assign yourself

Labels
bug

Type
No type

Fields — Give feedback
No fields configured for issues without a type.

Projects
No projects

Milestone
No milestone

Relationships
None yet

Development
Create a branch for this issue or link a pull request.

Notifications — Customize
Unsubscribe
You're receiving notifications because you're subscribed to this thread.

Participants

Transfer issue

ISSUE #828

1. What exactly failed?
2. Isolate
3. Find root cause
4. Fix/test
5. What else?

form-dev / form

Type / to search

Code | Issues 238 | Pull requests 16 | Agents | Discussions | Actions | Projects | Wiki | Security and quality | Insights | Settings

PushHide and PopHide together with regular Hide statement #828

New issue

Open

tueda on May 13 — Last edited by **tueda** — Collaborator

> Interestingly, if `Hide;` is replaced by `Hide F3;` in the example above, then after `PopHide;` only `F2` and `F3` become active again.

By the way, `Hide F4;` gives the same result, i.e., no error occurs for unknown names.

tueda added **bug** on May 14

At the moment, this results in

```
F1 =
  1 + x;

F2 =
  2 + x;

F3 =
  3 + x;
```

so all three expressions become active again.

The manual states the following about `PopHide`:

Undoes the action of the most recent pushhide statement (see 7.121). If there is no matching pushhide statement an error will result.

Based on this description, I would expect the final `PopHide;` to reactivate only `F2`.

Relationships

None yet

Development

Create a branch for this issue or link a pull request.

Notifications — Customize

Unsubscribe

You're receiving notifications because you're subscribed to this thread.

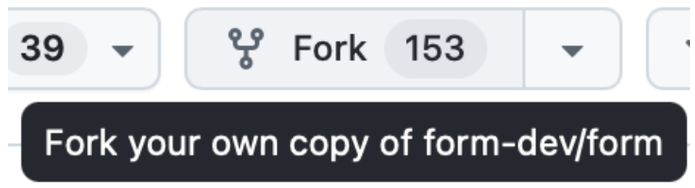
Participants

Transfer issue

GOOD PRACTICES

1 What exactly failed? — 2 Isolate — 3 Find root cause — 4 Fix/test — 5 What else?

- Work on your own fork



- Create new branch
  - Iterate there, `push --force`, etc
- Fix the bug
- Create a clean pull request back into `form-dev/form`.
- After merging, you can safely close your branch.

FINDING A MINIMAL PROGRAM

1 What exactly failed? — **2 Isolate** — 3 Find root cause — 4 Fix/test — 5 What else?

Full program

Large input
many modules

→

Minimal program

Small input
stable failure

→

Regression test

Known output
Fast enough for CI

- What must remain for the failure to occur?

  Cut input size first; then cut code paths.

  Keep the physics needed to trigger the bug; remove everything else.

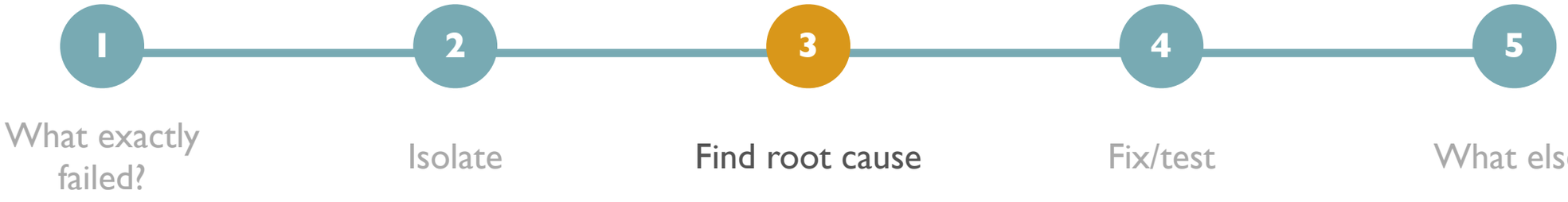
- When the minimal program is stable, it becomes the first regression test.

ISSUE #828: FINDING A MINIMAL PROGRAM

1. What exactly failed?
2. Isolate
3. Find root cause
4. Fix/test
5. What else?

```
    Local F = 1;
    .sort

Time =       0.00 sec    Generated terms =          1
               F         Terms in output =          1
                         Bytes used      =         20

    Hide G;
    Print;
    .end

Time =       0.00 sec    Generated terms =          1
               F         Terms in output =          1
                         Bytes used      =         20

   F =
      1;

  0.00 sec out of 0.00 sec
```

ISSUE #828: CLASSIFY THE FAILURE



- Build?
  - ➡ Compiler error? Linker flags?
- Run time bug?
  - ➡ Wrong result? Hang? Segfault?
- Compiler bug?
  - ➡ Look in functions that start with `Co`
- Regression?
  - ➡ Used to work, but not anymore
  - ➡ Slower than before?

ISSUE #828: CLASSIFY THE FAILURE

1. What exactly failed?
   Easily reproducible?
2. Isolate
   What is the smallest Form program?
3. **Find root cause**
   What breaks Form?
4. Fix/test
   Can the old failure never return?
5. What else?
   Update manual? Regression test(s)/ coverage

Most likely a compiler bug

- Lets use the debugger
- Set a breakpoint in `CoHide()` in `compcomm.c`

DEBUG BUILD AND DEBUGGERS

- Configuring Form for debugging

Small performance cost

```
$ ./configure --enable-debug --enable-backtrace
$ make -j
```

- This makes the executables

```
form     tform     tvorm     vorm
```

- Run debugger:

  - Standard debugger on linux: GDB (GNU Debugger)

```
$ gdb ./vorm
```

  - Standard debugger on macOS: LLDB (LLVM Debugger)

```
$ lldb ./vorm
```

- Step through the code and print the call stack, variables, watch and a lot more
- This can be setup in vscode nicely, see tips for developer's on the wiki.

DEBUGGERS

Action	GDB	LLDB
Start program	`gdb ./vorm`	`lldb ./vorm`
Run with arguments	`run program.frm`	`run program.frm`
Continue	`continue (c)`	`continue (c)`
Step into	`step (s)`	`step (s)`
Step over	`next (n)`	`next (n)`
Breakpoint function	`break Normalize`	`b Normalize`
Breakpoint file:line	`break sort.c:123`	`b sort.c:123`
Variable watch	`watch <variable>`	`watchpoint set variable <variable>`
Finish current function	`finish`	`finish`
Backtrace	`bt`	`bt`
Print variable	`print x`	`p x`
List source	`list`	`source list`
Quit	`quit`	`quit`
.	.	.
.	.	.
.	.	.

Debugger walkthrough 01: Stop at CoHide()

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - inp = "G"
  - ScratchBuf = <variable not available>
- Static
- Global
- Registers

CALL STACK — Paused on breakpoint

CoHide	compcomm.c	1596:18
CompileStatement	compiler.c	696:11
PreProcessor	pre.c	1130:17
main	startup.c	1819:2
start	@start	1751

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

compcomm.c 2

sources > C compcomm.c > CoHide(UBYTE *)

```
1590      #[ CoHide :
1591    */
1592
1593    int CoHide(UBYTE *inp) {
1594        GETIDENTITY
1595        WORD *ScratchBuf;
1596        if ( AR.Fscr[2].PObuffer == 0 ) {      form, 22 years ago • Initial revision
1597            ScratchBuf = (WORD *)Malloc1(AM.HideSize*sizeof(WORD),"hidesize");
1598            AR.Fscr[2].POsize = AM.HideSize * sizeof(WORD);
1599            AR.Fscr[2].POfull = AR.Fscr[2].POfill = AR.Fscr[2].PObuffer = ScratchBuf;
1600            AR.Fscr[2].POstop = AR.Fscr[2].PObuffer + AM.HideSize;
1601            PUTZERO(AR.Fscr[2].POposition);
1602        }
1603        return(SetExpr(inp,1,HIDE));
1604    }
1605
1606    /*
1607        #] CoHide :
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
               F        Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 02: allocation of the hide file

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - inp = "G"
  - ScratchBuf = <variable not available>
- Static
- Global
- Registers

CALL STACK — Paused on step
- CoHide compcomm.c 1597:35

Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > CoHide(UBYTE *)

```
1590      #[ CoHide :
1591    */
1592
1593    int CoHide(UBYTE *inp) {
1594        GETIDENTITY
1595        WORD *ScratchBuf;
1596        if ( AR.Fscr[2].PObuffer == 0 ) {
1597            ScratchBuf = (WORD *)Malloc1(AM.HideSize*sizeof(WORD),"hidesize");
1598            AR.Fscr[2].POsize = AM.HideSize * sizeof(WORD);
1599            AR.Fscr[2].POfull = AR.Fscr[2].POfill = AR.Fscr[2].PObuffer = ScratchBuf;
1600            AR.Fscr[2].POstop = AR.Fscr[2].PObuffer + AM.HideSize;
1601            PUTZERO(AR.Fscr[2].POposition);
1602        }
1603        return(SetExpr(inp,1,HIDE));
1604    }
1605
1606    /*
1607        #] CoHide :
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
               F        Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 03: allocation of the hide file

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - inp = "G"
  - ScratchBuf = 0
- Static
- Global
- Registers

CALL STACK — Paused on step
- CoHide compcomm.c 1598:26

Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > CoHide(UBYTE *)

```
1590      #[ CoHide :
1591    */
1592
1593    int CoHide(UBYTE *inp) {
1594        GETIDENTITY
1595        WORD *ScratchBuf;
1596        if ( AR.Fscr[2].PObuffer == 0 ) {
1597            ScratchBuf = (WORD *)Malloc1(AM.HideSize*sizeof(WORD),"hidesize");
1598            AR.Fscr[2].POsize = AM.HideSize * sizeof(WORD);      t68, 17 years ago • *** empt
1599            AR.Fscr[2].POfull = AR.Fscr[2].POfill = AR.Fscr[2].PObuffer = ScratchBuf;
1600            AR.Fscr[2].POstop = AR.Fscr[2].PObuffer + AM.HideSize;
1601            PUTZERO(AR.Fscr[2].POposition);
1602        }
1603        return(SetExpr(inp,1,HIDE));
1604    }
1605
1606    /*
1607        #] CoHide :
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
              F         Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 04: allocation of the hide file

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - inp = "G"
  - ScratchBuf = 0
- Static
- Global
- Registers

CALL STACK — Paused on step
- CoHide compcomm.c 1599:63

Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > CoHide(UBYTE *)

```
1590      #[ CoHide :
1591    */
1592
1593    int CoHide(UBYTE *inp) {
1594        GETIDENTITY
1595        WORD *ScratchBuf;
1596        if ( AR.Fscr[2].PObuffer == 0 ) {
1597            ScratchBuf = (WORD *)Malloc1(AM.HideSize*sizeof(WORD),"hidesize");
1598            AR.Fscr[2].POsize = AM.HideSize * sizeof(WORD);
1599            AR.Fscr[2].POfull = AR.Fscr[2].POfill = AR.Fscr[2].PObuffer = ScratchBuf;
1600            AR.Fscr[2].POstop = AR.Fscr[2].PObuffer + AM.HideSize;
1601            PUTZERO(AR.Fscr[2].POposition);
1602        }
1603        return(SetExpr(inp,1,HIDE));
1604    }
1605
1606    /*
1607        #] CoHide :
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
               F        Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 05: allocation of the hide file

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - inp = "G"
  - ScratchBuf = 0
- Static
- Global
- Registers

CALL STACK — Paused on step
- CoHide compcomm.c 1600:43

Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > CoHide(UBYTE *)

```
1590      #[ CoHide :
1591    */
1592
1593    int CoHide(UBYTE *inp) {
1594        GETIDENTITY
1595        WORD *ScratchBuf;
1596        if ( AR.Fscr[2].PObuffer == 0 ) {
1597            ScratchBuf = (WORD *)Malloc1(AM.HideSize*sizeof(WORD),"hidesize");
1598            AR.Fscr[2].POsize = AM.HideSize * sizeof(WORD);
1599            AR.Fscr[2].POfull = AR.Fscr[2].POfill = AR.Fscr[2].PObuffer = ScratchBuf;
1600            AR.Fscr[2].POstop = AR.Fscr[2].PObuffer + AM.HideSize;      t68, 17 years ago •
1601            PUTZERO(AR.Fscr[2].POposition);
1602        }
1603        return(SetExpr(inp,1,HIDE));
1604    }
1605
1606    /*
1607        #] CoHide :
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
               F        Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 06: allocation of the hide file

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - inp = "G"
  - ScratchBuf = 0
- Static
- Global
- Registers

CALL STACK — Paused on step
- CoHide compcomm.c 1601:3

Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > CoHide(UBYTE *)

```
1590      #[ CoHide :
1591    */
1592
1593    int CoHide(UBYTE *inp) {
1594        GETIDENTITY
1595        WORD *ScratchBuf;
1596        if ( AR.Fscr[2].PObuffer == 0 ) {
1597            ScratchBuf = (WORD *)Malloc1(AM.HideSize*sizeof(WORD),"hidesize");
1598            AR.Fscr[2].POsize = AM.HideSize * sizeof(WORD);
1599            AR.Fscr[2].POfull = AR.Fscr[2].POfill = AR.Fscr[2].PObuffer = ScratchBuf;
1600            AR.Fscr[2].POstop = AR.Fscr[2].PObuffer + AM.HideSize;
1601            PUTZERO(AR.Fscr[2].POposition);      form, 22 years ago • Initial revision
1602        }
1603        return(SetExpr(inp,1,HIDE));
1604    }
1605
1606    /*
1607        #] CoHide :
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
               F        Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 07: Continue in SetExpr

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - inp = <variable not available>
  - ScratchBuf = 0
- Static
- Global
- Registers

CALL STACK
- CoHide compcomm.c 1603:9

Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

compcomm.c 2

sources > compcomm.c > CoHide(UBYTE *)

```
1608 >    #[ CoIntoHide :…
1591  */
1592
1593  int CoHide(UBYTE *inp) {
1594      GETIDENTITY
1595      WORD *ScratchBuf;
1596      if ( AR.Fscr[2].PObuffer == 0 ) {
1597          ScratchBuf = (WORD *)Malloc1(AM.HideSize*sizeof(WORD),"hidesize");
1598          AR.Fscr[2].POsize = AM.HideSize * sizeof(WORD);
1599          AR.Fscr[2].POfull = AR.Fscr[2].POfill = AR.Fscr[2].PObuffer = ScratchBuf;
1600          AR.Fscr[2].POstop = AR.Fscr[2].PObuffer + AM.HideSize;
1601          PUTZERO(AR.Fscr[2].POposition);
1602      }
1603      return( SetExpr(inp,1,HIDE));      form, 22 years ago • Initial revision
1604  }
1605
1606  /*
1607      #] CoHide :
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS … (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
               F        Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 08: Continue in SetExpr

RUN AND DEBUG (lldb) v...

VARIABLES

Local

- s = "G"
- setunset = 1
- par = 3
- error = <variable not available>
- name = <variable not available>
- c = <variable not available>
- w = <variable not available>
- numexpr = <variable not available>
- i = <variable not available>

Static

Global

CALL STACK

SetExpr	compcomm.c	1521
CompileStatement	compiler.c	696:11
PreProcessor	pre.c	1130:17
main	startup.c	1819:2
start	@start	1751

WATCH

BREAKPOINTS

- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

compcomm.c 2

sources > compcomm.c > SetExpr(UBYTE *, int, int)

```
1562 >      #[ CoDrop :…
1518    */
1519
1520    int SetExpr(UBYTE *s, int setunset, int par)
1521    {       form, 22 years ago • Initial revision
1522        WORD *w, numexpr;
1523        int error = 0, i;
1524        UBYTE *name, c;
1525        if ( *s == 0 ) {
1526            for ( i = 0; i < NumExpressions; i++ ) {
1527                w = &(Expressions[i].status);
1528                *w = SetExprCases(par,setunset,*w);
1529                if ( *w < 0 ) error = 1;
1530                if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1531                    Expressions[i].hidelevel = AC.HideLevel;
1532            }
1533            return(0);
1534        }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS ··· (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:19:03 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
              F         Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 09: Hide; without expression names

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - s = "G"
  - setunset = 1
  - par = 3
  - error = 0
  - name = <variable not available>
  - c = <variable not available>
  - w = <variable not available>
  - numexpr = <variable not available>
  - i = <variable not available>
- Static
- Global

CALL STACK
- SetExpr compcomm.c 1525:7
- Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > SetExpr(UBYTE *, int, int)

```
1517      #[ SetExpr :
1518    */
1519
1520    int SetExpr(UBYTE *s, int setunset, int par)
1521    {
1522        WORD *w, numexpr;
1523        int error = 0, i;
1524        UBYTE *name, c;
1525        if ( *s == 0 ) {      Josh Davies, 11 months ago • fix: make IntoHide; work as descri
1526            for ( i = 0; i < NumExpressions; i++ ) {
1527                w = &(Expressions[i].status);
1528                *w = SetExprCases(par,setunset,*w);
1529                if ( *w < 0 ) error = 1;
1530                if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1531                    Expressions[i].hidelevel = AC.HideLevel;
1532            }
1533            return(0);
1534        }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS ... (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
              F         Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 10: Hide [expression name];

RUN AND DEBUG (lldb) v...

VARIABLES

Local

- s = "G"
- setunset = 1
- par = 3
- error = 0
- name = ""
- c = '\xff'
- w = <null>
- numexpr = <variable not available>
- i = <variable not available>

Static

Global

CALL STACK — Paused on step

SetExpr compcomm.c 1535:10

Load More Stack Frames

WATCH

BREAKPOINTS

- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > SetExpr(UBYTE *, int, int)

```
1517    #[ SetExpr :
1535    while ( *s ) {      form, 22 years ago • Initial revision
1536        if ( *s == ',' ) { s++; continue; }
1537        if ( *s == '0' ) { s++; continue; }
1538        name = s;
1539        if ( ( s = SkipAName(s) ) == 0 ) {
1540            MesPrint("&Improper name for an expression: '%s'",name);
1541            return(1);
1542        }
1543        c = *s; *s = 0;
1544        if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545            w = &(Expressions[numexpr].status);
1546            *w = SetExprCases(par,setunset,*w);
1547            if ( *w < 0 ) error = 1;
1548            if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549                Expressions[numexpr].hidelevel = AC.HideLevel;
1550        }
1551        else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =       0.00 sec    Generated terms =          1
               F         Terms in output =          1
                         Bytes used      =         20

    Hide G;
```

Debugger walkthrough 11: Step over the name of the expression

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - s = "G"
  - setunset = 1
  - par = 3
  - error = 0
  - name = "G"
  - c = '\xff'
  - w = <null>
  - numexpr = <variable not available>
  - i = <variable not available>
- Static
- Global

CALL STACK — Paused on step
- SetExpr compcomm.c 1539:14

Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > SetExpr(UBYTE *, int, int)

```
1517    #[ SetExpr :
1535    while ( *s ) {
1536        if ( *s == ',' ) { s++; continue; }
1537        if ( *s == '0' ) { s++; continue; }
1538        name = s;
1539        if ( ( s = SkipAName(s) ) == 0 ) {      form, 22 years ago • Initial revision
1540            MesPrint("&Improper name for an expression: '%s'",name);
1541            return(1);
1542        }
1543        c = *s; *s = 0;
1544        if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545            w = &(Expressions[numexpr].status);
1546            *w = SetExprCases(par,setunset,*w);
1547            if ( *w < 0 ) error = 1;
1548            if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549                Expressions[numexpr].hidelevel = AC.HideLevel;
1550        }
1551        else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
              F         Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 12: Set character after the name to 0.

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - s = ""
  - setunset = 1
  - par = 3
  - error = 0
  - name = "G"
  - c = '\xff'
  - w = <null>
  - numexpr = <variable not available>
  - i = <variable not available>
- Static
- Global

CALL STACK
SetExpr compcomm.c 1543:7
Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > SetExpr(UBYTE *, int, int)

```
1517    #[ SetExpr :
1535    while ( *s ) {
1536        if ( *s == ',' ) { s++; continue; }
1537        if ( *s == '0' ) { s++; continue; }
1538        name = s;
1539        if ( ( s = SkipAName(s) ) == 0 ) {
1540            MesPrint("&Improper name for an expression: '%s'",name);
1541            return(1);
1542        }
1543        c = *s; *s = 0;      form, 22 years ago • Initial revision
1544        if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545            w = &(Expressions[numexpr].status);
1546            *w = SetExprCases(par,setunset,*w);
1547            if ( *w < 0 ) error = 1;
1548            if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549                Expressions[numexpr].hidelevel = AC.HideLevel;
1550        }
1551        else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS ... (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =       0.00 sec    Generated terms =          1
               F         Terms in output =          1
                         Bytes used      =         20

    Hide G;
```

Debugger walkthrough 13: Read the expression name

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - s = ""
  - setunset = 1
  - par = 3
  - error = 0
  - name = "G"
  - c = <variable not available>
  - w = <null>
  - numexpr = <variable not available>
  - i = <variable not available>
- Static
- Global

CALL STACK
- SetExpr compcomm.c 1544:19
- Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > SetExpr(UBYTE *, int, int)

```
1562    #[ CoDrop :…
1535    while ( *s ) {
1536        if ( *s == ',' ) { s++; continue; }
1537        if ( *s == '0' ) { s++; continue; }
1538        name = s;
1539        if ( ( s = SkipAName(s) ) == 0 ) {
1540            MesPrint("&Improper name for an expression: '%s'",name);
1541            return(1);
1542        }
1543        c = *s; *s = 0;
1544        if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {      t68, 18
1545            w = &(Expressions[numexpr].status);
1546            *w = SetExprCases(par,setunset,*w);
1547            if ( *w < 0 ) error = 1;
1548            if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549                Expressions[numexpr].hidelevel = AC.HideLevel;
1550        }
1551        else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS … (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:08:42 2026
    Local F = 1;
    .sort

Time =       0.00 sec    Generated terms =          1
               F         Terms in output =          1
                         Bytes used      =         20

    Hide G;
```

Debugger walkthrough 14: branch if the expression name was not declared earlier

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - s = ""
  - setunset = 1
  - par = 3
  - error = 0
  - name = "G"
  - c = <variable not available>
  - w = <null>
  - numexpr = 1
  - i = <variable not available>
- Static
- Global

CALL STACK
- SetExpr compcomm.c 1551:24

Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > SetExpr(UBYTE *, int, int)

```
1517    #[ SetExpr :
1543        c = *s; *s = 0;
1544        if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545            w = &(Expressions[numexpr].status);
1546            *w = SetExprCases(par,setunset,*w);
1547            if ( *w < 0 ) error = 1;
1548            if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549                Expressions[numexpr].hidelevel = AC.HideLevel;
1550        }
1551        else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
1552            MesPrint("&%s is not an expression",name);
1553            error = 1;
1554        }
1555        *s = c;
1556    }
1557    return(error);
1558  }
1559
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:15:36 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
              F         Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 15: Why do we skip over it?

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - s = ""
  - setunset = 1
  - par = 3
  - error = 0
  - name = "G"
  - c = <variable not available>
  - w = <null>
  - numexpr = <variable not available>
  - i = <variable not available>
- Static
- Global

CALL STACK — Paused on step
- SetExpr compcomm.c 1555:6

Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

compcomm.c 2

sources > compcomm.c > SetExpr(UBYTE *, int, int)

```
1517    #[ SetExpr :
1543        c = *s; *s = 0;
1544        if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545            w = &(Expressions[numexpr].status);
1546            *w = SetExprCases(par,setunset,*w);
1547            if ( *w < 0 ) error = 1;
1548            if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549                Expressions[numexpr].hidelevel = AC.HideLevel;
1550        }
1551        else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
1552            MesPrint("&%s is not an expression",name);
1553            error = 1;
1554        }
1555        *s = c;      form, 22 years ago • Initial revision
1556    }
1557    return(error);
1558    }
1559
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:15:36 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
              F         Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 16: Check if more expression names are given

RUN AND DEBUG (lldb) v...

VARIABLES
- Local
  - s = ""
  - setunset = 1
  - par = 3
  - error = 0
  - name = <variable not available>
  - c = <variable not available>
  - w = <null>
  - numexpr = <variable not available>
  - i = <variable not available>
- Static
- Global

CALL STACK
- SetExpr compcomm.c 1535:2
- Load More Stack Frames

WATCH

BREAKPOINTS
- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > SetExpr(UBYTE *, int, int)

```
1517    #[ SetExpr :
1535    while ( *s ) {      form, 22 years ago • Initial revision
1536        if ( *s == ',' ) { s++; continue; }
1537        if ( *s == '0' ) { s++; continue; }
1538        name = s;
1539        if ( ( s = SkipAName(s) ) == 0 ) {
1540            MesPrint("&Improper name for an expression: '%s'",name);
1541            return(1);
1542        }
1543        c = *s; *s = 0;
1544        if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545            w = &(Expressions[numexpr].status);
1546            *w = SetExprCases(par,setunset,*w);
1547            if ( *w < 0 ) error = 1;
1548            if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549                Expressions[numexpr].hidelevel = AC.HideLevel;
1550        }
1551        else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS ... (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:15:36 2026
    Local F = 1;
    .sort

Time =       0.00 sec    Generated terms =          1
               F         Terms in output =          1
                         Bytes used      =         20

    Hide G;
```

Debugger walkthrough 17: No more expression names: return

RUN AND DEBUG (lldb) v...

VARIABLES

Local

- s = ""
- setunset = 1
- par = 3
- error = 0
- name = "G"
- c = '\0'
- w = <null>
- numexpr = 1
- i = 136

Static

Global

CALL STACK — Paused on step

SetExpr compcomm.c 1558:1

Load More Stack Frames

WATCH

BREAKPOINTS

- C++: on throw
- C++: on catch
- compcomm.c sources 1596

MODULES

EXCLUDED CALLERS

C compcomm.c 2

sources > C compcomm.c > SetExpr(UBYTE *, int, int)

```
1517    #[ SetExpr :
1543        c = *s; *s = 0;
1544        if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545            w = &(Expressions[numexpr].status);
1546            *w = SetExprCases(par,setunset,*w);
1547            if ( *w < 0 ) error = 1;
1548            if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549                Expressions[numexpr].hidelevel = AC.HideLevel;
1550        }
1551        else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
1552            MesPrint("&%s is not an expression",name);
1553            error = 1;
1554        }
1555        *s = c;
1556    }
1557    return(error);
1558  }     form, 22 years ago • Initial revision
1559
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS ··· (lldb) vorm test.frm

```
FORM 5.0.0-beta.1 (Jun  4 2026, v5.0.0-beta.1-369-g6318119)  Run: Tue Jun 16 21:15:36 2026
    Local F = 1;
    .sort

Time =      0.00 sec    Generated terms =          1
              F         Terms in output =          1
                        Bytes used      =         20

    Hide G;
```

Debugger walkthrough 18: No more expression names: return

RUN AND DEBUG (lldb) v...

VARIABLES

CALL STACK

WATCH

BREAKPOINTS

- C++: on throw
- C++: on catch
- compcomm.c sources 1593

MODULES

EXCLUDED CALLERS

compcomm.c 2

sources > compcomm.c > SetExpr(UBYTE *, int, int)

```
1517    #[ SetExpr :
1543        c = *s; *s = 0;
1544        if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545            w = &(Expressions[numexpr].status);
1546            *w = SetExprCases(par,setunset,*w);
1547            if ( *w < 0 ) error = 1;
1548            if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549                Expressions[numexpr].hidelevel = AC.HideLevel;
1550        }
1551        else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
1552            MesPrint("&%s is not an expression",name);
1553            error = 1;
1554        }
1555        *s = c;
1556    }
1557    return(error);
1558    }      form, 22 years ago • Initial revision
1559
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS (lldb) vorm test.frm

```
    Hide G;
    Print;
    .end

Time =      0.00 sec    Generated terms =          1
               F        Terms in output =          1
                        Bytes used      =         20

   F =
      1;

  0.00 sec out of 73.84 sec
 form git:(issue-828) ✗
```

ISSUE #828: FIXING

1 What exactly failed? — 2 Isolate — **3 Find root cause** — 4 Fix/test — 5 What else?

```
1544   if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545       w = &(Expressions[numexpr].status);
1546       *w = SetExprCases(par,setunset,*w);
1547       if ( *w < 0 ) error = 1;
1548       if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549           Expressions[numexpr].hidelevel = AC.HideLevel;
1550   }
1551   else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
1552       MesPrint("&%s is not an expression",name);
1553       error = 1;
1554   }
```

ISSUE #828: FIXING

1 What exactly failed? — 2 Isolate — 3 Find root cause — 4 Fix/test — 5 What else?

```
1544  1544      if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545  1545          w = &(Expressions[numexpr].status);
1546  1546          *w = SetExprCases(par,setunset,*w);
1547  1547          if ( *w < 0 ) error = 1;
1548  1548          if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549  1549              Expressions[numexpr].hidelevel = AC.HideLevel;
1550  1550      }
1551       -      else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
     1551+      else {
1552  1552          MesPrint("&%s is not an expression",name);
1553  1553          error = 1;
1554  1554      }
```

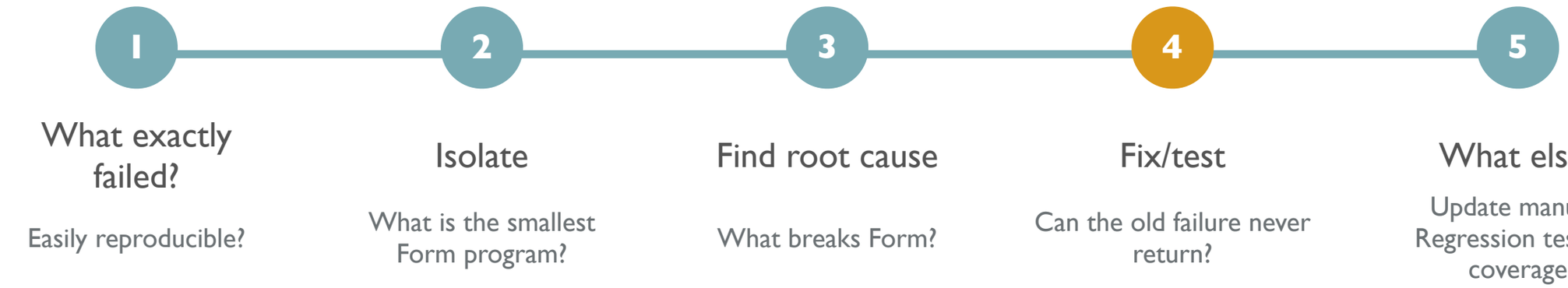
ISSUE #828: FIXING

1 What exactly failed? — 2 Isolate — 3 Find root cause — 4 Fix/test — 5 What else?

```
1544  1544      if ( GetName(AC.exprnames,name,&numexpr,NOAUTO) == CEXPRESSION ) {
1545  1545          w = &(Expressions[numexpr].status);
1546  1546          *w = SetExprCases(par,setunset,*w);
1547  1547          if ( *w < 0 ) error = 1;
1548  1548          if ( ( par == HIDE || par == INTOHIDE ) && setunset == 1 )
1549  1549              Expressions[numexpr].hidelevel = AC.HideLevel;
1550  1550      }
1551       -      else if ( GetName(AC.varnames,name,&numexpr,NOAUTO) != NAMENOTFOUND ) {
     1551+      else {
1552  1552          MesPrint("&%s is not an expression",name);
1553  1553          error = 1;
1554  1554      }
```

```
   Local F = 1;
    .sort

Time =       0.00 sec    Generated terms =          1
               F         Terms in output =          1
                         Bytes used      =         20

    Hide G;
test.frm Line 4 --> G is not an expression
    Print;
    .end
Program terminating at  -->
Terminate called from ../../sources/pre.c:1086 (PreProcessor)
  0.00 sec out of 0.00 sec
```

ISSUE #828: INTENDED BEHAVIOUR?



- What should the behaviour be?
  - ➡ Should `Hide G;` terminate,
  - ➡ or should it only print a warning message?

```
    Local F = 1;
    Print G;
test.frm Line 3 --> G is not the name of an active expression
    .sort

    Multiply 2;
    Print;

   F =
      2;

    .end
```

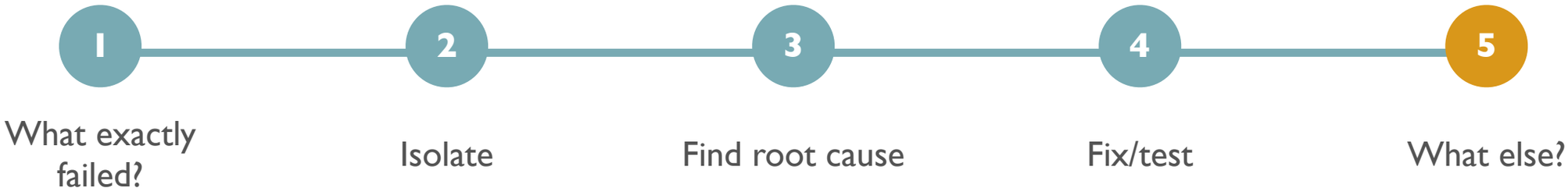
ISSUE #828: INTENDED BEHAVIOUR?

1. **What exactly failed?** — Easily reproducible?
2. **Isolate** — What is the smallest Form program?
3. **Find root cause** — What breaks Form?
4. **Fix/test** — Can the old failure never return?
5. **What else?** — Update manual? Regression test(s)/ coverage

- What should the behaviour be?
  - ➡ Should `Hide G;` terminate,
  - ➡ or should it only print a warning message?
- Is this code specific to only this issue?

```
   Local F = 1;
    .sort

Time =       0.00 sec    Generated terms =          1
               F         Terms in output =          1
                         Bytes used      =         20

    Drop G;
test.frm Line 4 --> G is not an expression
    Print;
    .end
Program terminating at  -->
Terminate called from ../../sources/pre.c:1086 (PreProcessor)
  0.00 sec out of 0.00 sec
```

ISSUE #828: REGRESSION TESTS



- Make regression test
  - ➡ Minimal Form program of earlier is a first good test.
  - ➡ Add the test to `check/fixes.frm`
  - ➡ Don't forget to add tests for `Drop G;` etc as well.
- Does the manual need updating?

ISSUE #828: PULL REQUEST

- Run the test suite before opening a PR:

```
make check
```

- Update your fork
  - ➡ On master branch of your fork:

```
git pull
```

  - ➡ Rebase the master

```
git switch issue-828
git rebase master
```

  - ➡ Push to remote

```
git push --force-with-lease origin issue-828
```

- Open PR, easily done from your fork on GitHub:
  - ➡ "This PR fixes #828..."

This automatically closes the issue after merging of the PR

- If the PR is already open, it is automatically updated, CI runs again etc.