

GUI for the Promotion of Critical Flight Software to ECSS Category A

ESA Software Product Assurance
Conference 2025

Baptiste Maheut

2025-09-23



1. Motivation
2. Introducing The Category A Qualification Visualizer
3. Use Case: An ARM Example
4. Outcome

Motivation

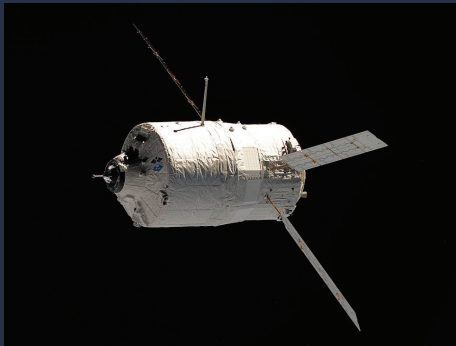


Figure 1: Automated Transfer Vehicle



Figure 2: European Service Module



Figure 3: Lunar I-Hab



Figure 4: ESA Space-Rider

Requirements defined by **ECSS-E-ST-40C-Rev.1** and **ECSS-Q-ST-80C-Rev.2**

1. Satisfy requirements for ECSS Category B software
2. 100% source code Modified Condition and Decision Coverage
3. Complete object-to-source code traceability
4. 100% structural coverage of object code (if 3. cannot be verified)

Problem

- Missing a systematic approach
- Missing a clear set of qualification tools

The ESM PDE relies on ad-hoc tools to gather traceability information¹

¹§4.7 in Critical Software for Human Spaceflight, Preden, Kaschner, Rettig, Rodriggs, 2019

One solution is to follow GTD GmbH's **Qualification Promotion Guidelines**²

²<https://data.gtd-gmbh.de/ecss-cat-a-guideline/>

Pros

- Define a generic Category-A qualification process
- Establish a solid theoretical background
- Provide detailed examples

Cons

- Lack a linear sequence of steps to be followed
- Require project-dependent actions
- Use many command-line tools

Introducing The Category A Qualification Visualizer

Contractual Context

The work is been carried out under ESA Contract No. 4000146271/24/NL/AR/ahh since 2024.

- Qualify the guidelines and the tools it recommends to DO-330 TQL-5

Qualified following **DO-178C** and **DO-330**

Tool Qualification Level 5

- The tool's output does not become part of the flight-software
- The tool's output is not used to justify the elimination or reduction of any verification process(es) other than the one automated by the tool nor any development process(es) that could have an impact on the flight-software.³

Delivered with: TQP (*Plan*), TOR (*Requirements*), and TVP (*Verification*)

³§4.4 in MP-TQP.00-QT

Its goal

Analyze flight software binary files to identify code introduced by the process of production that has not been verified nor tested.

Easy-to-use GUI for SW engineers developing the software, and PA engineers verifying the activities

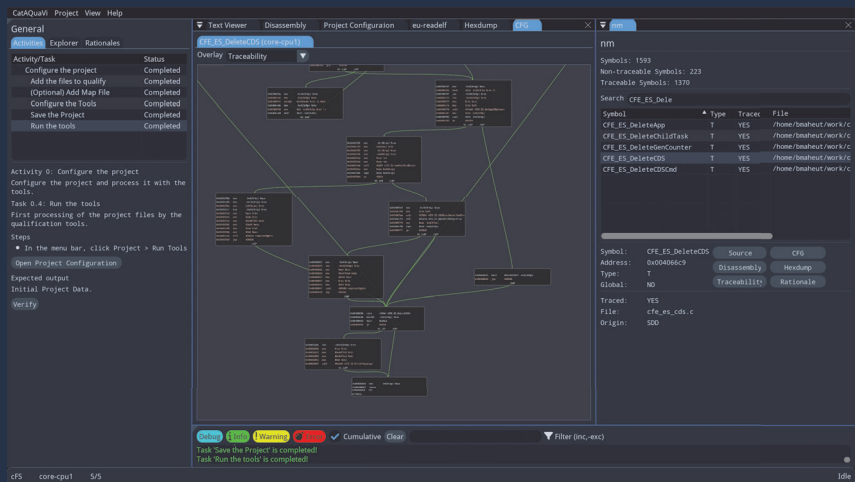


Figure 5: Overview of the GUI displaying one function of the NASA cFS framework

What it can do

- Identify all instructions, functions, and sections in a FSW executable
- Visualize Object Code Coverage and Object-To-Source Code Traceability
- Provide a checklist of activities to be done by the user
- Generate reports with evidence for ECSS Category A qualification

What it cannot do

- MC/DC verification. Use *mcdc-checker* and *gcov* instead

Project- and Target- Independent Tool

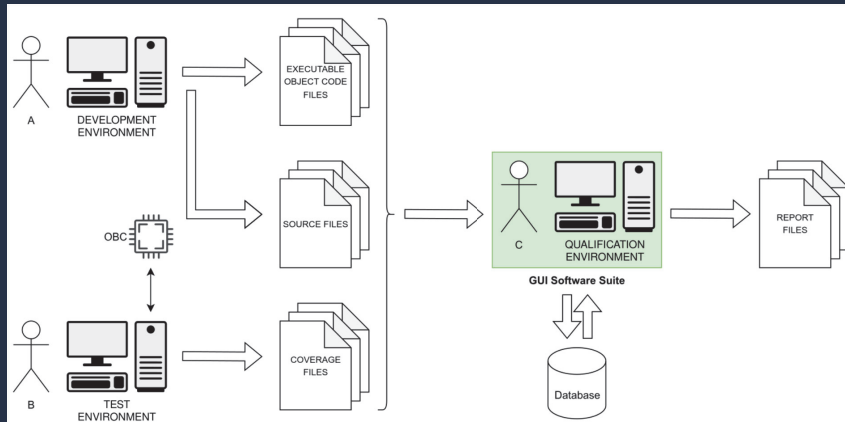


Figure 6: Typical operational environment of the GUI

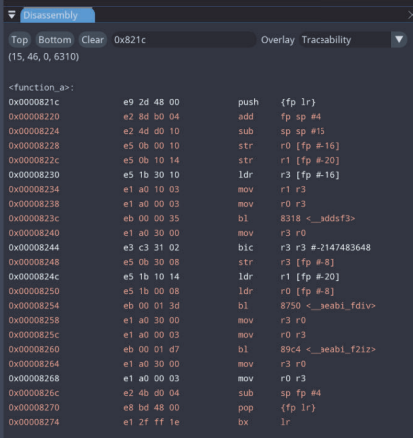
Use Case: An ARM Example

```
$ arm-none-eabi-objdump --disassemble=function_a a.out
```

```
0000821c <function_a>:
821c: e92d4800      push    {fp, lr}
8220: e28db004      add     fp, sp, #4
8224: e24dd010      sub     sp, sp, #16
8228: e50b0010      str     r0, [fp, #-16]
822c: e50b1014      str     r1, [fp, #-20] ; 0xffffffffec
8230: e51b3010      ldr     r3, [fp, #-16]
8234: e1a01003      mov     r1, r3
8238: e1a00003      mov     r0, r3
823c: eb000035      bl      8318 <__addsf3>
8240: e1a03000      mov     r3, r0
8244: e3c33102      bic     r3, r3, #-2147483648 ; 0x80000000
8248: e50b3008      str     r3, [fp, #-8]
824c: e51b1014      ldr     r1, [fp, #-20] ; 0xffffffffec
8250: e51b0008      ldr     r0, [fp, #-8]
8254: eb00013d      bl      8750 <__aeabi_fdiv>
8258: e1a03000      mov     r3, r0
825c: e1a00003      mov     r0, r3
8260: eb0001d7      bl      89c4 <__aeabi_f2iz>
8264: e1a03000      mov     r3, r0
8268: e1a00003      mov     r0, r3
826c: e24bd004      sub     sp, fp, #4
8270: e8bd4800      pop     {fp, lr}
8274: e12ffffe      bx      lr
```

```
$ eu-readelf --debug-dump=decodedline a.out
```

```
CU [b] example.c
line:col SBPE* disc isa op address (Statement Block Prologue Epilogue *End)
/home/bmaheut/work/swpa_example/example.c (mtime: 0, length: 0)
11:33 S      0 0 0x0000821c <function_a>
15:22 S      0 0 0x00008230 <function_a+0x14>
15:8 S       0 0 0x00008244 <function_a+0x28>
17:13 S      0 0 0x0000824c <function_a+0x30>
18:1 S       0 0 0x00008268 <function_a+0x4c>
```



Address	Instruction	Comment
0x0000821c	e9 2d 48 00	push {fp, lr}
0x00008220	e2 8d b0 04	add fp, sp, #4
0x00008224	e2 4d d0 10	sub sp, sp, #16
0x00008228	e5 0b 00 10	str r0, [fp, #-16]
0x0000822c	e5 0b 10 14	str r1, [fp, #-20]
0x00008230	e5 1b 30 10	ldr r3, [fp, #-16]
0x00008234	e1 a0 10 03	mov r1, r3
0x00008238	e1 a0 00 03	mov r0, r3
0x0000823c	eb 00 00 35	bl 8318 <__addsf3>
0x00008240	e1 a0 30 00	mov r3, r0
0x00008244	e3 c3 31 02	bic r3, r3, #-2147483648
0x00008248	e5 0b 30 08	str r3, [fp, #-8]
0x0000824c	e5 1b 10 14	ldr r1, [fp, #-20]
0x00008250	e5 1b 00 08	ldr r0, [fp, #-8]
0x00008254	eb 00 01 3d	bl 8750 <__aeabi_fdiv>
0x00008258	e1 a0 30 00	mov r3, r0
0x0000825c	e1 a0 00 03	mov r0, r3
0x00008260	eb 00 01 d7	bl 89c4 <__aeabi_f2iz>
0x00008264	e1 a0 30 00	mov r3, r0
0x00008268	e1 a0 00 03	mov r0, r3
0x0000826c	e2 4b d0 04	sub sp, fp, #4
0x00008270	e8 bd 48 00	pop {fp, lr}
0x00008274	e1 2f ff 1e	bx lr

Figure 7: The GUI uses both *eu-readelf* and *objdump* to display the traceability of *function_a* (all its instructions trace back to source code).

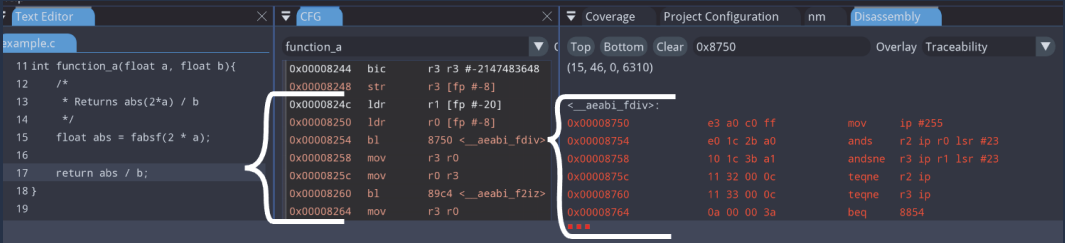


Figure 8: A divide operation calls two *aeabi* functions, which do not trace back to source code

Verification reports including:

- Activities Summary
- Object-To-Source Code Traceability
- Added Functions
- Structural Coverage of Object Code

In Markdown and HTML format

Additional CSV files with complete list of evidence and rationales

	A	B	C	D	E	
1	name	address	size	defined in	orig	hash
2	function_a	0x0000821c	92	/home/bmah/SDD		b73b8b
3	main	0x00008278	148	/home/bmah/SDD		1b061f
4	_addsf3	0x00008318	444			4de4dc
5	_aeabi_cfcmp	0x00008934	24			e5f1cf2
6	_aeabi_cfcmp	0x00008934	24			a30e13
7	_aeabi_cfcmp	0x00008924	40			beb2f3f
8	_aeabi_f2iz	0x000089c4	92			7a2391
9	_aeabi_fadd	0x00008318	444			c92519
10	_aeabi_fcmp	0x0000894c	24			9dff163
11	_aeabi_fcmp	0x00008994	24			2c4a90
12	_aeabi_fcmp	0x000089ac	24			f04936
13	_aeabi_fcmp	0x0000897c	24			aec7f2c
14	_aeabi_fcmp	0x00008964	24			ef136a8
15	_aeabi_fdiv	0x00008750	352			38a41a
16	_aeabi_fm	0x000085b8	408			140e0b
17	_aeabi_fsub	0x0000830c	456			9c3741
18	_aeabi_fsub	0x00008314	448			f686a58
19	_aeabi_i2f	0x000084dc	32			8e71ea
20	_aeabi_idiv	0x0000df74	0			
21	_aeabi_idiv0	0x0000e0bc	4			fc898b6
22	_aeabi_idivmod	0x0000e09c	32			c0553f5
23	_aeabi_i2f	0x0000850c	172			d4efa8f
24	_aeabi_idiv0	0x0000e0bc	4			5c0ad9
25	_aeabi_ui2f	0x000084d4	40			e3b333
26	_aeabi_ui2f	0x000084fc	188			6e109b
27	_aeabi_uidivmod	0x0000e0c0	0			
28	_ascii_mbtowc	0x0000c52c	68			9de08c
29	_ascii_wctomb	0x0000da00	48			e267c2

Outcome

- A TQL-5 GUI tool to perform and verify ECSS Category A qualification activities with a systematic approach
- Facilitate but does not eliminate human verification activities
- Available next year

End of the document.