

## Call Coverage with Testwell CTC++ and Imagix 4D

### What is Call Coverage?

While function coverage reflects if every function in a program was called during test at all, for call coverage all *possible* function calls are under consideration.

Full call coverage is achieved if every possible call was actually executed under test. For every single function, the call coverage can be measured as ratio of executed calls to possible calls.

The safety standard ISO 26262-6 recommends to measure call coverage for integration tests (see 10.4.6).

Call coverage is also known as *call pair coverage*.

### Deduction with Testwell CTC++

With test coverage analyzer Testwell CTC++, call coverage is not measured directly, but can be derived from stronger coverage measures:

If 100% condition coverage can be achieved for the whole software under test, then all function calls were executed. So full call coverage is achieved.

For this conclusion, it is only necessary to measure condition coverage if function calls appear in composite conditions like

```
if (x == 5 && foo())
```

In this example, the condition `x==5` must be evaluated as true, so that `foo` is called at this place.

If these kind of calls do not occur, 100% statement coverage is sufficient to prove call coverage.

### Detailed Analysis with Imagix 4D

With the graphical analysis tool Imagix 4D, reporting of call coverage can be further refined. For that purpose, Imagix 4D offers a plug-in to import and analyze reports from Testwell CTC++.

There is a dedicated report showing call coverage as a percentage for every function. Additionally, every possible call that was *not* executed under test is shown (column "Caller" with "Line" and "File" at the end.)

File	Callee	Caller	Func.Cov.	Call Cov.	(Line)
					Line File
myhome.c	main	Function has no callers	73%	-	(10)
regulators.c	air_condition	temperature_control	0%	0%	(42) 69 regulators.c
	close_windows	main	100%	50%	(20) 28 myhome.c
	heat	main	100%	50%	(37) 31 myhome.c
	lights	main main	50%	34%	(4) 27 myhome.c 60 myhome.c
	open_windows	main temperature_control	100%	34%	(25) 50 myhome.c 65 regulators.c
	open_windows_for		100%	100%	(30)
	temperature_control		50%	100%	(47)
sensors.c	strange_smell		100%	100%	(13)
	temperature_inside	main	100%	67%	(3) 29 myhome.c

This extended analysis is especially interesting in two cases:

- At integration test level, achieving full statement coverage or condition coverage is sometimes quite hard. With Imagix 4D and the Testwell CTC++ plug-in, this additional testing effort can be avoided by measuring call coverage directly.
- As long as call coverage is not fully achieved, the integrated report directly shows the reasons – missing function calls are displayed and can be traced back to the source code. The call coverage expressed in percent offers a quantitative impression how far the tests are away from achieving full call coverage.