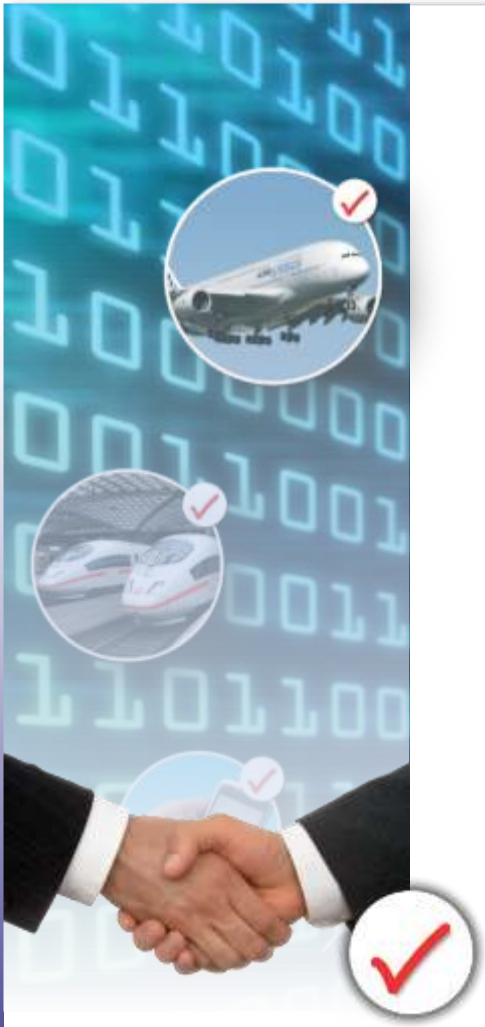
Verifysoft TECHNOLOGY

Testwell CTC++ Test Coverage Analyser for C, C++, Java and C#

Coverage on Host

On-Target Coverage for Embedded Systems

Agenda



1. Verifysoft Short Introduction
2. History of Testwell CTC++
3. Why Code Coverage?
4. Safety Standards and Code Coverage
5. Different Coverage Levels
6. Compiler Support
7. How does it work? Code Instrumentation
8. Support for Embedded Targets
9. Testwell CTC++ Packages and Qualification Kit
10. Different Reports
11. Supported Platforms/IDE and Tool Integrations
12. Live Demo

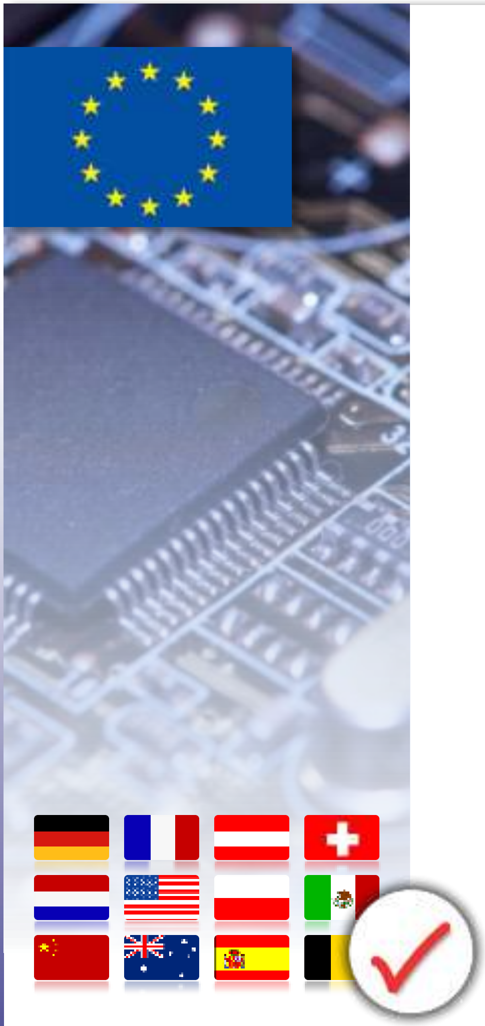
1. Verifysoft Short Introduction

Technologiepark Offenburg
In der Spoeck 10-12
77656 Offenburg
Germany

- ✓ Phone: +49 781 127 8118-0 (Germany)
- ✓ Phone: +33 3 68 33 58 84 (France)
- ✓ Fax: +49 781 63 920-29
- ✓ Email: info@verifysoft.com

www.verifysoft.com

2. History of Testwell CTC++



1989 Start of CTC++ development by Nokia group

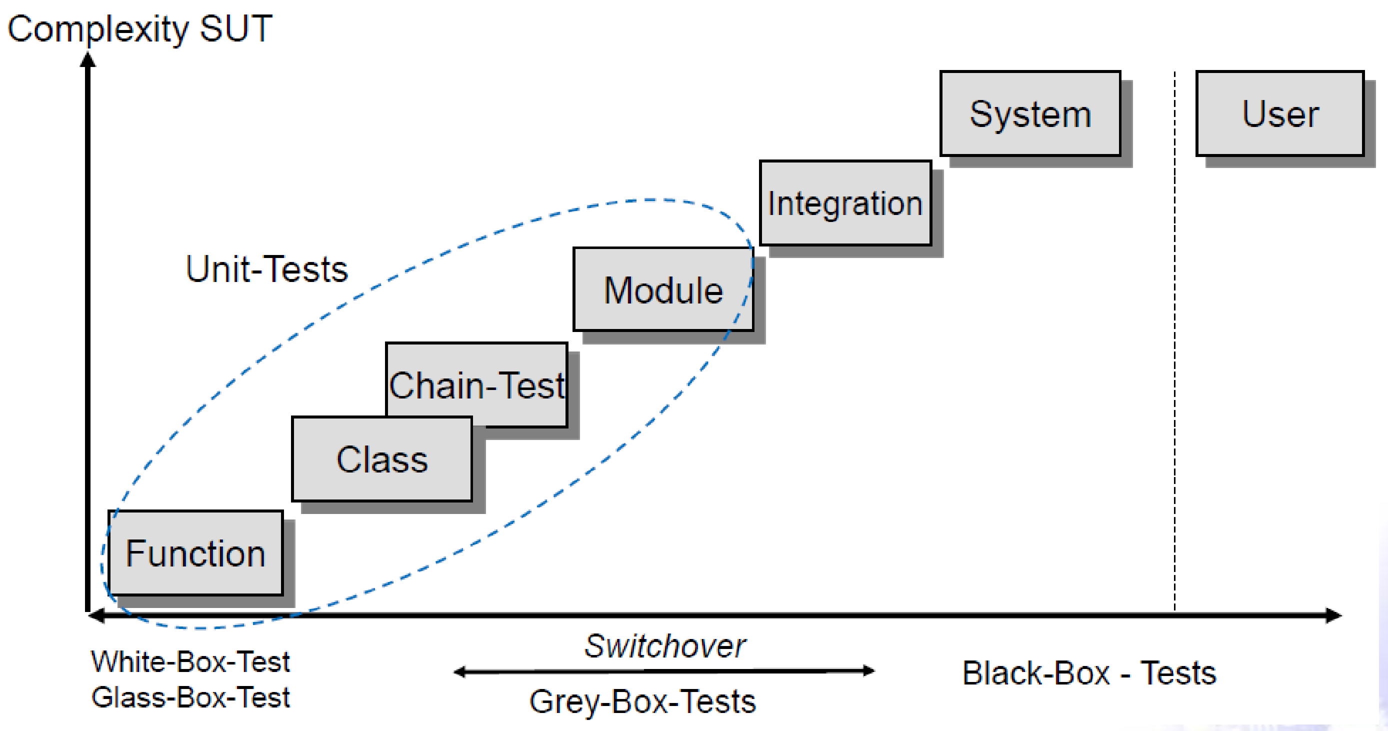
1992 Foundation of Testwell Oy, Tampere (Finland)
with the mission of further development of CTC++

2003 Foundation of Verifysoft Technology GmbH, Offenburg
as distributor for Testwell tools in Europe

2013 Verifysoft purchased Testwell tools

Several hundred CTC++ customers worldwide.
More than 1,000 licenses successfully in use.
Ongoing development.
Qualification-Kit
for DO-178C, IEC 61508, EN 50128, ISO 26262

3. Why Code Coverage?



Code Coverage ← **Test Coverage** → **Requirement Coverage**

3. Why Code Coverage?

Verifysoft TECHNOLOGY

Cause-Reason-Graph

Static Testing

Back-to-Back Testing

Classification Tree Method (CTM)

CRUD

Realtime Testing

Rare Event Testing

Load Tests

Random Testing

Recovery Tests
Stress Tests

Monkeytest

Fuzzing (Fuzz Testing)

Equivalent Classes
Multidimensional Equivalent Classes
Boundary Value Analysis
Critical Value Analysis
Informal Tests
Smoke Tests
Basis

Control Flow Oriented Testing

Advanced

Evolutionary Testing

Pairwise Testing

Established test technique for critical Embedded Systems
Test-End criterion (White-Box-Tests)
Necessary to fulfill requirements of safety standards.

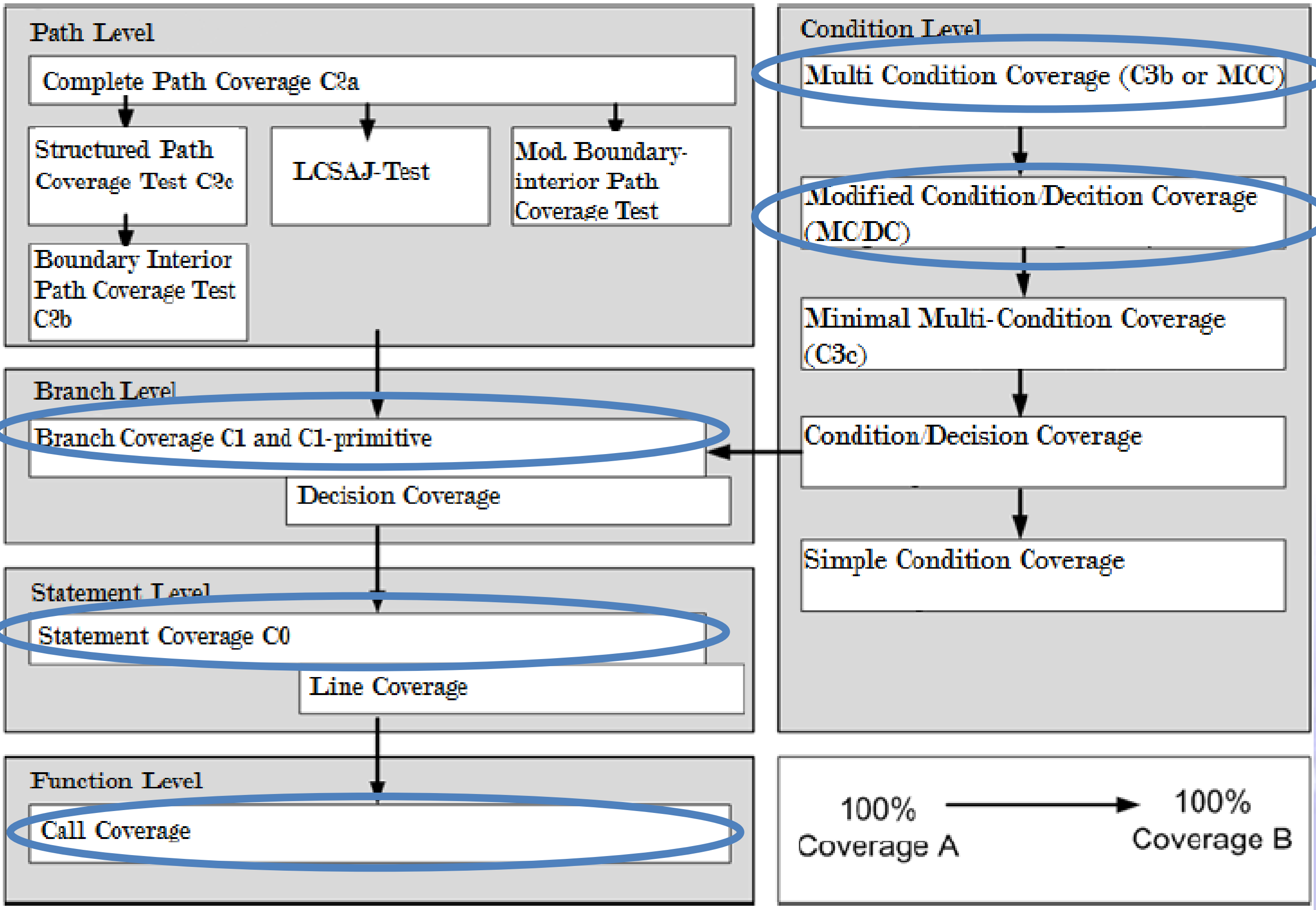
Code Coverage:
shows the parts of the code which have been
executed / not executed
tested / not tested

3. Why Code Coverage?

Why measure Code Coverage ?

- Write better (more adapted) tests
- Avoid redundant tests
- Know when you can stop testing
- Prove to your customers you have good quality
- Insure that your development partner delivers good quality
- Find Dead Code
- Required to obtain certifications
- Mandatory for safety critical development
  (standards DO-178C, IEC 61508, EN 50128, ISO 26262, ...)

4. Safety Standards and Code Coverage


Path Level
Complete Path Coverage C2a
Structured Path Coverage Test C2c
LCSAJ-Test
Mod. Boundary-interior Path Coverage Test
Boundary Interior Path Coverage Test C2b
Branch Level
Branch Coverage C1 and C1-primitive
Decision Coverage
Statement Level
Statement Coverage C0
Line Coverage
Function Level
Call Coverage
Condition Level
Multi Condition Coverage (C3b or MCC)
Modified Condition Decition Coverage (MC/DC)
Minimal Multi-Condition Coverage (C3c)
Condition Decision Coverage
Simple Condition Coverage
100% Coverage A
100% Coverage B

4. Safety Standards and Code Coverage

DIN EN 61508-3

General Industry

SIL: Safety Integrity Level

Method		SIL 1	SIL 2	SIL 3	SIL 4
...	...	...	...	...	...
7a	Function Coverage	++	++	++	++
7b	Statement Coverage	+	++	++	++
7c	Branch Coverage	+	+	++	++
7d	MC/DC	+	+	+	++

Table B.2 from DIN EN 61508-3

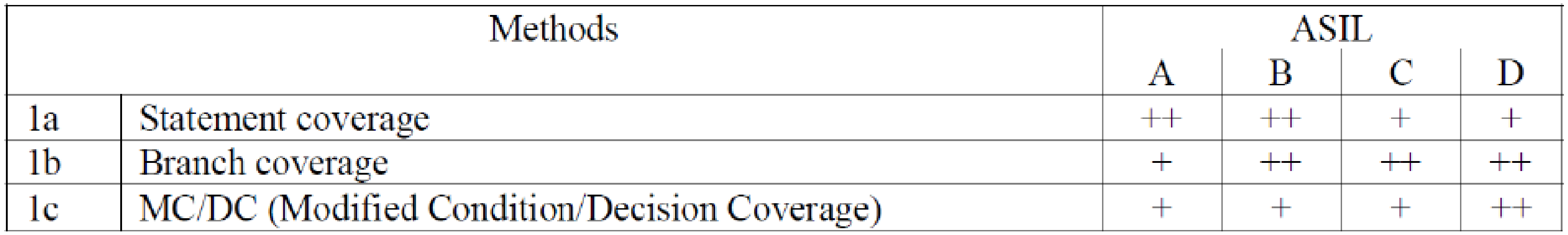
++	Highly recommended
+	Recommended

Verifysoft TECHNOLOGY

ISO 26262-6

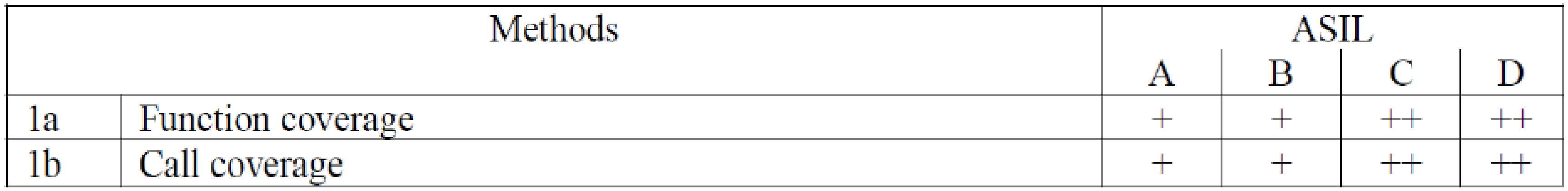
Automotive

ASIL: Automotive Safety Integrity Level



Methods		ASIL			
		A	B	C	D
1a	Statement coverage	++	++	+	+
1b	Branch coverage	+	++	++	++
1c	MC/DC (Modified Condition/Decision Coverage)	+	+	+	++

Table 12 (Software Unit Level), ISO 26262-6



Methods		ASIL			
		A	B	C	D
1a	Function coverage	+	+	++	++
1b	Call coverage	+	+	++	++

Table15 (Software Architectural Level), ISO 26262-6

++	Highly recommended
+	Recommended

4. Safety Standards and Code Coverage

Verifysoft TECHNOLOGY

Aerospace

DO-178B/C

Level	Impact	Coverage Level	% of Systems	% of Software
A	Catastrophic	MC/DC, C1, C0	20-30%	40%
B	Hazardous/Severe	C1, C0	20%	30%
C	Major	C0	25%	20%
D	Minor	-	20%	10%
E	No Effect	-	10%	5%

Statement Coverage C_0, Branch Coverage C_1, Modified Condition/ Decision Coverage MC/DC

Medical Systems

IEC 62304

„… it might be **desirable** to use white box methods to more efficiently accomplish certain tests, initiate stress conditions or faults, or increase code coverage of the qualification tests.“ (IEC 62304, Chapter B.5.7 Software System testing)

Code Coverage Requirements: CENELEC EN 50128

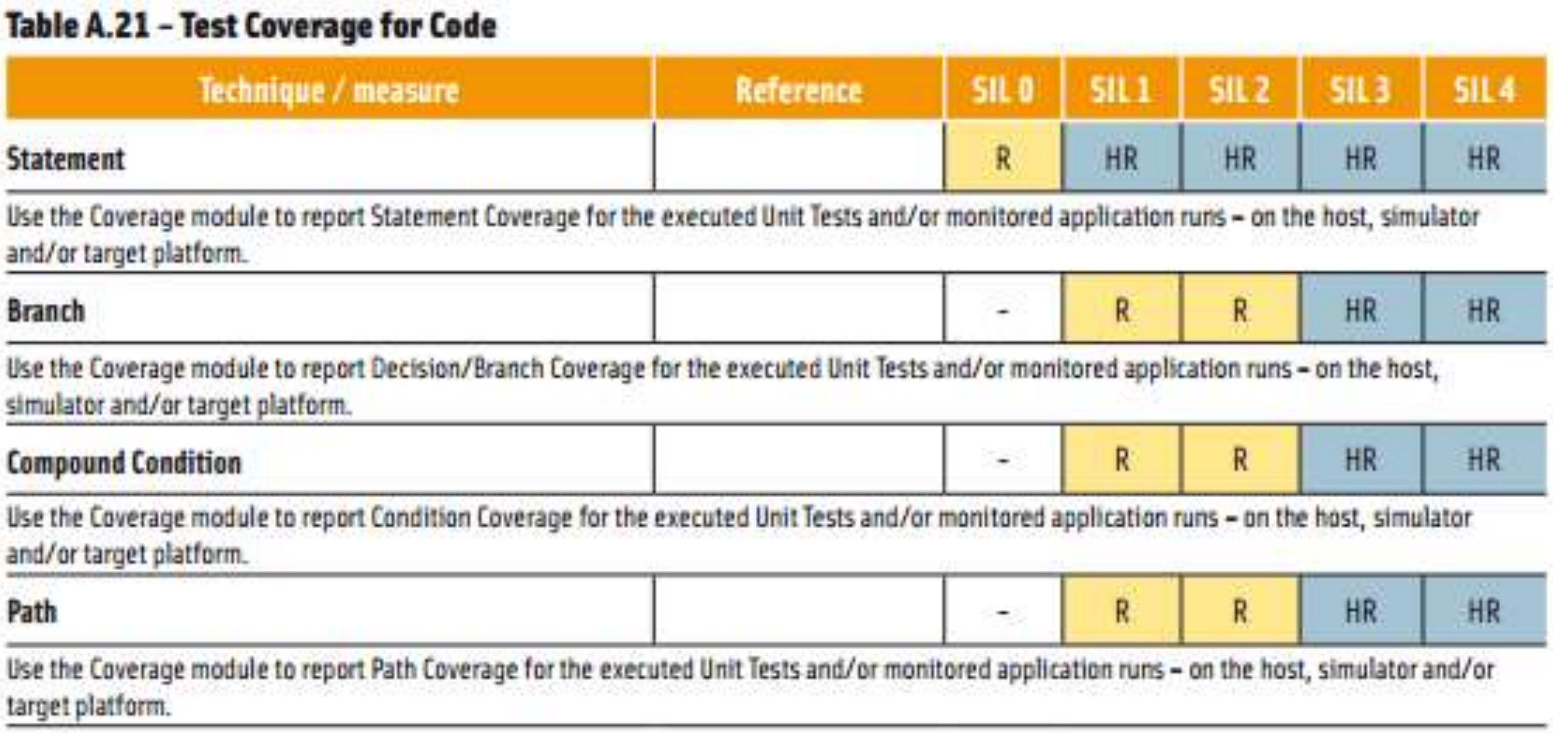


Table A.21 – Test Coverage for Code

Technique / measure	Reference	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
Statement		R	HR	HR	HR	HR
Use the Coverage module to report Statement Coverage for the executed Unit Tests and/or monitored application runs – on the host, simulator and/or target platform.						
Branch		-	R	R	HR	HR
Use the Coverage module to report Decision/Branch Coverage for the executed Unit Tests and/or monitored application runs – on the host, simulator and/or target platform.						
Compound Condition		-	R	R	HR	HR
Use the Coverage module to report Condition Coverage for the executed Unit Tests and/or monitored application runs – on the host, simulator and/or target platform.						
Path		-	R	R	HR	HR
Use the Coverage module to report Path Coverage for the executed Unit Tests and/or monitored application runs – on the host, simulator and/or target platform.						

5. Different Coverage Levels

Testwell CTC++ supports **all** required **coverage levels**:

- Function Coverage
- Decision Coverage
- Statement Coverage
- Condition Coverage
- Modified Condition/Decision Coverage (MC/DC)

- Multicondition Coverage (MCC)

works together with all unit-test tools

6. Compiler Support

Testwell CTC++ works with **all compilers**

Support is available for (as of March 2014, for actual list refer to www.verifysoft.com/en_compilers.html):

Altium Tasking classic toolsets, VX-toolset toolsets, c166, cc166, ccm16c, cc51
ARM DS-5, Keil MDK-ARM
Borland/Inprise/Paradigm/Codegear bcc, bcc32, pcc, pcc32 (Paradigm)
Ceva DSP all (just use gcc settings)
Cosmic cx6805, cx6808, cx6812, cxs12x, cxs12z, cxxgate, cx6811, cx6816, cx332, cxst10, cxstm8, cxst7, cxcf, cx56K, cxppc
Freescale/Metrowerks mwccmcf, mwcceppc, mwccmcore, mwcc56800, mwcc56800e, chc12, chc08
Fujitsu/Softune fcc907s, fcc911s, fcc896s
gcc and all gcc based cross-compilers i586-mingw32msvc-gcc, x86_64-linux-gnu-gcc, m68k-palmos-coff-gcc, tricore-gcc, arm-linux-gnueabi-gcc, arm-none-eabi-gcc, arm-none-linux-gnueabi-gcc, arm-elf-gcc, arm-montavista-linux-gnueabi-gcc, pic30-gcc, pic32-gcc, avr-gcc, xc16-gcc, mlx16-gcc, thumb-epoc-pe-gcc, arm4-epoc-pe-gcc, armv-epoc-pe-gcc, powerpc-wrs-linux-gnu-e500v2-glibc_small-gcc, *-gcc, *-*-gcc, *-*-*-gcc
GHS/GreenHills/Multi ccv850, cxv850, ccmips, cxmips, ccarm, cxarm, ccthumb, cxthumb, ccppc, cxppc,gcc (GreenHill, not GNU)
Hitatchi shc, shcpp, ch38, cxrx
HI-Tech PICC (Windows and Linux) picc, picc18, picc32, dspicc, xc16-gcc, xc32-gcc,
HP HPUX CC, HP C++, aCC
IAR compilers and toolchains icc430, icc78k, icc78k0r, icc8051, iccarm, iccavr, iccavr32, icccf, icchcs12, iccdspic, iccmaxq, iccpic18, icccr16c, iccv850, icch8, iccm8k, iccm32c, iccm16c, iccr32c, iccrl78, iccrx, iccsam8, iccstm8
Intel (all platforms) icc, ic86, ic96
Java compilers Javac, jikes, ecj, gcj, kaffe
Keil c51, c166, c251, ca/ cx51, cx2, tcc / armcc
LLVM clang, clang++ / Matlab/Simulink / lcc
Metaware (both Windows and Linux host) hcac, hcarc, hcarm
Microchip MPLAB C pic30-gcc, pic32-gcc

--->

6. Compiler Support

Testwell CTC++ works with all compilers (continued)

→

Microsoft compilers cl on host, both 32 and 64 bit / cl for Smartphones and PocketPC / csc C# compiler / vjc J# compiler
Mono compilers dmcs, mcs, gmcs, smcs
Motorola chc12, chc08
Pathscale pathcc/pathCC
Renesas shc, shcpp, ch38, ccrx
Raisonance rc51, rcmp
Sun WorkShop compilers, javac
Symbian various compilers
TI Code Composer Studio (Windows) cl2000, cl16x, cl470, cl55, cl500, cl430
Texas Instruments Linux compilers cl2000, cl16x, cl470, cl55, cl500, cl430
Trimedia tmcc
VisualDSP++ ccblkfn, cc21k, ccts
Windriver ccarm, ccsimpc, g++simpc, g++arm, cchppa, ccsimso, ccsparc, cc68k, cc386, cc960, ccmips, ccppc

You have not seen your compiler? Contact us!
We will adapt Testwell CTC++ to your compiler within a few days and without any cost (adaptation can even be done by the customer).

Testwell CTC++ supports **all** compilers!
No unsupported compilers!

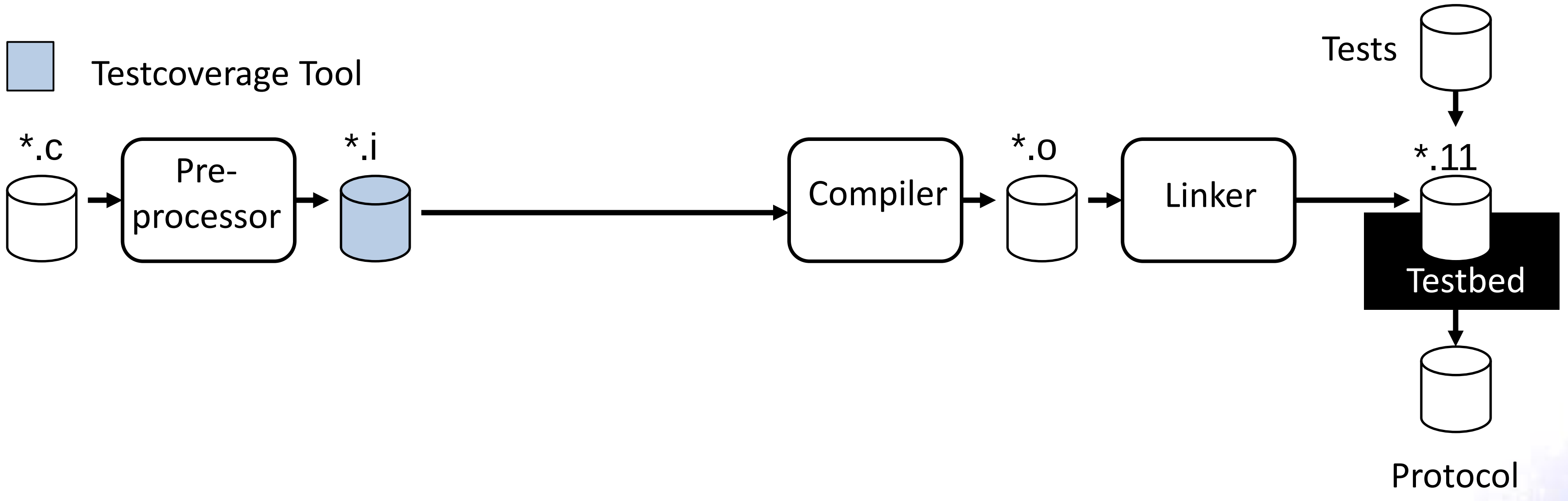
7. Code Instrumentation

Instrumentation

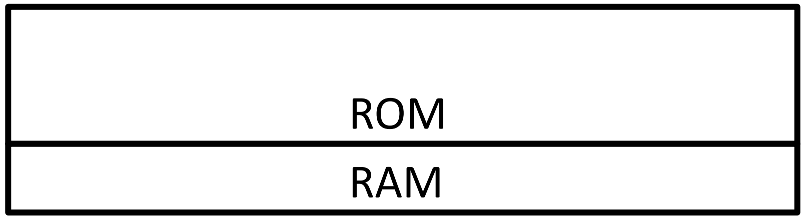
- Adding of global counters (Integer-Arrays) into the source code
- Storage of information about counter instrumentation
- Increment counters with each run
- Storage of counter values
- Analysis of the counter values for reporting

7. Code Instrumentation

Tool-Chain

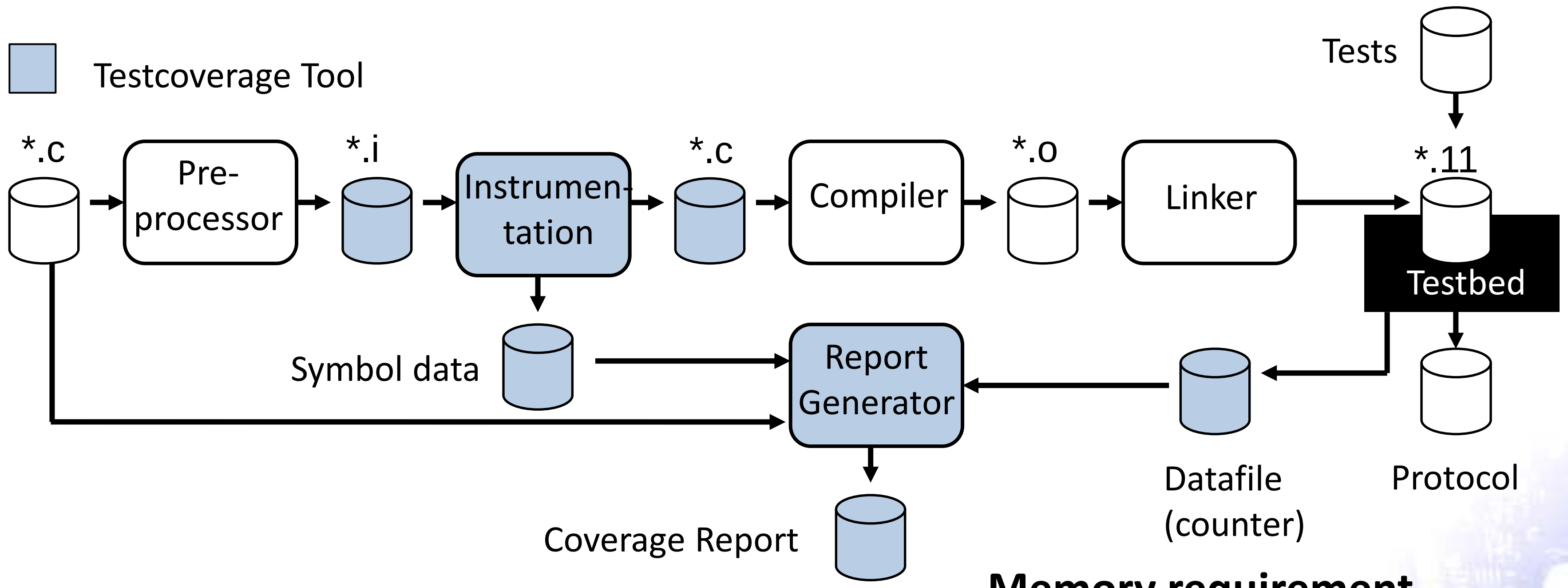


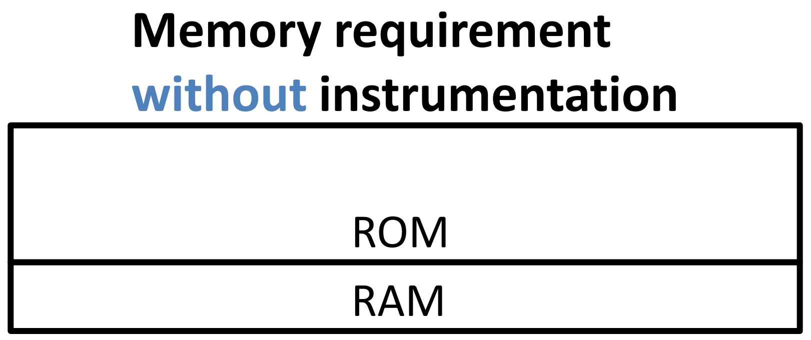
Memory requirement
without instrumentation

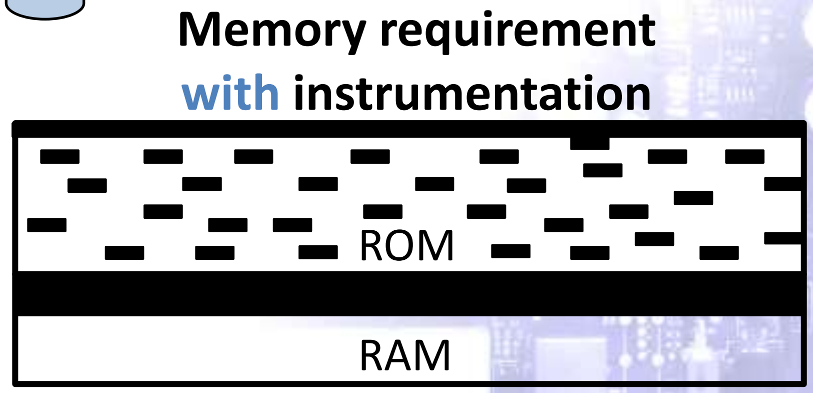


7. Code Instrumentation

Tool-Chain





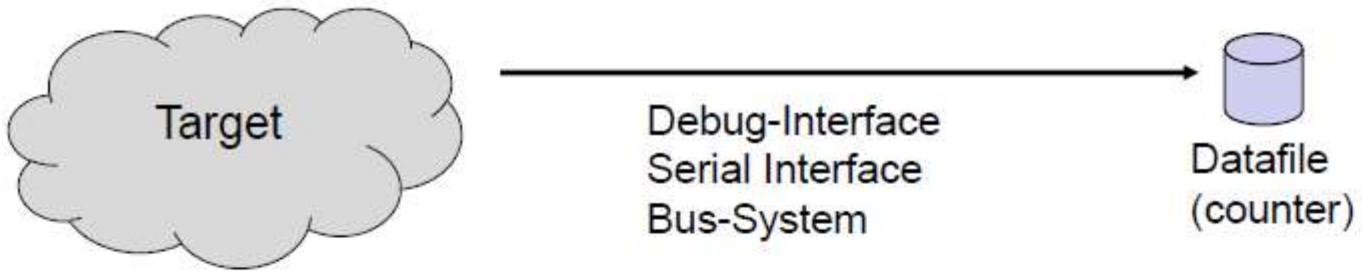


7. Code Instrumentation

- **RAM**
- **ROM**

$\left.\begin{array}{l}\textbf{RAM}\\\textbf{ROM}\end{array}\right\}$ Reason for lack of memory: **80 % RAM, 20 % ROM** (pract. experience)

- Mostly no filesystem (so counters have to be stored in memory)



- Limited amount of interfaces on the target device (transfer of datafile)
  Consider additional testing interfaces in the hardware design
  (design for test)

7. Code Instrumentation

```
int goo( int a, int b, int c)
{
   int x;

   if (((a>0) || (b>0)) && (c>0))
   {
      x = 1;
   }
   else
   {
      x = 0;
   }

   return x;
}
```

ROM-Usage

Without instrumentation:	60 Byte
Function Coverage:	67 Byte
Branch Coverage:	118 Byte
Condition Coverage:	285 Byte

Simple example with small code and big instrumentation overhead (mean 30 % of code size).

Additional RAM-Usage without Bit-Coverage

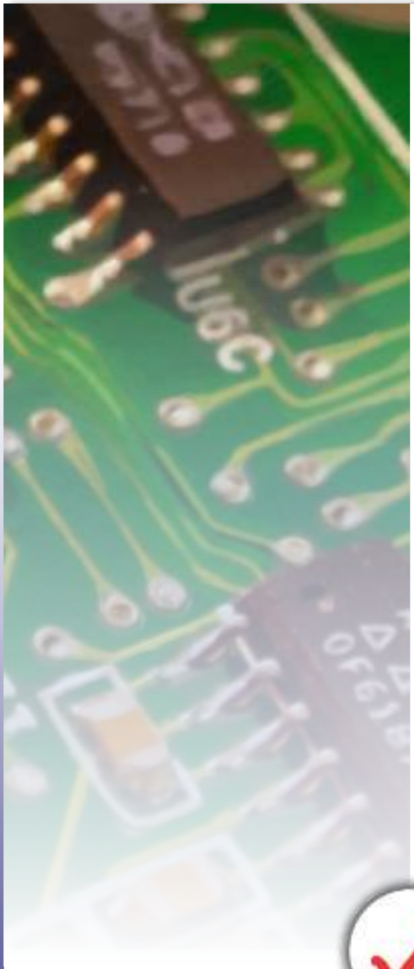
Function Coverage:	1 Integer
Branch Coverage:	4 Integer
Condition Coverage:	7 Integer

Integer:
32 Bit (unsigned long) as default

Additional RAM-Usage using Bit-Coverage

Function Coverage:	1 Bit
Branch Coverage:	4 Bit
Condition Coverage:	7 Bit

8. Support for Embedded Targets and native projects



Testwell CTC++ is **the ideal tool for embedded targets**

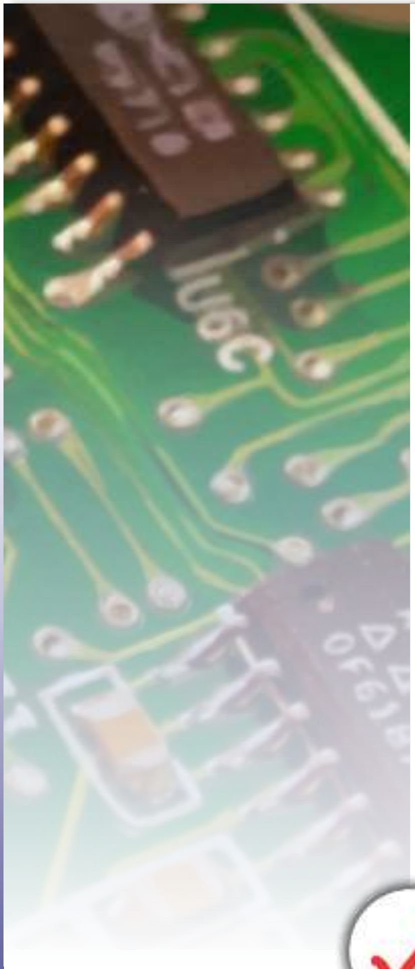
- Dramatically easy to use !
- Very low instrumentation overhead on your C files
- Works with all targets
  - Host-Target add-on is provided as source code and so can be easily adapted to new targets
- … even with smallest targets and microcontrollers
- Supports all compilers/cross-compilers

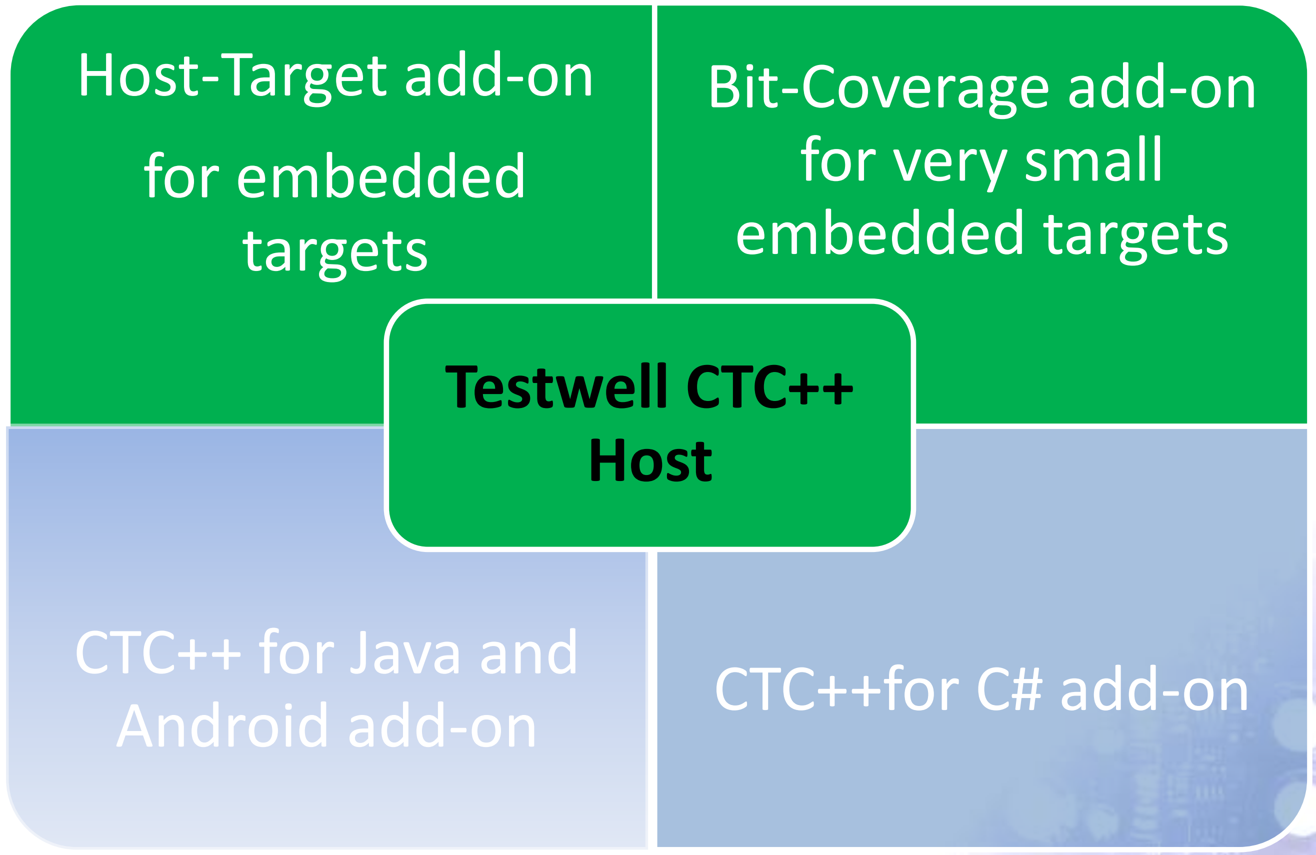
8. Support for Embedded Targets and native projects

Testwell CTC++ is **the ideal tool for native projects**

- Setup and usage are straightforward
- Java and C# support on top of C & C++
- Very fast analysis
- Interfacing with MS Visual Studio IDE
- …even on large projects

9. Testwell CTC++ Packages and Qualification Kit




Host-Target add-on for embedded targets
Bit-Coverage add-on for very small embedded targets
Testwell CTC++ Host
CTC++ for Java and Android add-on
CTC++for C# add-on

You only need one code coverage tool for C, C++,Java and C#
One license covers all embedded targets and all compilers

9. Testwell CTC++ Packages and Qualification Kit

Compliance with Standards
DO-178C - IEC 61508 - IEC 62304 - ISO 26262

Testwell CTC++ can be used to obtain certification in automotive, railway, avionics and medical industries

Tool-Qualification Kits available

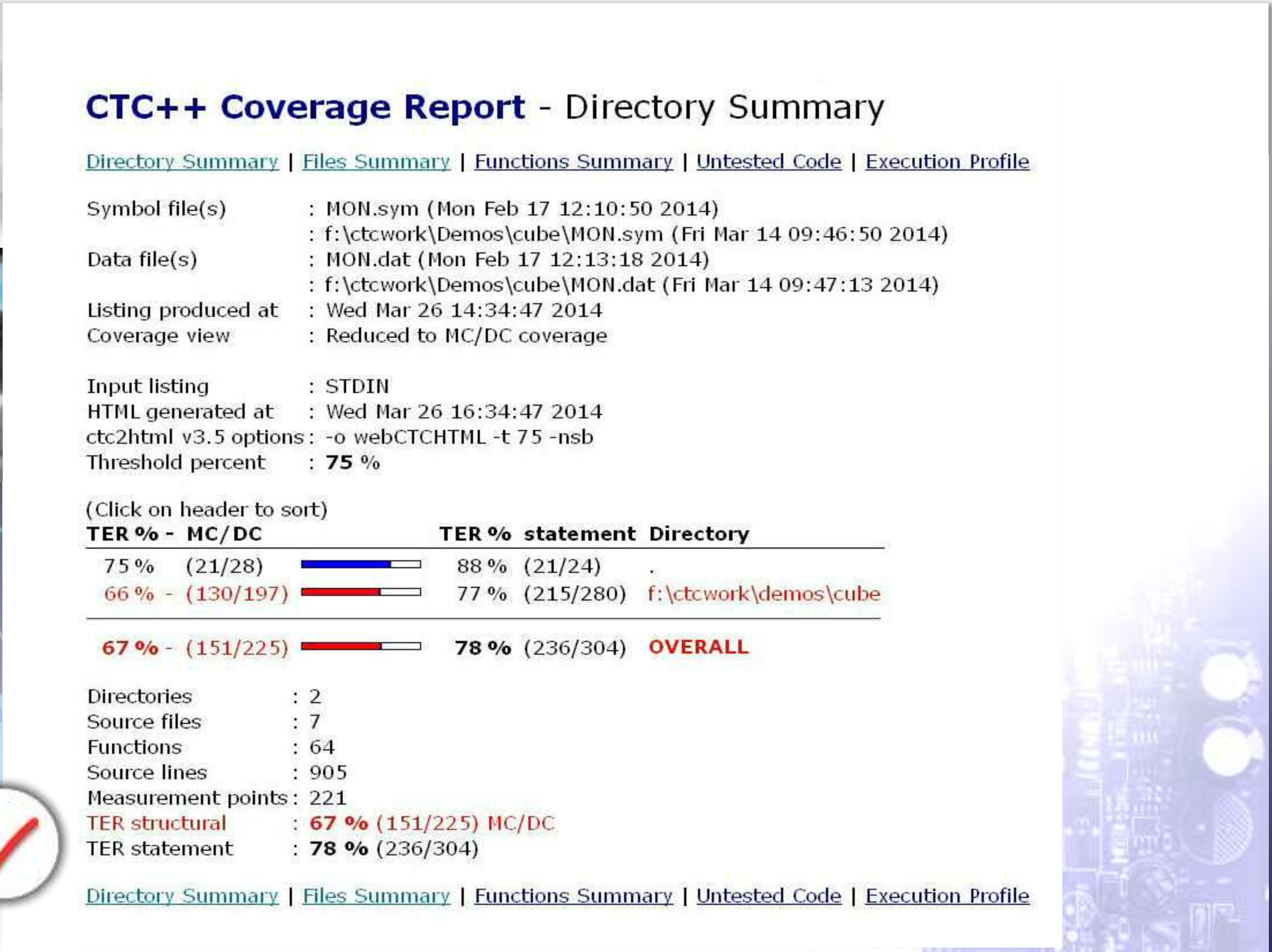
10. Different Reports

Reports in text, XML,HTML

Directory Summary
Files Summary
Functions Summary

Execution Profile
Untested Code Listing
Execution Time Listing

10. Different Reports



CTC++ Coverage Report - Directory Summary

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile

```
Symbol file(s)          : MON.sym (Mon Feb 17 12:10:50 2014)
                        : f:\ctcwork\Demos\cube\MON.sym (Fri Mar 14 09:46:50 2014)
Data file(s)            : MON.dat (Mon Feb 17 12:13:18 2014)
                        : f:\ctcwork\Demos\cube\MON.dat (Fri Mar 14 09:47:13 2014)
Listing produced at     : Wed Mar 26 14:34:47 2014
Coverage view           : Reduced to MC/DC coverage

Input listing           : STDIN
HTML generated at       : Wed Mar 26 16:34:47 2014
ctc2html v3.5 options   : -o webCTCHTML -t 75 -nsb
Threshold percent       : 75 %
```

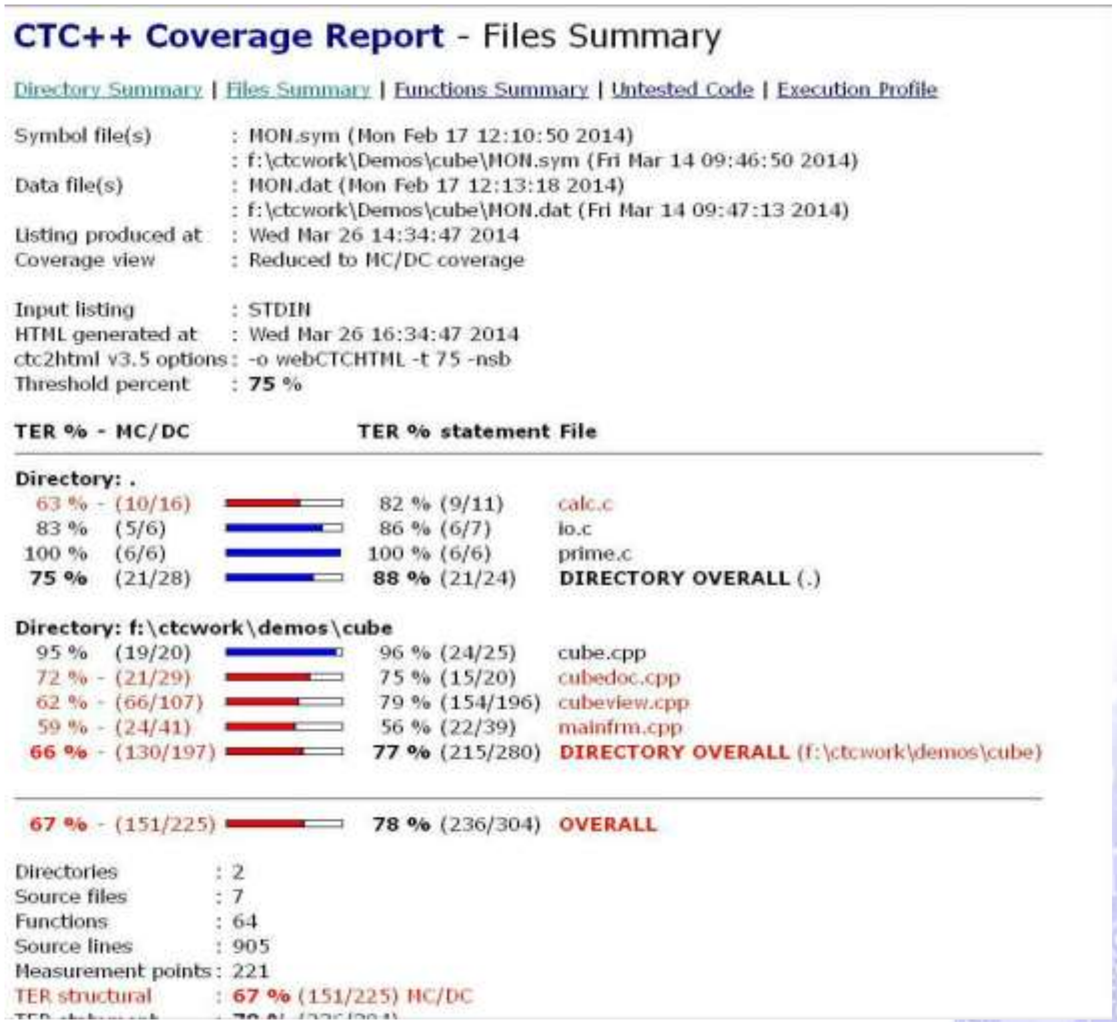
(Click on header to sort)

TER % - MC/DC			TER % statement		Directory
75 %	(21/28)		88 %	(21/24)	.
66 % -	(130/197)		77 %	(215/280)	f:\ctcwork\demos\cube
67 % -	(151/225)		**78 %**	(236/304)	**OVERALL**

```
Directories        : 2
Source files       : 7
Functions          : 64
Source lines       : 905
Measurement points : 221
TER structural     : 67 % (151/225) MC/DC
TER statement      : 78 % (236/304)
```

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile

10. Different Reports



CTC++ Coverage Report - Files Summary

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile

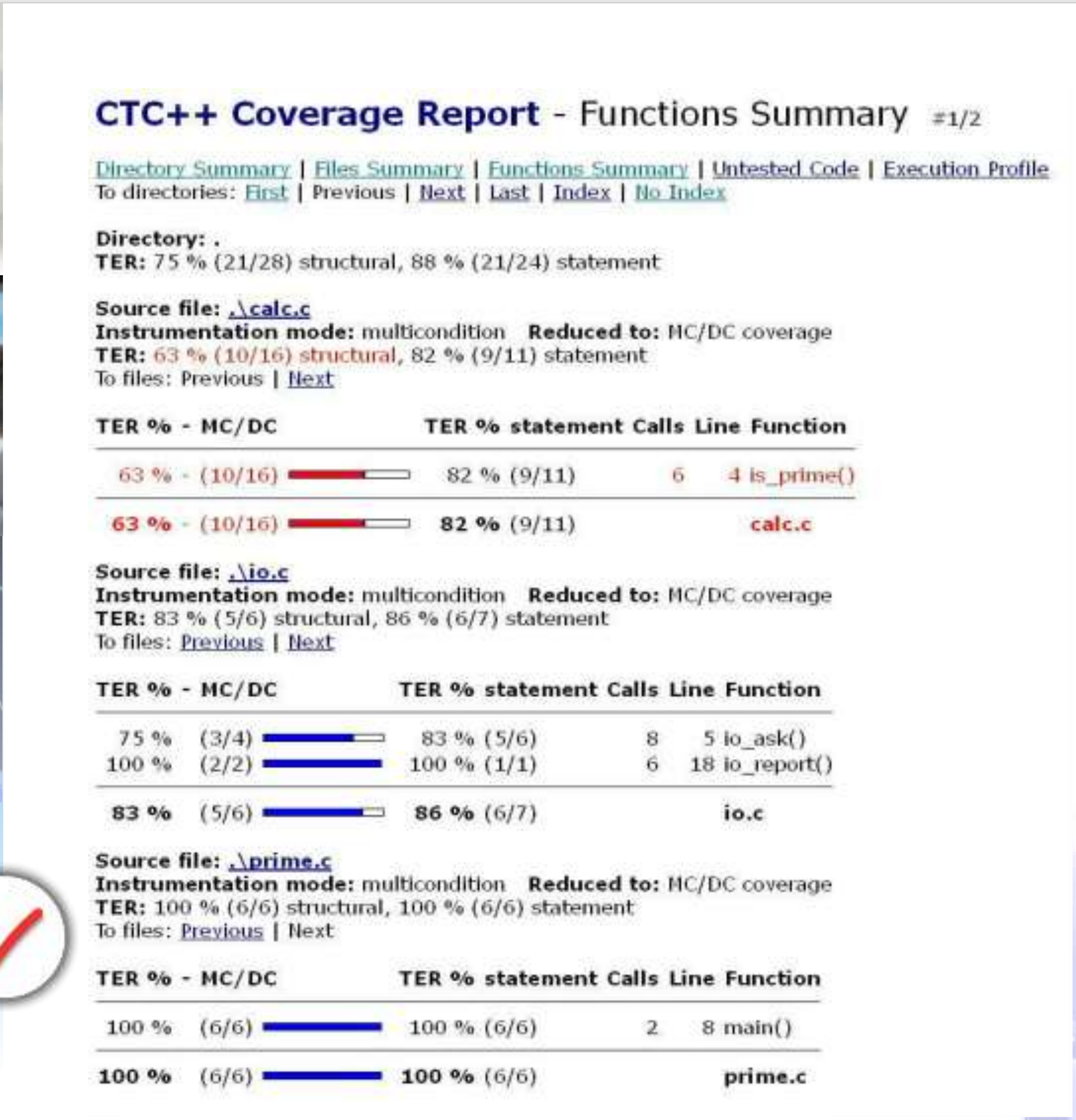
Symbol file(s)	: MON.sym (Mon Feb 17 12:10:50 2014)
	: f:\ctcwork\Demos\cube\MON.sym (Fri Mar 14 09:46:50 2014)
Data file(s)	: MON.dat (Mon Feb 17 12:13:18 2014)
	: f:\ctcwork\Demos\cube\MON.dat (Fri Mar 14 09:47:13 2014)
Listing produced at	: Wed Mar 26 14:34:47 2014
Coverage view	: Reduced to MC/DC coverage

Input listing	: STDIN
HTML generated at	: Wed Mar 26 16:34:47 2014
ctc2html v3.5 options	: -o webCTCHTML -t 75 -nsb
Threshold percent	: **75 %**

TER % - MC/DC	TER % statement	File
Directory: .		
63 % - (10/16)	82 % (9/11)	calc.c
83 % (5/6)	86 % (6/7)	io.c
100 % (6/6)	100 % (6/6)	prime.c
75 % (21/28)	**88 %** (21/24)	**DIRECTORY OVERALL** (.)
Directory: f:\ctcwork\demos\cube		
95 % (19/20)	96 % (24/25)	cube.cpp
72 % - (21/29)	75 % (15/20)	cubedoc.cpp
62 % - (66/107)	79 % (154/196)	cubeview.cpp
59 % - (24/41)	56 % (22/39)	mainfrm.cpp
66 % - (130/197)	**77 %** (215/280)	**DIRECTORY OVERALL** (f:\ctcwork\demos\cube)
67 % - (151/225)	**78 %** (236/304)	**OVERALL**

Directories	: 2
Source files	: 7
Functions	: 64
Source lines	: 905
Measurement points	: 221
TER structural	: **67 %** (151/225) MC/DC

10. Different Reports



CTC++ Coverage Report - Functions Summary #1/2

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile
To directories: First | Previous | Next | Last | Index | No Index

Directory: .
TER: 75 % (21/28) structural, 88 % (21/24) statement

Source file: .\calc.c
Instrumentation mode: multicondition **Reduced to:** MC/DC coverage
TER: 63 % (10/16) structural, 82 % (9/11) statement
To files: Previous | Next

TER % - MC/DC		TER % statement		Calls	Line	Function
63 %	(10/16)	82 %	(9/11)	6	4	is_prime()
63 %	(10/16)	**82 %**	(9/11)			**calc.c**

Source file: .\io.c
Instrumentation mode: multicondition **Reduced to:** MC/DC coverage
TER: 83 % (5/6) structural, 86 % (6/7) statement
To files: Previous | Next

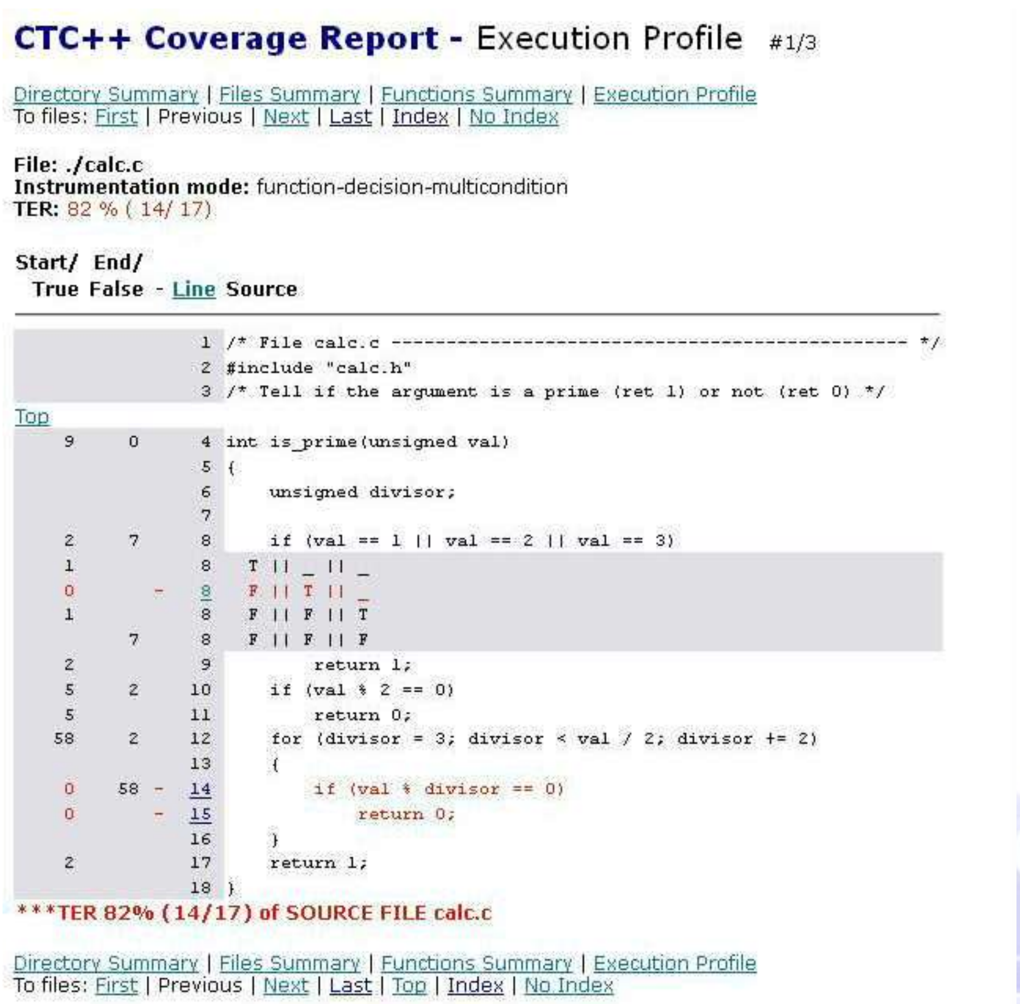
TER % - MC/DC		TER % statement		Calls	Line	Function
75 %	(3/4)	83 %	(5/6)	8	5	io_ask()
100 %	(2/2)	100 %	(1/1)	6	18	io_report()
83 %	(5/6)	**86 %**	(6/7)			**io.c**

Source file: .\prime.c
Instrumentation mode: multicondition **Reduced to:** MC/DC coverage
TER: 100 % (6/6) structural, 100 % (6/6) statement
To files: Previous | Next

TER % - MC/DC		TER % statement		Calls	Line	Function
100 %	(6/6)	100 %	(6/6)	2	8	main()
100 %	(6/6)	**100 %**	(6/6)			**prime.c**

10. Different Reports

Verifysoft TECHNOLOGY



CTC++ Coverage Report - Execution Profile #1/3

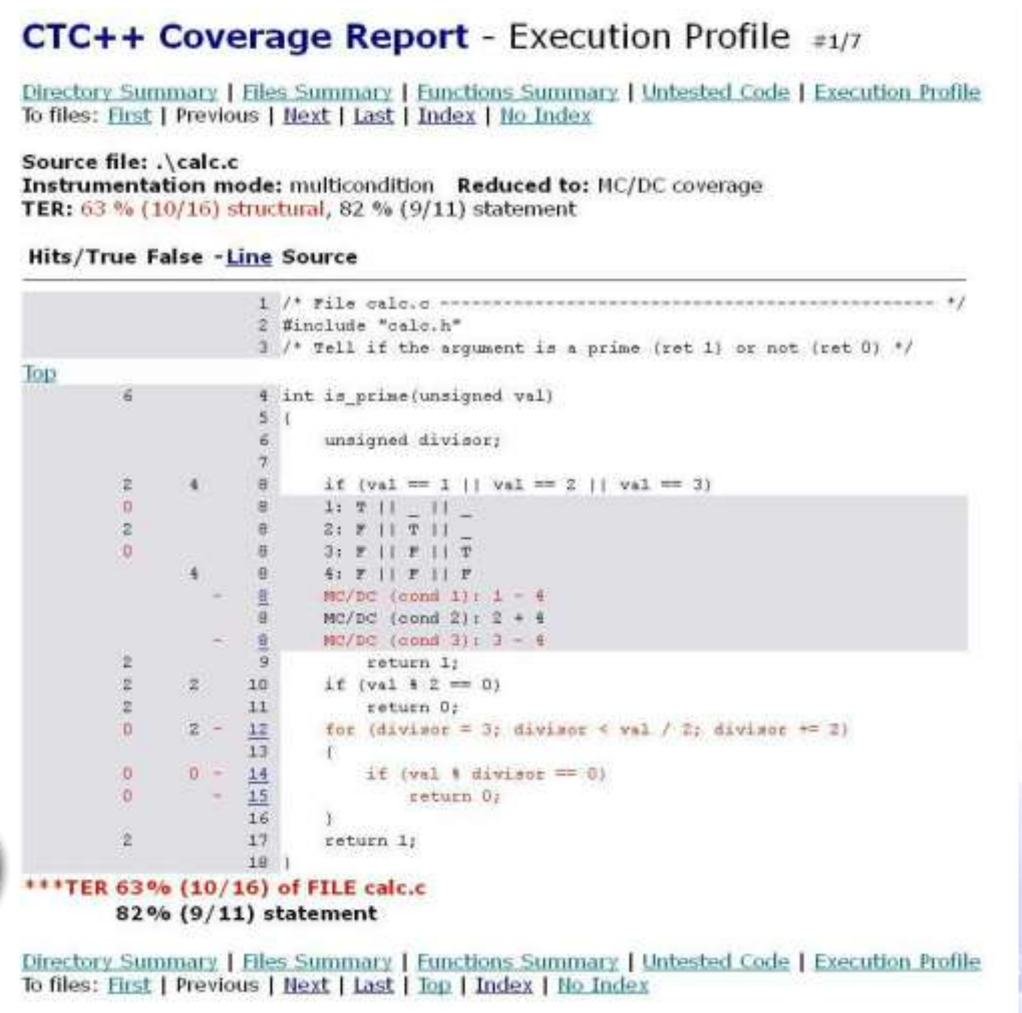
Directory Summary | Files Summary | Functions Summary | Execution Profile
To files: First | Previous | Next | Last | Index | No Index

File: ./calc.c
Instrumentation mode: function-decision-multicondition
TER: 82 % (14/ 17)

```
Start/  End/
  True False  -  Line Source
------------------------------------------------------------------------------
                    1 /* File calc.c ---------------------------------------------- */
                    2 #include "calc.h"
                    3 /* Tell if the argument is a prime (ret 1) or not (ret 0) */
Top
     9      0       4 int is_prime(unsigned val)
                    5 {
                    6     unsigned divisor;
                    7
     2      7       8     if (val == 1 || val == 2 || val == 3)
     1              8   T || _ || _
     0          -   8   F || T || _
     1              8   F || F || T
            7       8   F || F || F
     2              9         return 1;
     5      2      10     if (val % 2 == 0)
     5             11         return 0;
    58      2      12     for (divisor = 3; divisor < val / 2; divisor += 2)
                   13     {
     0     58   -  14         if (val % divisor == 0)
     0          -  15             return 0;
                   16     }
     2             17     return 1;
                   18 }
***TER 82% (14/17) of SOURCE FILE calc.c
```

Directory Summary | Files Summary | Functions Summary | Execution Profile
To files: First | Previous | Next | Last | Top | Index | No Index

10. Different Reports



CTC++ Coverage Report - Execution Profile #1/7

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile
To files: First | Previous | Next | Last | Index | No Index

Source file: .\calc.c
Instrumentation mode: multicondition **Reduced to:** MC/DC coverage
TER: 63 % (10/16) structural, 82 % (9/11) statement

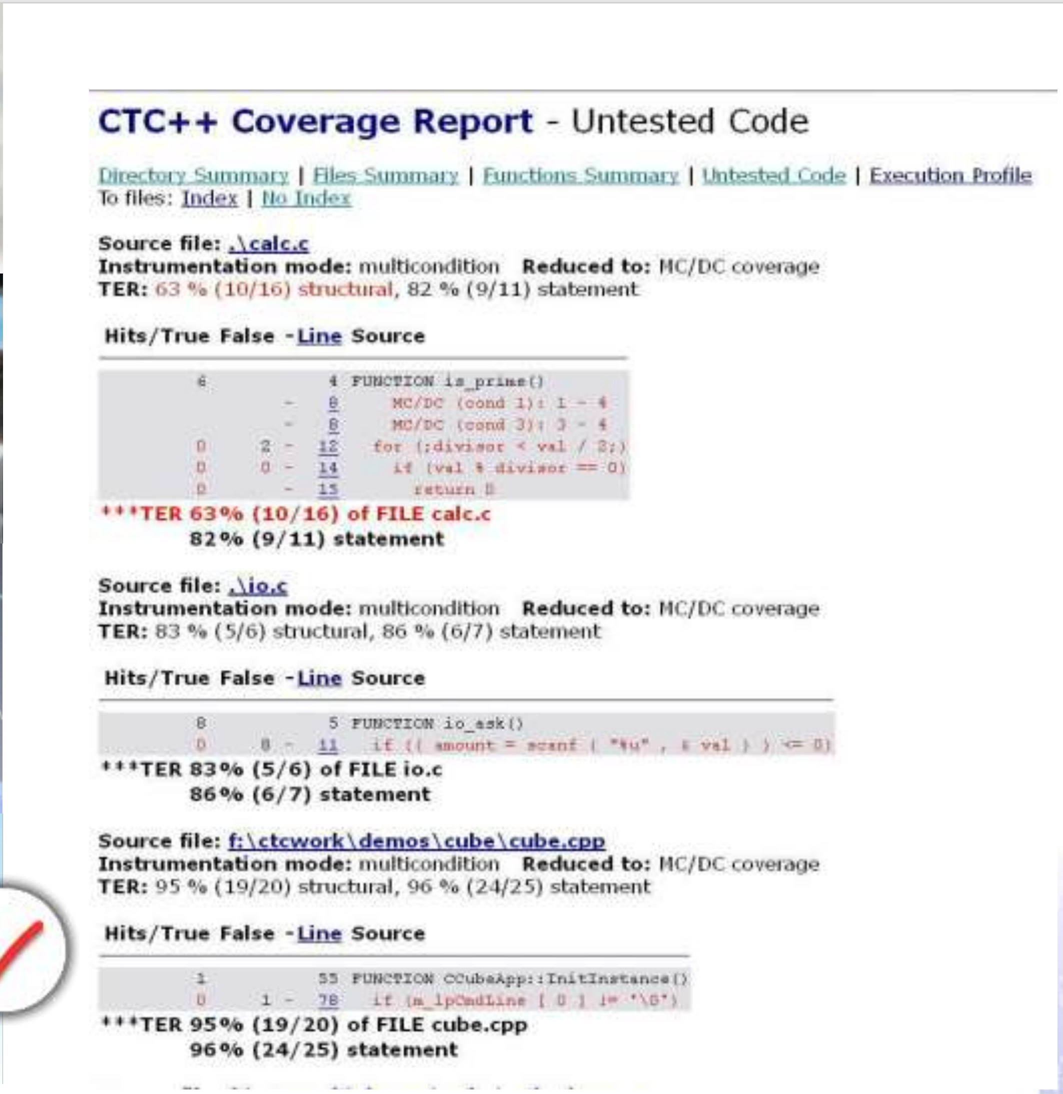
Hits/True False -Line Source

```
                    1 /* File calc.c ------------------------------------------------ */
                    2 #include "calc.h"
                    3 /* Tell if the argument is a prime (ret 1) or not (ret 0) */
Top
      6             4 int is_prime(unsigned val)
                    5 {
                    6     unsigned divisor;
                    7
      2     4       8     if (val == 1 || val == 2 || val == 3)
      0             8     1: T || _ || _
      2             8     2: F || T || _
      0             8     3: F || F || T
            4       8     4: F || F || F
              -     8     MC/DC (cond 1): 1 - 4
                    8     MC/DC (cond 2): 2 + 4
              -     8     MC/DC (cond 3): 3 - 4
      2             9         return 1;
      2     2      10     if (val % 2 == 0)
      2            11         return 0;
      0     2 -    12     for (divisor = 3; divisor < val / 2; divisor += 2)
                   13     {
      0     0 -    14         if (val % divisor == 0)
      0       -    15             return 0;
                   16     }
      2            17     return 1;
                   18 }
```

***TER 63% (10/16) of FILE calc.c**
82% (9/11) statement

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile
To files: First | Previous | Next | Last | Top | Index | No Index

10. Different Reports



CTC++ Coverage Report - Untested Code

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile
To files: Index | No Index

Source file: .\calc.c
Instrumentation mode: multicondition   **Reduced to:** MC/DC coverage
TER: 63 % (10/16) structural, 82 % (9/11) statement

```
 Hits/True False -Line Source

       6             4 FUNCTION is_prime()
                 -   8     MC/DC (cond 1): 1 - 4
                 -   8     MC/DC (cond 3): 3 - 4
       0      2  -  12   for (;divisor < val / 2;)
       0      0  -  14     if (val % divisor == 0)
       0         -  15       return 0
***TER 63% (10/16) of FILE calc.c
       82% (9/11) statement
```

Source file: .\io.c
Instrumentation mode: multicondition   **Reduced to:** MC/DC coverage
TER: 83 % (5/6) structural, 86 % (6/7) statement

```
 Hits/True False -Line Source

       8             5 FUNCTION io_ask()
       0      8  -  11   if (( amount = scanf ( "%u" , & val ) ) <= 0)
***TER 83% (5/6) of FILE io.c
       86% (6/7) statement
```

Source file: f:\ctcwork\demos\cube\cube.cpp
Instrumentation mode: multicondition   **Reduced to:** MC/DC coverage
TER: 95 % (19/20) structural, 96 % (24/25) statement

```
 Hits/True False -Line Source

       1            55 FUNCTION CCubeApp::InitInstance()
       0      1  -  78   if (m_lpCmdLine [ 0 ] != '\0')
***TER 95% (19/20) of FILE cube.cpp
       96% (24/25) statement
```

11. Supported Platforms/IDE and Tool Integrations

Supported Platforms

Windows, Linux, FreeBSD, Solaris, HP-UX

MacOSX, AIX, others on request

11. Supported Platforms/IDE and Tool Integrations

IDE-Integrations

Visual Studio v7.0 and later
IAR (all platforms)
Borland 5.02
Fujitsu Softune
Eclipse
Others on request

Works also with:
MP-LAB, Keil, …

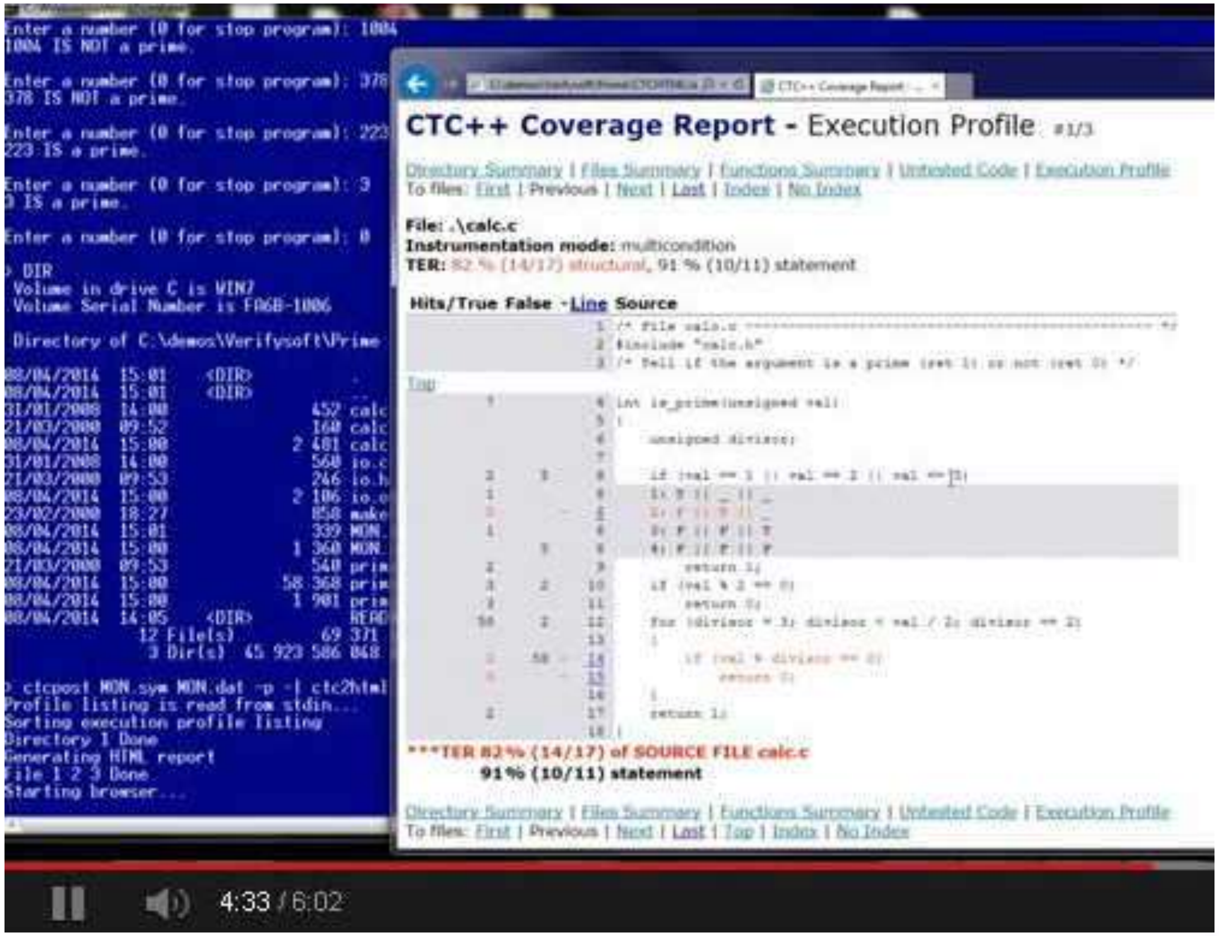
11. Supported Platforms/IDE and Tool Integrations

Integrations
with Tool-Chains and Testing environments
include

CATIA - AUTOSAR Builder (Dassault Systèmes)
dSpace SystemDesk
dSpace TargetLink
PikeTech TPT
QTronic TestWeaver / Silver
Imagix 4D
SonarQube
… (ask for other integrations)

Further information: www.verifysoft.com

12. Live Demo



```
Enter a number (0 for stop program): 1004
1004 IS NOT a prime.

Enter a number (0 for stop program): 378
378 IS NOT a prime.

Enter a number (0 for stop program): 223
223 IS a prime.

Enter a number (0 for stop program): 3
3 IS a prime.

Enter a number (0 for stop program): 0

> DIR
 Volume in drive C is WIN7
 Volume Serial Number is FA6B-1006

 Directory of C:\demos\Verifysoft\Prime

08/04/2014  15:01    <DIR>          .
08/04/2014  15:01    <DIR>          ..
31/01/2008  14:00               452 calc
21/03/2000  09:52               160 calc
08/04/2014  15:00             2 481 calc
31/01/2008  14:00               560 io.c
21/03/2000  09:53               246 io.h
08/04/2014  15:00             2 106 io.o
23/02/2000  18:27               858 make
08/04/2014  15:01               339 MON.
08/04/2014  15:00             1 360 MON.
21/03/2000  09:53               540 prim
08/04/2014  15:00            58 368 prim
08/04/2014  15:00             1 901 prim
08/04/2014  14:05    <DIR>          READ
              12 File(s)         69 371
               3 Dir(s)  45 923 586 048

> ctcpost MON.sym MON.dat -p - | ctc2html
Profile listing is read from stdin...
Sorting execution profile listing
Directory 1 Done
Generating HTML report
File 1 2 3 Done
Starting browser...
```

CTC++ Coverage Report - Execution Profile #1/3

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile
To files: First | Previous | Next | Last | Index | No Index

File: .\calc.c
Instrumentation mode: multicondition
TER: 82 % (14/17) structural, 91 % (10/11) statement

```
Hits/True False -Line Source
                   1 /* File calc.c ------------------------------------------ */
                   2 #include "calc.h"
                   3 /* Tell if the argument is a prime (ret 1) or not (ret 0) */
Top
     7             4 int is_prime(unsigned val)
                   5 {
                   6     unsigned divisor;
                   7
     2     3       8     if (val == 1 || val == 2 || val == 3)
     1             8     1: T || _ || _
     0          -  8     2: F || T || _
     1             8     3: F || F || T
           3       8     4: F || F || F
     2             9         return 1;
     3     2      10     if (val % 2 == 0)
     3            11         return 0;
    58     2      12     for (divisor = 3; divisor < val / 2; divisor += 2)
                  13     {
     0    58    - 14         if (val % divisor == 0)
     0          - 15             return 0;
                  16     }
     2            17     return 1;
                  18 }
***TER 82 % (14/17) of SOURCE FILE calc.c
        91 % (10/11) statement
```

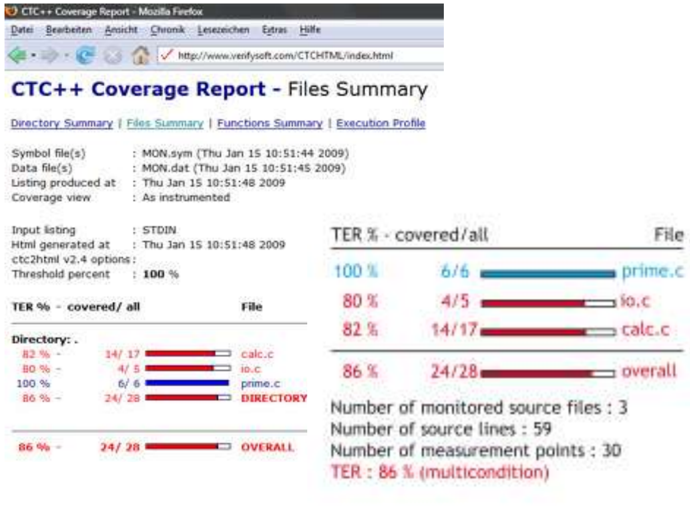
Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile
To files: First | Previous | Next | Last | Top | Index | No Index

4:33 / 6:02

For an online presentation, please visit
http://www.verifysoft.com/en_ctcpp_online_presentations.html

Testwell CTC++

Testwell CTC++ Test Coverage Analyser for C and C++
CTC++ add-on for Java and Android
CTC++ add-on for C#


CTC++ Coverage Report - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://www.verifysoft.com/CTCHTML/index.html

CTC++ Coverage Report - Files Summary

Directory Summary | Files Summary | Functions Summary | Execution Profile

```
Symbol file(s)         : MON.sym (Thu Jan 15 10:51:44 2009)
Data file(s)           : MON.dat (Thu Jan 15 10:51:45 2009)
Listing produced at    : Thu Jan 15 10:51:48 2009
Coverage view          : As instrumented

Input listing          : STDIN
Html generated at      : Thu Jan 15 10:51:48 2009
ctc2html v2.4 options  :
Threshold percent      : 100 %

TER %  - covered/ all          File

Directory: .
  82 % -       14/ 17          calc.c
  80 % -        4/ 5           io.c
 100 %          6/ 6           prime.c
  86 % -       24/ 28          DIRECTORY

  86 % -       24/ 28          OVERALL
```

```
TER % - covered/all     File

100 %     6/6           prime.c
 80 %     4/5           io.c
 82 %    14/17          calc.c

 86 %    24/28          overall

Number of monitored source files : 3
Number of source lines : 59
Number of measurement points : 30
TER : 86 % (multicondition)
```

All coverage levels
- Statement coverage
- Function coverage
- Decision/branch coverage
- Condition coverage
- Modified condition/decision cov.
  - MC/DC coverage
- Multicondition coverage (MCC)

All compilers
All embedded targets!
Works with all unit test tools

Compliance with Standards

DO-178C - IEC 61508 - IEC 62304 - ISO 26262

Customers

Verifysoft TECHNOLOGY

AIRBUS DEFENCE & SPACE · AIRBUS HELICOPTERS · AKKA TECHNOLOGIES · Alcatel-Lucent · AMG · ATTENSITY · Audi

BARCO · BMW · BOMBARDIER · BOSCH · BULL · BT · Federal Office for Information Security

carmeq software & systems · CINTERION WIRELESS MODULES · COMARCH INFORMATION TECHNOLOGY · Continental · DAIMLER · DELTA DELTA ENERGY SYSTEMS · DGA

dSPACE · EATON · ESO · esa · gemalto · General Electric imagination at work · Giesecke & Devrient security at work

HELLA · HITACHI Inspire the Next · Honeywell · Infineon technologies · intel inside · jenaoptronik · KNORR-BREMSE

LIEBHERR · LUK · MAGNETI MARELLI · MAQUET GETINGE GROUP · MBtech Mercedes-Benz technology · Mentor Graphics · Oberthur Card Systems

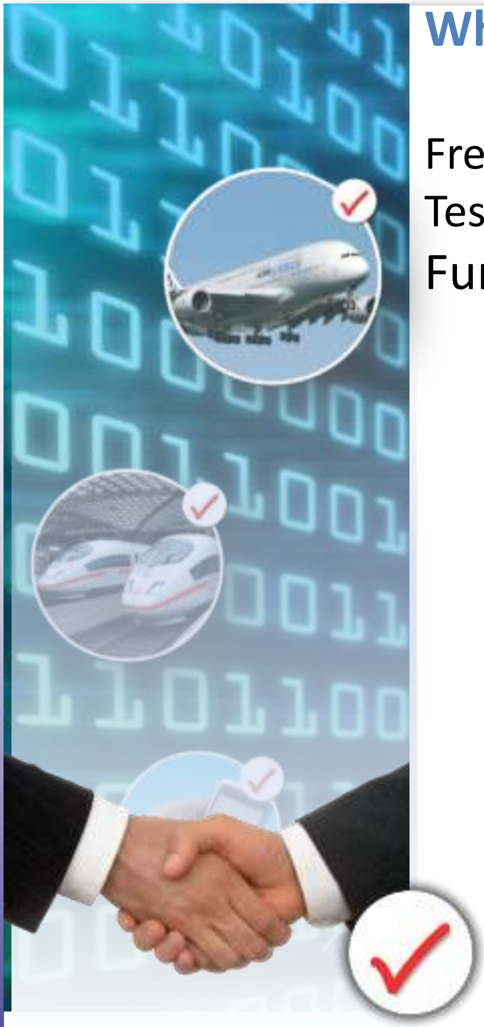
NXP · PHILIPS · Raytheon · RENESAS · RWE · SAFRAN Morpho · SAGEM

Schindler · Schneider Electric · SICK · SIEMENS · Sopra group · SPIE · ST

T-Mobile · T-Systems · TEXAS INSTRUMENTS · THALES · TOLL COLLECT service on the road · TOMTOM · VW

VIESSMANN · Westinghouse · WINCOR NIXDORF · ZEISS · ZF · ZF Lenksysteme · ZODIAC AEROSPACE

Thank You



What can we do for you?

Free tool evaluation incl. support
Testwell CTC++ Training
Further information: www.verifysoft.com

Thank you for your time!

Your Verifysoft Team