

Testwell CTC++ Test Coverage Analyser

Qu'est-ce que Testwell CTC++?

Testwell CTC++ est l'outil le plus performant pour mesurer la couverture de code sur ordinateur hôte et sur toutes les cibles embarquées, même très compactes.

L'outil est utilisé pour répondre aux exigences des normes de sécurité. Des centaines de clients à travers le monde utilisent Testwell CTC++ avec succès. Testwell CTC++ est le premier outil choisi par les entreprises qui ont à réaliser et à démontrer une couverture de code élevée dans l'aérospatiale, l'automobile, le transport, le médical ou autres industries.

Pourquoi la couverture de code?

La couverture de code est une mesure qui décrit le degré auquel le code source d'un programme est testé. Cela peut être considéré comme une mesure indirecte de la qualité du logiciel.

La couverture de code trouve les zones d'un programme qui n'ont pas encore été exécutés par les cas de test. De cette façon, il vous accompagne en créant des cas de test supplémentaires pour augmenter la couverture de code et vous empêche d'écrire des cas de tests redondants. La couverture de code est très utile lors de la phase de test, mais elle peut être également bénéfique lors de tests d'intégration et à d'autres moments, en fonction de vos processus de test.

La couverture de code est «fortement recommandée» (ce qui signifie de facto obligatoire) pour la sécurité des logiciels critiques. Les normes de sécurité comme la DO-178C (Considérations logicielles dans les systèmes aéroportés et équipement de certification), l'IEC / EN 61508 (sécurité fonctionnelle des systèmes électriques / électroniques, électroniques programmables relatifs à la sécurité), l'EN 50128 (Applications ferroviaires - Communication, Systèmes de traitement de signalisation et - Logiciel pour le contrôle ferroviaire et des systèmes de protection), l'IEC 60880 (puissance nucléaire) et l'ISO 26262 (sécurité fonctionnelle des véhicules routiers) demandent différents niveaux de couverture de code en fonction du niveau de l'application de sécurité.

Mais elle est également utilisée dans le développement de logiciels non-critiques, pour garantir leur sécurité et pour des questions de qualité. La couverture de code permet d'assurer un code de haute qualité.





Le succès avéré de Testwell CTC ++

Testwell CTC++, analyseur de couverture, répond aux besoins de ses utilisateurs depuis 1989. Les premières versions ont été développées au sein du groupe NOKIA sur le site de Tampere (Finlande) par les sociétés Oy Ab SoftPlan, Nokia Data Systems Oy et ICL Personal Systems Oy. En 1992, la société Testwell a été fondée avec pour mission de développer, de commercialiser et d'assurer le soutien technique des outils "Testwell". Le fondateur de la société Testwell, Monsieur Olavi Poutanen, était déjà en charge du développement de ces outils lorsque ceux-ci étaient développés par les sociétés précédemment présentées.

Après avoir développé les ventes pendant dix ans en Europe, Verifysoft Technology GmbH (Offenburg/Allemagne) acquiert Testwell CTC++ en 2013.

Aujourd'hui, des centaines de clients dans plus de 30 pays utilisent et recommandent l'outil Testwell CTC++.

Qu'est ce qui garantit le succès de Testwell CTC++ ?

Testwell CTC++ est le premier choix pour l'analyse de couverture de code.

L'outil est très facile à utiliser. Il dispose d'une prise en charge des langages de programmation C, C++, Java et C# et analyse tous les niveaux de couverture jusqu'à la couverture de condition / décision modifiée (MC / DC) et la couverture de conditions multiples. Testwell CTC++ fonctionne avec tous les compilateurs et compilateurs croisés. L'outil a un coût d'instrumentation très faible. Preuve à l'appui, il fonctionne sur toutes les cibles et microcontrôleurs embarqués.

Testwell CTC++ est très utilisé pour le développement de logiciels embarqués.

L'outil est intégré dans de nombreux environnements de développement et beaucoup de chaînes outillées et environnements de test. Les rapports de couverture sont clairs et significatifs. Testwell CTC++ a été comparé à une Jeep de l'armée: Simple à utiliser, peut être utilisé sur tous les terrains ou presque. Vous êtes à la recherche d'un outil de couverture de code qui fonctionne dans toutes les situations? Testwell CTC++ est votre meilleur choix! Testwell CTC++ est utilisé pour la sécurité du développement critique. Il est conforme aux normes de sécurité comme la DO-178C, l'ISO 26262, l'IEC 60880 et l'IEC 61508. Nous délivrons également un kit de qualification pour Testwell CTC++, qui est utilisé lors des processus de certification.



Comment fonctionne Testwell CTC++?

Comme annoncé précédemment, Testwell CTC++ est très facile à utiliser. Aucune modification du code de l'utilisateur n'est nécessaire. L'analyse de couverture de code se fait par l'instrumentation du code. Des compteurs d'exécution sont automatiquement ajoutés au code source et comptent combien de fois le code a été exécuté (testé). L'instrumentation est réalisée en ajoutant simplement "ctc" avant la commande de compilation / édition de liens.

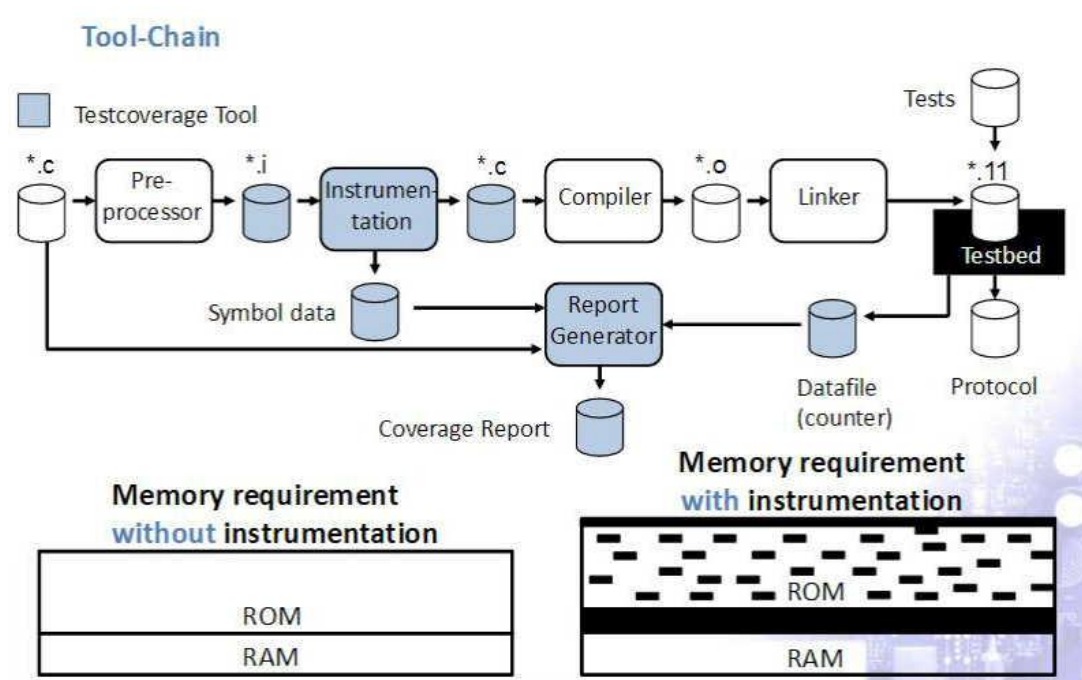
Il est très simple d'appliquer les changements nécessaires sur les makefiles afin de construire les cibles instrumentées. Cela ne nécessite pas de modifications au fichier makefile lui-même.

Testwell CTC++ peut être utilisé via **la ligne de commande** ou directement à partir de **nombreux environnements de développement**.

Lorsqu'il est utilisé en mode ligne de commande (par makefiles ou autres scripts de compilation), l'instrumentation est juste une phase insérée avant la commande de compilation / lien. Aucune modification des fichiers sources ou de construction n'est nécessaire. Les suites de tests sont effectuées avec le programme instrumenté, de la même façon que pour le programme d'origine. Comme Testwell CTC++ a une très faible surcharge d'instrumentation sur la taille et la vitesse d'exécution, ainsi que des concepts techniques spéciaux pour l'analyse de couverture de code sur cible, il peut être utilisé pour toutes cibles embarquées (même les plus compactes) et tous les microcontrôleurs.

Les rapports de couverture et autres profils d'exécution sont facilement obtenus en texte brut, HTML, XML et formulaires Excel. La documentation est **instructive et bien organisée**. Les rapports donnent à la fois une vue de haut niveau, qui montre les pourcentages de couverture de synthèse des différents niveaux et une vue détaillée.

Naviguer parmi les résultats de couverture en HTML est très facile. Le pourcentage de couverture est montré par le tableau d'ensemble sous différentes couleurs. Un zoom sur de détail du niveau peut être fait en quelques clics et les emplacements de code non testés sont clairement indiqués, **associés au code source d'origine**.



Support de tous les compilateurs / compilateurs croisés

Chaque licence de Testwell CTC ++ prend en charge tous les compilateurs et compilateurs croisés.

Même les nouveaux compilateurs que vous pourriez utiliser à l'avenir seront pris en charge par votre licence Testwell CTC++. Pas besoin d'engendrer des coûts supplémentaires, Testwell CTC++ fonctionnera avec tous les nouveaux compilateurs que vous utiliserez.

Nous avons des paramètres prêts pour un grand nombre de compilateurs. Voici la liste actuelle (juin 2016):

Altium Tasking: cc166, ccm16c

Borland/Inprise/Paradigm/Codegear compilers: bcc, bcc32, pcc, pcc32 (Paradigm)

Cosmic compilers: cx6805, cx6808, cx6812, cxs12x, cxxgate, cx6811, cx6816, cx332, cxst10, cxstm8, cxst7

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, HPUX CC, HP C++, aCC

IAR compilers and toolchains: iccm16c, icc430, icc8051, iccarm, iccavr, iccavr32, icccf, icchcs12, iccmaxq, iccdspic, iccpic18, icccl16c, icc78k, icc78k0r, iccv850, icch8, iccm32c, iccr32c, iccsam8

Fujitsu/Softune: fcc907s, fcc911s

GHS/GreenHills/Multi: ccv850, cxv850, ccmips, cxmips, ccarm, cxarm, ccppc, cxppc, gcc (GreenHill, not GNU)

Hitachi: shc, shcpp, ch38, ccrx

HI-Tech PICC compilers (Windows and Linux): picc, picc18, picc32, dspicc

VisualDSP++: ccbkfn, cc21k, ccts

Intel compilers (all platforms): icc, ic86, ic96

Java compilers: javac, jikes, ecj, gcj, kaffe

Keil compiler: c51, c166, c251, ca, armcc

Matlab/Simulink: lcc

Metaware: hcarm and others

Microsoft compiler: cl on host both 32 and 64 bit, cl for Smartphones and PocketPC, csc C# compiler, vjc J# compiler

Mitsubishi: nc30, nc308, nc77, nc79

Mono compilers: mcs, gmcs, smcs, dmcs

Motorola: chc12

NEC: ca830, ca850

Pathscale pathcc/pathCC

Renesas: shc, shcpp, ch38, ccrx, nc, nc308, nc77, nc79A, cc32R, CS+/CubeSuite+ cc78k0, cc78k0r, cx, ca850

Sun compilers: WorkShop compilers, javac

Symbian: various compilers

TI Code Composer Studio: cl2000

Trimedia: tmcc

Windriver: ccarm, ccsmipc, g++simpc, g++arm, cchppa, cesimso, ccsparc, cc68k, cc386, cc960, ccmips, ccppc

Veillez noter que nos clients ont déjà utilisé Testwell CTC++ avec d'autres compilateurs que ceux présents dans cette liste.

L'adaptation aux nouveaux compilateurs est facile et gratuite pour nos clients. Vous pouvez même le faire vous-même.

D'autres paramètres du compilateur sont ajoutés dès que nos clients en ont le besoin. Merci de cliquer sur ce lien http://www.verifysoft.com/en/code_coverage_all_compilers.html pour avoir la liste actualisée.



Intégration de Testwell CTC++ dans les EDIs

Testwell CTC++ est intégré dans des nombreuses chaînes outillées, environnements de test et outils de qualité logicielle.

Testwell CTC++ analyseur de couverture de code est actuellement (juin 2016) intégré dans les EDIs suivants : Visual Studio, IAR-Workbench, Borland 5.02, BeckIPC, Eclipse, Fujitsu Softune, Renesas. L'outil fonctionne également avec Keil μ Vision IDE (- c51, C166, c251, ca, - armcc), et ARM DS-5 armcc.

Il est possible de faire fonctionner Testwell CTC ++ avec d'autres EDIs, même s'ils ne sont pas répertoriés ici.

Les exigences d'une telle intégration sont les suivants: Editiable Commandlinepattern, Editiable Makefile-generation, outils modifiables (like Code::Blocks-Tools).

Si une ou plusieurs de ces conditions sont remplies, vous pouvez intégrer CTC ++ vous-même.

Une vidéo expliquant comment intégrer Testwell CTC ++ dans votre IDE est disponible ici:

https://www.youtube.com/watch?v=Pen_YNk_fQ0&feature=youtu.be

Merci de nous contacter si vous êtes intéressé par une intégration de Testwell CTC ++ dans d'autres EDIs.

Intégration dans de nombreuses chaînes outillées et environnements de test

Testwell CTC ++ Test coverage analyser est actuellement (Juin 2016) intégré avec les chaînes outillées, les environnements de test et outils de qualité du logiciel suivants:

- CATIA Systems – AUTOSAR Builder (Dassault Systemes)
- dSpace SystemDesk
- dSpace TargetLink
- Imagix 4D
- Jenkins
- Lauterbach
- MATLAB Simulink
- PikeTec Time Partition Testing (TPT)
- SonarQube
- QTronic TestWeaver
- QTronic Silver

Merci de nous contacter si vous êtes intéressés par une intégration de Testwell CTC++ dans une autre chaîne outillée ou environnement de test. Verifysoft Technology GmbH est également intéressé par des partenariats avec d'autres distributeurs d'outils qui pourraient intégrer Testwell CTC++ dans leur(s) outil(s).

Pour rappel, Testwell CTC++ fonctionne avec la plupart des outils de test unitaires.

Des rapports clairs et significatifs

Testwell CTC++ analyse la couverture de code pour tous les niveaux de couverture jusqu'au niveau MC/DC et conditions multiples, et montre les résultats d'analyses par des rapports clairs et significatifs. Les rapports donnent à la fois une vue au plus haut niveau, qui montre les pourcentages de couverture pour plusieurs niveaux de synthèse, ainsi qu'une vue détaillée, où les informations d'exécution/non-exécution sont associées aux emplacements réels du code source.

Rapport de couverture (format HTML, hiérarchique avec 4 niveaux)

- Résumé du dossier : information générale d'en-tête

[Directory Summary](#) | [Files Summary](#) | [Functions Summary](#) | [Untested Code](#) | [Execution Profile](#)

Symbol file(s) : f:\ctcwork\Demos\cube\MON.sym (Tue Dec 01 11:54:53 2015)
: MON.sym (Thu Dec 03 11:39:22 2015)
Data file(s) : f:\ctcwork\Demos\cube\MON.dat (Tue Dec 01 11:55:29 2015)
: MON.dat (Thu Dec 03 11:40:14 2015)
Listing produced at : Thu Dec 03 11:47:52 2015
Coverage view : Reduced to MC/DC coverage
Input listing : <stdin>
HTML generated at : Thu Dec 3 13:47:52 2015
ctc2html v5.1 options : -t 75 --enable-stmtthreshold=85 -o webCTCHTML
Structural threshold : **75 %**
Statement threshold : **85 %**

(Click on header to sort)

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

Directories : 2
Source files : 7
Headers extracted : 0
Functions : 64
Source lines : 905
Measurement points : 221
TER structural : **67 % (151/225) MC/DC**
TER statement : **78 % (236/304)**

- Résumé des fichiers : zoom dans les fichiers des dossiers

ctc2html v5.1 options : -t 75 --enable-stmtthreshold=85 -o webCTCHTML
Structural threshold : **75 %**
Statement threshold : **85 %**

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: .		
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
67 % - (151/225) 	78 % - (236/304) 	OVERALL

Directories : 2
Source files : 7
Headers extracted : 0
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](#)

- Résumé des Fonctions : zoom dans les méthodes et fonctions contenus dans les fichiers

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)	3	4	is_prime()
				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)	4	5	io_ask()
100 % (2/2)	100 % (1/1)	3	18	io_report()
				io.c

Source file: [prime.c](#)
Instrumentation mode: multicondition **Reduced to:** MC/DC coverage
TER: 100 % (6/6) structural, 100 % (6/6) statement

- Profil d'exécution : zoom dans la vue détaillée, les compteurs d'exécution sont indiqués dans le code source. Les lignes exécutées partiellement sont indiquées en rouge.

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			
	3	4	int is_prime(unsigned val)
		5	{
		6	unsigned divisor;
		7	
	1	8	if (val == 1 val == 2 val == 3)
0		8	1: T _ _
1		8	2: F T _
0		8	3: F F T
	2	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
	1	9	return 1;
	1	10	if (val % 2 == 0)
	1	11	return 0;
0	1	12	for (divisor = 3; divisor < val / 2; divisor += 2)
		13	{
0	0	14	if (val % divisor == 0)
0		15	return 0;

Le listing du profil d'exécution affiche combien de fois chaque portion de code a été exécutée. Il affiche les portions de code n'ayant pas encore été exécutées durant le de test (rapport textuel).

Example of CTC++ Execution Profile Listing

```
*****
*          CTC++, Test Coverage Analyzer for C/C++, Version 8.0          *
*          *          *          *          *          *          *          *
*          EXECUTION PROFILE LISTING                                  *
*          *          *          *          *          *          *          *
*          Copyright (c) 1993-2013 Testwell Oy                       *
*          Copyright (c) 2013-2015 Verifysoft Technology GmbH        *
*****

Symbol file(s) used   : MON.sym (Tue Nov 17 08:13:14 2015)
Data file(s) used    : MON.dat (Tue Nov 17 08:13:44 2015)
Listing produced at  : Tue Nov 17 08:14:16 2015
Coverage view        : As instrumented

MONITORED SOURCE FILE : prime.c
INSTRUMENTATION MODE  : multicondition

HITS/TRUE      FALSE      LINE DESCRIPTION
-----
1              1          8 FUNCTION main()
3              1          12 while (( prime_candidate = io_ask ( ) ) > 0)
2              1          14   if (is_prime ( prime_candidate ))
                   15   }+
                   16   else
                   17   }+
                   18   }+
1              1          19   return 0
                   20 }

***TER 100 % ( 6/ 6) of FUNCTION main()
100 % ( 6/ 6) statement
-----
```

Le listing du code non testé affiche les portions de code non testées (rapport textuel)

```
HITS/TRUE      FALSE      LINE DESCRIPTION
-----
4              0          5 FUNCTION io_ask()
0              4 -        11   if (( amount = scanf ( "%u" , & val ) ) <= 0)
-----

MONITORED SOURCE FILE : calc.c
INSTRUMENTATION MODE  : multicondition

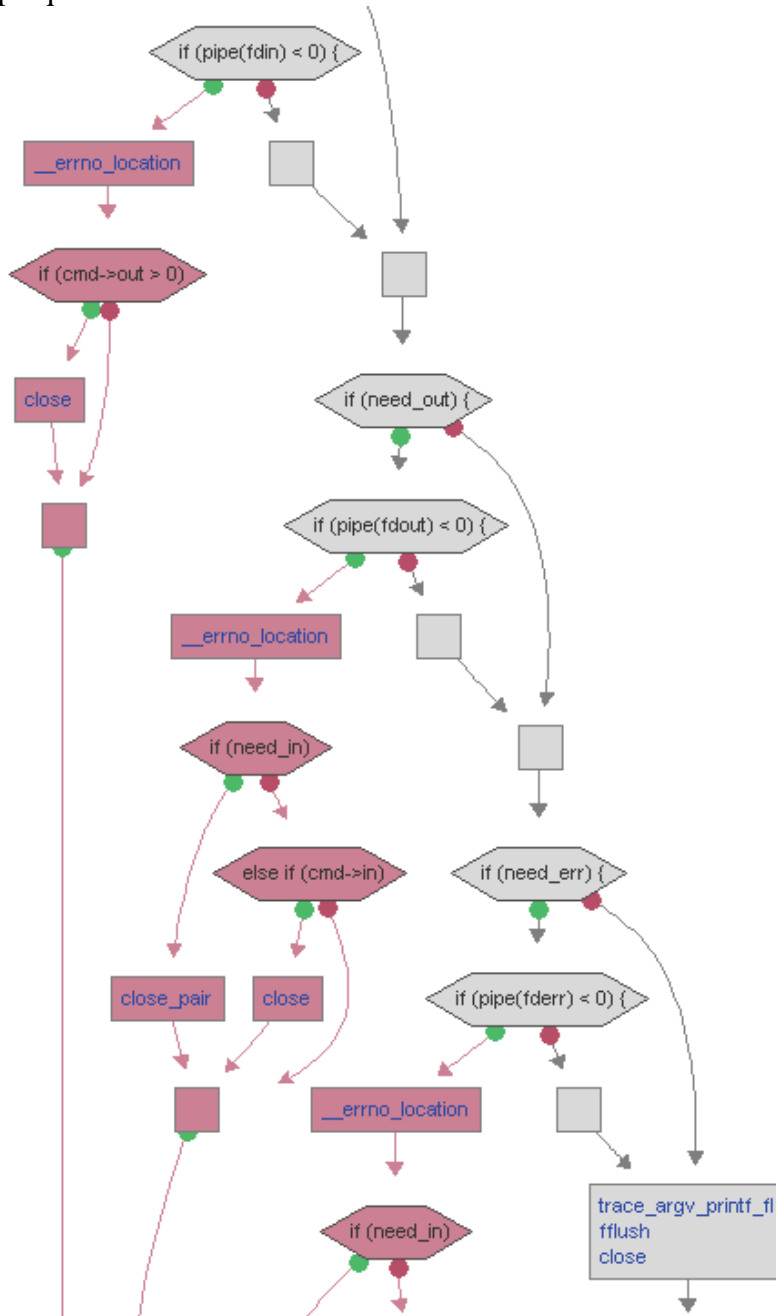
HITS/TRUE      FALSE      LINE DESCRIPTION
-----
3              0          4 FUNCTION is_prime()
1              2          8   if (val == 1 || val == 2 || val == 3)
0              -          8       1: T || _ || _
0              -          8       3: F || F || T
0              1 -        12   for (;divisor < val / 2;)
0              0 -        14       if (val % divisor == 0)
0              -          15       return 0
-----

SUMMARY
=====
Source files       : 3
Headers extracted  : 0
Source lines       : 59
Measurement points : 27
TER                : 76 % (22/29) multicondition
TER                : 88 % (21/24) statement
```


Le listing du temps d'exécution affiche le temps d'exécution cumulé et moyen des fonctions (rapport textuel).

Contrôle de flux Graphique (Imagix 4D)

En utilisant Testwell CTC++ avec Imagix 4D les données de couverture sont représentées dans un graphique de flux de contrôle.



Dans ce schéma, les symboles en forme de losange représentent les décisions. Les blocs de lignes de code sont représentés dans des rectangles. Le code non exécuté est représenté dans les formes et chemins en rouge.

Annotations dans les rapports de couverture

Avec Testwell CTC++ vous pouvez introduire des commentaires de textes à votre code source, qui resteront disponibles tout au long du processus CTC++ au terme d'un Rapport HTML.

Les annotations peuvent être utilisées, par exemple, pour expliquer pourquoi certaines portions de code ne sont pas exécutées dans les tests (image ci-dessous CF., texte jaune).

Hits/True	False	Line	Source
			<code>#include "foo.h"</code>
		1	<code>/* File calc.c ----- */</code>
		2	<code>#include "foo.h"</code>
		3	<code>#include "calc.h"</code>
		4	<code>/* Tell if the argument is a prime (ret 1) or not (ret 0) */</code>
Top			
9		5	<code>int is_prime(unsigned val)</code>
		6	<code>{</code>
		7	<code> unsigned divisor;</code>
		8	<code>#pragma CTC_ANNOTATION header files als echte Dateien von ctspost</code>
		9	<code> foo();</code>
2	7	10	<code> if (val == 1 val == 2 val == 3)</code>
1		10	<code>1: T _ _</code>
0		10	<code>2: F T _</code>
1		10	<code>3: F F T</code>
	7	10	<code>4: F F F</code>
2		11	<code> return 1;</code>
5	2	12	<code> if (val % 2 == 0)</code>
5		13	<code> return 0;</code>
58	2	14	<code> for (divisor = 3; divisor < val / 2; divisor += 2)</code>
		15	<code> {</code>
0	58	16	<code> if (val % divisor == 0)</code>
0		17	<code> return 0;</code>
		18	<code> }</code>
2		19	<code> return 1;</code>
		20	<code>}</code>
***TER 82 % (14/17) of FILE calc.c			
92 % (11/12) statement			
Directory Summary Files Summary Functions Summary Untested Code Execution Profile			
To files: First Previous Next Last Top Index No Index			

Exécution de la couverture du noyau

Avec Testwell CTC ++, vous pouvez analyser la couverture de test d'un noyau en cours d'exécution (Kernelcoverage).

Mesurer la couverture du code de l'espace noyau est généralement un défi pour les outils de couverture basés sur de l'instrumentation. Dans le code de l'espace du noyau, les sondes instrumentées ne peuvent utiliser aucune fonction de bibliothèques ou d'appels système alors que cela est possible au niveau de l'espace utilisateur. La méthode de Testwell CTC++ pour implémenter cette instrumentation et son support de la couche temps-réel suppose du code C de base sans utilisation des services système. Ainsi il peut être exécuté sans problème dans l'espace noyau.

Langages de programmation supportés

Testwell CTC++ analyse le code écrit en C, C++, Java et C#. Initialement, Testwell CTC++ a été développé pour les langages C et C++. Depuis 2007 nous fournissons des modules additionnels pour étendre les fonctionnalités aux langages de programmation Java et C#.

Kit de qualification disponible

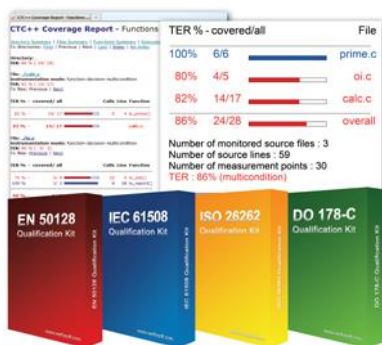
Le kit de qualification pour Testwell CTC++ fournit de la documentation, des cas de test et des procédures qui vous permettent de qualifier l'analyseur de couverture Testwell CTC++ pour les projets basés sur les normes de sécurité DO-178C, EN-50128, IEC 61508, IEC 60880 et ISO 26262. Le kit contient des plans de qualification de l'outil, ses exigences opérationnelles et d'autres éléments nécessaires pour la qualification de Testwell CTC++ pour une utilisation en sécurité des projets critiques. Pour chaque fonction utilisée de Testwell CTC++, l'utilisateur est en mesure d'exécuter des cas de test dans son environnement qui démontrent l'absence d'erreurs. Le kit facilite la certification des systèmes embarqués qui utilisent Testwell CTC++ pour l'analyse de couverture de test. L'utilisateur peut modifier les artefacts dans le kit de qualification de l'outil pour son projet spécifique.

Le kit de qualification pour Testwell CTC++ est conforme aux normes DO-178C, EN-50128, IEC 61508, IEC 60880 et ISO 26262. Il est composé de:

D'outils d'aide à la qualification qui guide l'utilisateur à travers la qualification et génère les documents suivants:

- Rapport sur la classification de l'outil
- Plan/rapport de qualification de l'outil
- Manuel de sécurité de l'outil
- Plan de test
- L'unité d'automatisation des tests
- La suite de tests avec des cas de test
- Le manuel d'utilisation du kit de qualification.

Le kit de qualification pour Testwell CTC++ est utilisé avec succès par des sociétés des secteurs de l'aérospatial et de l'automobile. La plupart de nos clients confirme que le kit est l'un des meilleurs du marché.



Que disent nos clients à propos de Testwell CTC++?

Plus de 400 clients dans plus de 30 pays utilisent Testwell CTC++ avec satisfaction pour s'assurer de la qualité de leur projet et notamment pour mesurer la couverture de code dans les projets critiques.

Voici ce que les utilisateurs Testwell CTC++ disent sur l'outil:

"Testwell CTC++ est utilisé dans l'un de nos projets sur cibles embarquées. Cela a été facile à intégrer sur la cible embarquée. CTC++ dispose d'un mode d'emploi bien écrit et Verifysoft offre un bon service à sa clientèle."

Anna Andgart, développeur de logiciels
ABB AB, Technologies de contrôle, Suède

"Nous utilisons Testwell CTC++ très intensivement au cours de notre développement et le test de logiciels critiques pour les centrales nucléaires.

Cet outil n'est pas seulement utilisé sur les systèmes hôtes, mais aussi sur les différentes plates-formes cibles embarquées. L'objectif est de mesurer facilement les couvertures de type déclaration/décision ou MC / DC pour satisfaire les exigences de test du logiciel de domaine nucléaire norme IEC60880 et le document de position commune "licences de la sécurité des logiciels critiques pour les réacteurs nucléaires" qui stipule les exigences du point de vue européen des régulateurs pour le nucléaire.

Le soutien de Verifysoft est très utile, rapide, direct et simple. "

Thorsten Oertel, Senior Firmware Engineer
AREVA GmbH, Allemagne

"Une couverture manquante est principalement basée sur des exigences manquantes ou des spécifications de test incomplètes. Nous utilisons donc CTC++ dans le développement d'un système d'aide à la conduite avancée pour trouver ces exigences manquantes et compléter nos spécifications de test. Cela nous aide à mettre en place un processus de développement selon la norme ISO 26262. Il était facile d'intégrer CTC++ dans notre ensemble d'outils de développement existant et CTC++ produit des résultats immédiats."

Michael Kalusche, Project manager
Bertrandt Ingenieurbüro GmbH, Allemagne

"Nous utilisons CTC++ sur les logiciels embarqués des combinés d'instruments pour les tests des modules. Les gros avantages sont une reproductibilité élevée et une exécution rapide. Avec cet outil, nous pouvons tester les modules logiciels complets sur chaque petite modification et pas seulement la partie modifiée. Il en découle des résultats beaucoup plus sûrs dans un temps plus court."

Iaran Gadotti, R&D Manager
Continental Brasil Indústria Automotiva Ltda., Brasil



"IAV est l'un des plus importants fournisseurs de services de développement de l'industrie automobile en développement logiciel pour l'électronique de l'habitable. Nous avons choisi Testwell CTC++: parce qu'il soutient ISO26262, SPICE et ASIL B et aussi car il peut facilement être intégré dans les divers environnements différents de nos clients. Testwell CTC++ fournit un excellent support pour les activités de test d'IAV au niveau module pour mesurer la couverture de code, en parallèle sur le calculateur hôte et la plateforme cible.

En outre, nous avons intégré Testwell CTC++ dans notre intégration continue et construit un processus pour identifier rapidement les lacunes dans la couverture du code afin de surveiller en permanence la qualité et la couverture du code. Testwell CTC++ est désormais partie intégrante de la chaîne d'outils IAV."

**Marko Meyer, gestionnaire principal de projet
IAV GmbH, Allemagne**

"Nous utilisons CTC++ largement pour accompagner nos développements et nos tests de logiciels de sécurité critique sur nos cibles embarquées. CTC++ nous permet de traquer la couverture de plusieurs fragments de notre logiciel, afin que nous puissions ajuster nos tests décrits précisément pour couvrir toutes les fonctions de branches. Cela sert en tant que confirmation de certifications pour l'autorité de contrôle."

**Thomas Bartzick, Department of Software Testing
ISH, Allemagne**

"Nous utilisons CTC++ dans nos projets aérospatiaux. L'objectif est "Analyse de couverture de test basée sur les exigences" pour satisfaire les exigences de tests DO178B. L'outil nous aide dans notre analyse de code source C par exemple pour localiser le code mort. Nous sommes en mesure d'utiliser CTC++ sans aucun problème."

**Michael Goersdorf, développement de logiciels/ logiciels de R&D
Kappa optronique GmbH, Allemagne**

"Nous utilisons Testwell CTC++ pour notre projet embarqué. Cela nous a permis de découvrir tous les parties de code et les chemins de contrôle non couverts, à cause d'une spécification de test unitaires incomplète. L'intégration est simple et le support clients est excellent"

**Srinivasulu, Project Manager
Knorr-Bremse Technical Center, India**

"Testwell CTC++ est un outil qualifié IEC61508 T3 que nous utilisons pour tester les applications embarquées. Nous l'avons utilisé à la fois pour l'analyse de couverture de code sur cible et sur Windows. L'outil est facile à intégrer sur des applications embarquées et les outils d'analyse génèrent de très bons rapports."

**Santiago P. Sánchez-Manjavacas, Architecte principal Firmware / FPGA
Kongsberg Maritime, plate-forme produit, Norvège**



"Nous utilisons CTC++ pour mesurer la couverture des tests unitaires et système de nos cibles (embarquées). Les rapports sont parfaits, simples et contiennent ce dont nous avons besoin. Le soutien de Verifysoft est convenable et rapide. L'outillage de CTC peut facilement être ajouté pour faciliter la construction des versions."

Kees Valkhof, Testeur

Lely, aux Pays-Bas

"Nous utilisons Testwell CTC++ pour mesurer la couverture de code du logiciel embarqué de nos produits médicaux au niveau du test unitaire.

Testwell CTC++ a pu être très bien intégré dans notre environnement de développement et délivre depuis l'information désirée rapidement et avec fiabilité.

Le support technique à propos des questions éventuelles était rapide et toujours en mesure de nous aider.

Nous pourrions recommander Testwell CTC++ sans aucune restriction."

Heiko Schmidt, Software Team Manager

MAQUET Cardiopulmonary AG Rastatt, Allemagne

"REC Global est le partenaire de développement de logiciels embarqués de plusieurs grands équipementiers automobiles. Comme Testwell CTC++ est reconnu comme l'outil de choix pour l'assurance de la qualité du logiciel par nos clients, nous avons suivi leur exemple et l'avons également implémenté sur nos projets. Nous l'utilisons pour les tests d'applications embarquées. Les rapports de CTC++ servent en tant que mesure objective de la qualité du test et nous aident à améliorer notre processus de développement."

Borivoje Dermanovic, Project Manager

REC Global, Croatie

"Volvo Powertrain utilise CTC ++ car ce logiciel supporte la norme ISO26262 et SPICE et parce qu'il fonctionne bien avec la mesure de la couverture de code dans notre plate-forme de module de test."

Johannes Foufas, développeur

Volvocars, Suède

"Nous utilisons Testwell CTC ++ pour analyser la couverture de test pour nos tests unitaires et tests de système de notre produit de sécurité qui est développé selon la norme IEC61508. Les rapports créés par Testwell CTC ++ sont clairs, simples, et contiennent toutes les informations nécessaires. Il était facile d'intégrer Testwell CTC ++ dans notre chaîne de développement outillée. Verifysoft fournit un excellent soutien à la clientèle."

Thomas Schneider, Senior Software Engineer

Schneider Electric Automation GmbH, Allemagne



"Nous mettons en œuvre l'automatisation des tests sur des projets avec du code existant depuis quinze ans, devant être conforme aux normes des logiciels définies par l'office de contrôle des produits pharmaceutiques et alimentaires des Etats-Unis (US FDA). L'évaluation de la couverture de test basée sur les spécifications fonctionnelles ("tests basés sur les exigences") est inadaptée dans cette situation. Testwell CTC++ nous permet d'exécuter de nombreuses suites de tests et de déterminer les voies logiques qui ne sont pas couvertes. Testwell CTC++ est un excellent produit."

Robert Evans, Software Development Engineer
Siemens Medical Diagnostics, USA

Résumé :

Des centaines de clients utilisent Testwell CTC++ comme analyseur de couverture de code car il supporte tous les compilateurs et toutes les cibles embarquées, même les plus petites et tous les microcontrôleurs embarqués.

Plus d'informations :

Pour plus d'informations, y compris sur les nouveautés de Testwell CTC ++, veuillez vous référer à notre page d'accueil : http://www.verifysoft.com/fr_ctcpp.html

Pour les webinaires et les vidéos sur Testwell CTC ++, ainsi que sur les normes de sécurité qui y sont associées, merci de vous référer au lien :

http://www.verifysoft.com/fr_ctcpp_online_presentations.html

Actuellement (juin 2016), voici les vidéos disponibles:

Présentation générale:

- Testwell CTC++ General Presentation
- Code Coverage for Embedded Targets
- Testwell CTC++ Short Introduction (Prime Example)
- Safety Standards and related Code Coverage Levels
- ISO 26262 and Code Coverage
- Usage with special IDEs / environments:
 - Usage of Testwell CTC++ with Microsoft Visual Studio 2008 IDE
 - Usage of Testwell CTC++ with IAR Embedded Workbench IDE
 - Usage of Testwell CTC++ for Embedded Targets (demo is based on an Atmel ATmega 328p μ Controller)
 - How to integrate Testwell CTC++ in your IDE?
 - Integration of Testwell CTC++ in Eclipse
 - Testwell CTC++ and MATLAB/Simulink interface example
 - Testwell CTC++ Code Coverage Lauterbach Trace32
 - Usage of Testwell CTC++ with Renesas CS+ IDE
 - Usage of Testwell CTC++ with Microship MPLAB IDE
- Videos for CTC++ Users:
 - Code Coverage on Embedded Targets with HoTa for Testwell CTC++ Users

Pour toutes questions, merci de contacter notre équipe Verifysoft Technology :

Verifysoft Technology GmbH
In der Spöck 10-12
77656 Offenburg
Germany
qualify @ verifysoft.com
Téléphone France: +33 3.68.33.58.84



Copyright Verifysoft Technology GmbH Version 1.1.2 07/2016

Testwell CTC++ is a product and a trademark of Verifysoft Technology GmbH, Imagix 4D is a product of Imagix Inc. USA, all other trademarks are property of their respective holders. Photos: Fotolia (p 1, 2, 5, 11,12,), all other Verifysoft Technology GmbH