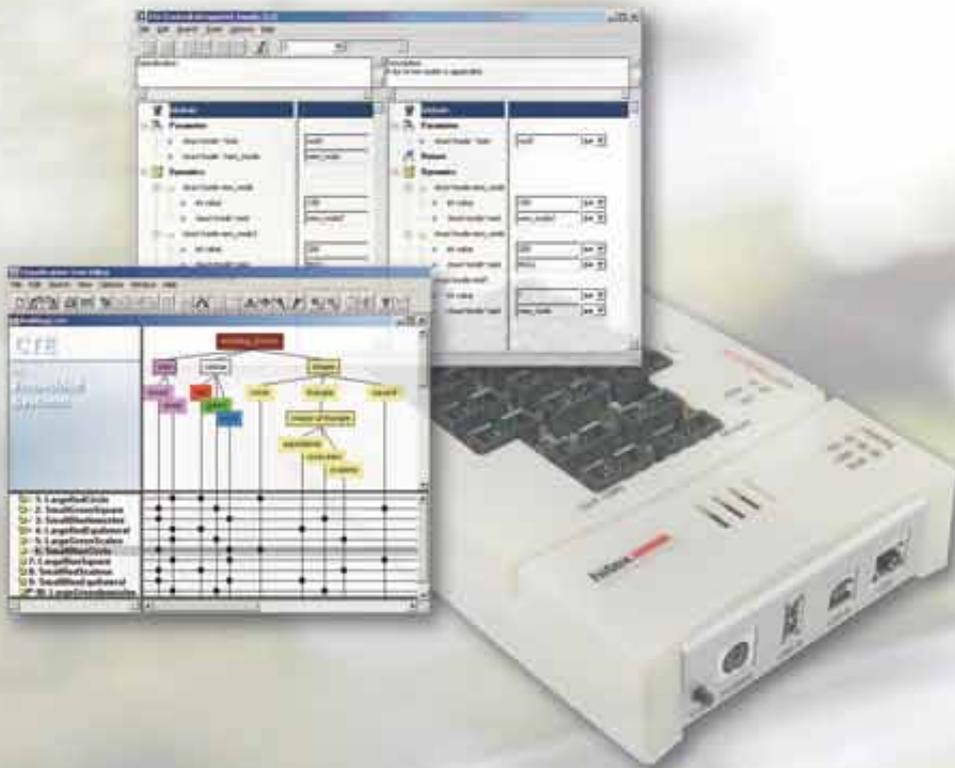


# Tessy

Test d'intégration et unitaire dynamique automatisé  
pour des applications embarquées

# CTE

Éditeur de classification arborescente  
pour spécifications du cas de test



*Embedding Software Quality*

# Le meilleur outil de test unitaire et d'intégration automatisé

## Tessy – Un outil de test irremplaçable

Tessy exécute des tests d'intégration et unitaire dynamique automatisé pour des logiciels embarqués et détermine en parallèle la couverture de code. Ce type de test est obligatoire pour l'obtention des certifications telles que DO-178B, CEI 61508 ou ISO 26262.

L'Éditeur de classification arborescente (CTE) fournit une méthode permettant de spécifier et créer systématiquement des cas de test. Il utilise pour ce faire la méthode de classification arborescente, une fonction spécifique de Tessy.

Tessy exécute les tests automatiquement, en évalue les résultats et génère les rapports de test correspondants.

Tessy permet d'éliminer les tests manuels, faisant ainsi gagner énormément de temps à l'ingénieur chargé

## Fonctions clés

- > Exécution automatisée des tests
- > Création de rapport de test
- > Couverture de code sans effort supplémentaire
- > Test de non-régression et d'intégration
- > Essentiel pour l'obtention de certification
- > Pour les projets distribués géographiquement
- > Test réalisés sur l'hôte ou sur le matériel cible correspondant
- > Prend en charge C et C++
- > Accélère le développement – rapide retour sur investissement

[www.hitex.com/perm/tessy.html](http://www.hitex.com/perm/tessy.html)



*Tessy vous aide à développer des applications critiques pour la sécurité*

du développement des applications embarquées. L'accélération du processus de développement offre non seulement un retour sur investissement plus rapide mais aussi un logiciel de meilleure qualité et des tests documentés !

Il est possible d'exécuter les tests sur l'hôte ayant servi au développement ou sur le matériel cible correspondant.

Tessy est utilisé avec succès dans d'importants projets impliquant des dizaines d'utilisateurs dans plusieurs sites, et ce dans le monde entier. Tessy est très largement employé pour les applications des secteurs de l'automobile, de l'aérospatial, de l'électronique aéronautique, du chemin de fer, de la santé, de l'armée et dans l'industrie. N'hésitez pas à nous demander des témoignages !

Les tests de non-régression représentent un facteur clé de la qualité du logiciel. Tessy est l'outil parfait pour cette tâche.

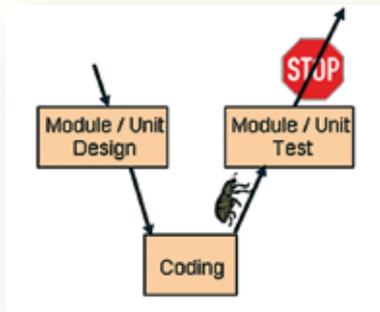
# Des tests systématiques, rigoureux et spécifiques



## Qu'est-ce que le test unitaire ?

Pendant le test unitaire de programmes C, une seule fonction de niveau C est testée rigoureusement et isolément du reste de l'application. Ce test unitaire est souvent aussi appelé test modulaire.

Rigoureux signifie que les cas de test sont spécialement élaborés pour l'unité en question et qu'ils incluent des données d'entrée non prévues par l'unité soumise au test. Isolée signifie que le résultat du test ne dépend pas du comportement des autres unités de l'application. Cette spécificité peut être obtenue en appelant directement l'unité soumise au test et en remplaçant les appels des autres unités par des fonctions bouchons (« stub »).



***Le test unitaire élimine les erreurs à un stade précoce du projet et empêche leur apparition à un stade ultérieur du processus de développement***

## Quels sont les avantages du test unitaire ?

### Simplification de la spécification du cas de test

Au lieu d'essayer de créer des cas de test pour l'ensemble des unités entrant en interaction, les cas de test unitaires sont spécifiques à l'unité testée (système du « diviser pour conquérir »). Les cas de test peuvent facilement inclure des données d'entrée non prévues par l'unité testée, méthode dont la mise en œuvre peut s'avérer difficile avec des tests généraux de système.

### Facilité d'isolation des erreurs

Si l'unité est testée séparément des autres unités, il est alors plus simple de détecter la cause de l'échec d'un cas de test. L'erreur doit être liée à l'unité testée et pas à une unité d'un niveau inférieur dans la hiérarchie d'appel.

### Détection précoce des erreurs

Il est possible d'exécuter le test unitaire dès que l'unité à tester peut être compilée avec succès. Il est alors possible de détecter très tôt les erreurs qu'elle pourrait contenir.

### Économies

Il est largement admis que la correction d'erreurs détectées à un stade avancé d'un projet est plus onéreuse que celle d'erreurs détectées précocement. Le test unitaire vous permet donc de réaliser des économies.

### Confiance accrue

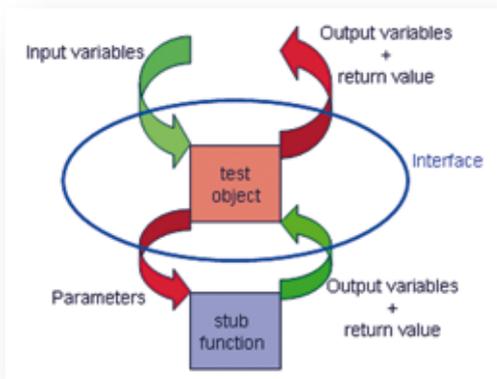
Le test unitaire accroît votre confiance. Grâce au test unitaire, l'application sera composée d'unités ayant été testées de manière exhaustive. Le succès du test de l'ensemble de l'application devient donc plus probable.

# Exploration du workflow de test

## 1. Lancez un test

Tessy démarre en analysant le module source puis laisse l'utilisateur sélectionner la fonction à tester. Il identifie ensuite l'interface de l'objet du test tel que les variables globales, les paramètres et les fonctions appelées par l'objet du test.

Tessy détermine si une variable est en entrée, en sortie ou les deux. Les résultats sont affichés dans l'éditeur TIE (Test Interface Editor). Tessy peut y être dirigé pour allouer la mémoire à utiliser comme cible pour les pointeurs de l'interface.

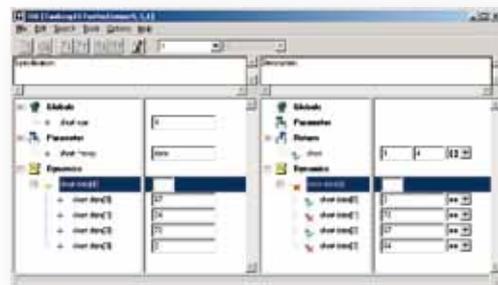


*L'interface sépare l'objet du test du reste de l'application*

## 2. Définissez votre cas

Tessy propose plusieurs méthodes de spécification des valeurs des cas de test.

- > La saisie interactive des données de test s'effectue via l'Éditeur des données de test (TDE), qui fonctionne comme un assistant.
- > Tessy combine automatiquement les valeurs fournies par l'utilisateur pour les cas de test. La création de cas de test destinés à couvrir des plages de valeur peut s'exécuter simplement.
- > Il est possible d'importer les données de test à partir de fichiers (classeurs Excel ou fichier de texte simple). Ceci permet de calculer les données de test avec Excel ou d'importer des données de test enregistrées.
- > Les cas de test générés systématiquement avec la méthode de classification arborescente peuvent être importés de l'Éditeur de classification arborescente (CTE). Ces tests peuvent inclure des données de test.
- > Tessy peut générer des données aléatoires.



*Il est possible d'entrer les données de test de manière interactive dans l'Éditeur de données de test (TDE)*

Un cas de test comprend des valeurs pour les variables d'entrée, les résultats prévus et une méthode de comparaison entre les résultats obtenus et ceux prévus pour décider de la réussite ou de l'échec d'un test. Il est possible d'accepter des écarts par rapport aux résultats prévus.

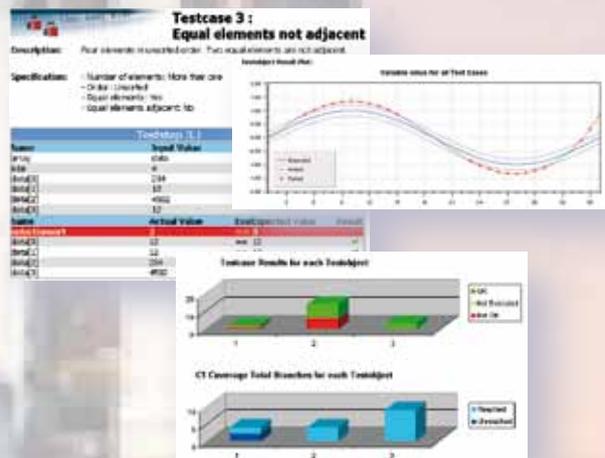
### 3. Rencontrez votre pilote de test

Tessy génère alors du code source pour le pilote de test chargé d'appeler la fonction en cours de test. Si cette fonction appelle une autre fonction, Tessy est capable de créer une fonction bouchon (« stub ») pour remplacer la fonction appelée. Ce processus est nécessaire pour le test unitaire au sens strict et utile si la fonction appelée n'est pas encore implémentée.

Tessy fournit deux types de fonction bouchon :

- > Un type permet à l'utilisateur de spécifier les valeurs prévues des variables d'entrée de la fonction bouchon, qui sont comparées aux valeurs obtenues par Tessy. En outre, ce type de fonction bouchon vous permet de spécifier les entrées de la fonction bouchon dans la fonction en cours de test, c'est-à-dire la valeur renvoyée.
- > L'autre type de fonction bouchon permet à l'utilisateur de fournir le code source du corps du bouchon.

Tessy peut également vérifier l'ordre des appels de la fonction bouchon.



*Les rapports de test sont générés automatiquement*

### 4. Allez Tessy !

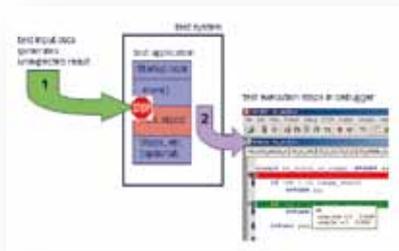
Tessy utilise un compilateur croisé pour compiler et lier le code source du pilote, la fonction testée et toute fonction bouchon; il télécharge ensuite l'exécutable ainsi obtenu dans le système de test. Il peut s'agir d'un émulateur in-circuit utilisé en mode autonome ou d'un émulateur connecté à un système cible ou encore d'un système de débogage JTAG / BDM / OCDS. Il peut également s'agir d'une simulation du microcontrôleur cible s'exécutant sur le PC hôte. Le test peut également être réalisé sur le PC hôte à l'aide d'un compilateur GNU natif.

Tessy exécute chaque cas de test puis détermine si celui-ci a réussi ou échoué. Un rapport de test est ensuite généré, avec des niveaux de détails et des formats configurables.

# Pourquoi Tessy facilite les tests

## Un débogage de rêve

Si un cas de test échoue, un débogage simple et efficace est exécuté. Tessy est capable d'exécuter à nouveau un cas de test et de commander le débogueur utilisé pour qu'il arrête l'exécution du test au début de la fonction soumise au test. Les fonctions du débogueur peuvent alors être utilisées pour démasquer le coupable. Une fois le code source modifié pour résoudre le problème, il est aisé d'exécuter une nouvelle fois le cas de test en question (ainsi que tous les autres) pour contrôler que la correction est efficace.



*Tessy facilite le débogage d'un cas de test ayant échoué*

## Réutilisation des données de test et gain de temps

Si un élément d'interface d'une fonction testée a été modifié pendant le processus de développement, Tessy permet à l'utilisateur de réutiliser les données de test de l'ancienne interface. Cette fonctionnalité s'avère très utile pour les tests de non-régression.



*Les anciens éléments d'interface peuvent être affectés aux nouveaux à l'aide de l'Éditeur d'affectation des données d'interface (IDA)*

## Test de non-régression – Vos modifications ont-elles entraîné des erreurs ?

Les tests de non-régression peuvent révéler si de nouvelles erreurs ont été introduites lors du processus de développement de l'application, comme le débogage d'autres sections, la réécriture de la fonction testée, l'emploi d'une nouvelle version du compilateur ou le portage du logiciel dans une autre architecture de microcontrôleur. Tessy intègre une fonctionnalité de test de non-régression facile à utiliser. Celle-ci constitue une aide irremplaçable pour le contrôle du logiciel modifié, assurant ainsi la qualité du logiciel.

## Tests en arrière-plan – Rentrez tranquillement chez vous

Tessy permet à l'utilisateur d'exécuter un ensemble sélectionné de cas de tests sans qu'il ait besoin d'intervenir. Il devient ainsi possible d'exécuter un test de non-régression exhaustif pendant la nuit et d'en analyser les résultats le lendemain.

## Prise en charge des fichiers ASAP2

Tessy prend en charge les fichiers ASAP2, ce qui permet à l'utilisateur d'employer des valeurs physiques (par exemple la température en degré Celsius) plutôt que la représentation interne.

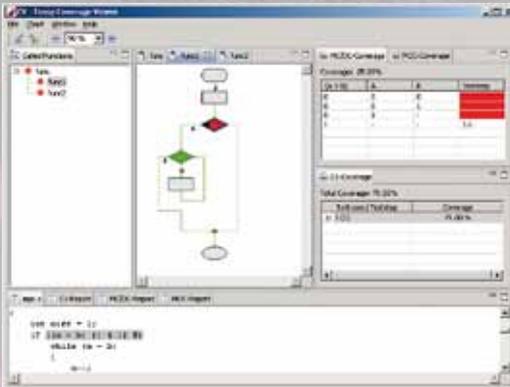
## Éditeur d'environnement – Gérez vos configurations

Il est possible de créer des configurations de Tessy pour un projet donné à l'aide de l'Éditeur d'environnement de Tessy (TEE). Plusieurs utilisateurs peuvent ainsi utiliser facilement la même configuration, simplifiant ainsi les projets de grande taille.

# Tessy, l'outil irremplaçable pour un logiciel de qualité

## Test d'intégration – Contrôle de l'interaction

Tessy peut tester l'interaction d'unités opérant conjointement, même si les unités ne s'appellent pas les unes les autres, par exemple les instructions push() et pop() d'une pile. Plusieurs unités sont agencées pour former une unité de plus grande taille, habituellement appelée module ou composant. Un cas de test appelle les unités dans un certain ordre et définit les variables du composant. Les appels résultants aux autres composants ainsi que les variables peuvent être évalués par Tessy. Les cas de test d'intégration peuvent également inclure un temps simulé (test du composant temporel).



Examen interactif des résultats de couverture

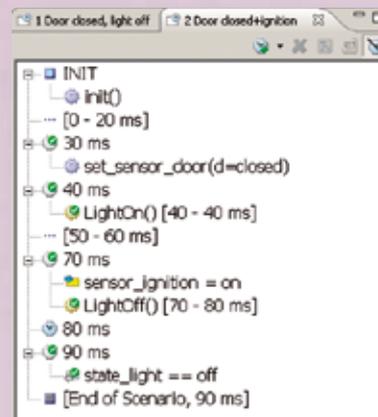
## Couverture de code – Garantie d'un test exhaustif

Sans effort supplémentaire, Tessy peut déterminer les mesures suivantes de couverture de code pendant le test:

- > Couverture de branche/décision (C1)
- > Condition modifiée/couverture de décision (MC/DC)
- > Couverture des conditions multiples (MCC).

L'afficheur de couverture de Tessy vous permet de visualiser les résultats. L'utilisateur peut afficher de manière interactive le code source lié à une certaine branche ou décision du logiciel. D'un seul clic, vous pouvez voir quelle branche du logiciel n'a pas été exécutée pendant les tests ou quelles branches du logiciel ont été exécutées par un cas de test déterminé.

La table de vérité des conditions d'une décision est affichée. Elle indique si des cas de test supplémentaires sont requis et, si nécessaire, leur valeurs. Tous les résultats de test de couverture sont également disponibles sous forme de rapports imprimables.



Un cas de test d'intégration avec simulation de temps pour le composant

## Microcontrôleurs pris en charge

Tessy prend couramment en charge plus de 130 combinaisons de microcontrôleurs/compilateurs croisés/débogueurs. Tessy est ainsi capable de traiter le code spécifique de microcontrôleurs non-ANSI-C qu'utilise certains compilateurs croisés. Tessy étant compatible avec plusieurs débogueurs, il peut effectuer les tests de manière automatique. La liste des combinaisons prises en charge est constamment étendue. Veuillez consulter [www.hitex.com/perm/tessy.html](http://www.hitex.com/perm/tessy.html).

Tessy s'exécute sous Windows NT / 2000 / XP / Vista et Windows 7. Tessy et CTE ont tous les deux pour origine l'ancien laboratoire de technologie logicielle de Daimler AG à Berlin, Allemagne.

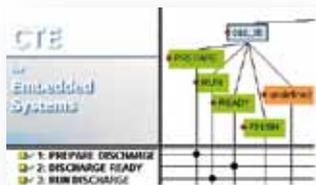
# La spécification systématique de cas de test

## CTE et la méthode de classification arborescente

La méthode de classification arborescente assiste un développeur devant faire face à des problèmes tels que:

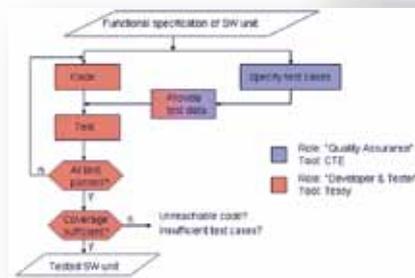
- > trouver les cas de tests «adaptés»
- > minimiser l'ensemble de cas de test tout en garantissant leur exhaustivité
- > estimer l'étendue des tests nécessaires
- > définir les critères de fin d'exécution des tests sans risquer l'intégrité du processus de test

La méthode de classification arborescente transforme de manière systématique une spécification de problème en une série de cas de test sensible aux erreurs et peu redondante. Cette méthode classe les aspects pertinents pour le test au moyen de la méthode de segmentation d'équivalence. Les spécifications du cas de test en sont le résultat.



*Extrait d'une spécification de cas de test dans le CTE*

La méthode de classification arborescente est intuitive et d'apprentissage facile. Elle fait appel et encourage la créativité du développeur. La réflexion sur la spécification d'un problème étant au début du processus, la méthode de classification arborescente révèle également les incohérences ou les omissions de la spécification du problème.



*La spécification de cas de test et l'exécution peuvent être séparées à l'aide de Tessy et du CTE*

L'Éditeur de classification arborescente (CTE) est un outil graphique prenant en charge la méthode de classification arborescente. L'éditeur peut exporter les spécifications de cas de test vers Tessy et vers des fichiers.

Bien que le CTE soit inclus dans Tessy, son utilisation n'est pas limitée aux systèmes embarqués. Il est donc disponible comme produit séparé.



Visitez notre site Web [www.hitex.com](http://www.hitex.com)

**Hitex Allemagne**  
Greschbachstraße 12 Tel. +49-721-9628-0  
D-76229 Karlsruhe Fax +49-721-9628-149  
E-mail sales@hitex.de

**Hitex USA**  
2070 Business Center Tel. 800-45-HITEX  
Drive, Suite 280 Tel. +1-949-863-0320  
Irvine, CA 92612 Fax +1-949-863-0331  
E-mail info@hitex.com

**Hitex Royaume-Uni**  
Warwick University Tel. +44-(0)24-7669-2066  
Science Park Fax +44-(0)24-7669-2131  
GB-Coventry CV4 7EZ E-mail info@hitex.co.uk

Le seul objet de cette brochure est de vous fournir des informations générales. Nous suivons une politique de développement continu, il est donc possible que des modifications ou des améliorations techniques soient apportées. Les marques commerciales des autres sociétés utilisées dans le texte font uniquement référence aux produits de ces sociétés. Hitex, HISIM, HITOP, DProbe, JProbe, USB Agent, Tanto et Tantino sont des marques commerciales de Hitex Development Tools GmbH. Tous droits réservés © 2011.

B0-Tessy-Findd Mar 2011-001