Les nouveautés de Cantata 8.0?

Cantata 8.0, disponible depuis Juin 2018, est une nouvelle version majeure contenant de nouvelles fonctionnalités inédites ainsi qu'une interface utilisateur repensée.

Ce document couvre les changements principaux de Cantata version 8.0.

Introduction

Cantata 8.0, disponible depuis Juin 2018, est une nouvelle version majeure contenant de nouvelles fonctionnalités inédites. En effet Cantata permet désormais d'automatiser la maintenance des tests, de simplifier l'exécution de vos tests sur différentes plateformes embarquées, et d'instrumenter votre code source de manière très fine.

Cantata Version 8.0 poursuit les efforts entamés en version 7.2 pour simplifier et rationaliser l'interface graphique tout en préservant les fonctions avancées de l'outil dont les utilisateurs Cantata sont familiers.

Cantata 8.0 intègre des corrections et améliorations dont la liste complète figure dans la Release Notes du produit. Les améliorations les plus significatives sont détaillées dans les chapitres ci-dessous.

Code Change Analysis

Tester au plus tôt dans le cycle de vie du développement logiciel contribue à améliorer la qualité des systèmes et permet de réduire les efforts de reprise du code. Cependant, même les projets les mieux organisés font face à de nombreux changements du code source, et ce après le démarrage des phases de tests. Les sociétés souhaitent capitaliser leurs efforts de tests en les mettant à jour en fonction des changements du code, mais cela peut entrainer un coût significatif de maintenance des tests. Les tests unitaires et d'intégration sont très dépendants de la structure du code source, et sont par conséquent fragilisés par les changements du code. La mise à jour des tests peut nécessiter plusieurs itérations et tentatives de build afin de déterminer quels changements doivent être apportés aux tests afin de les resynchroniser avec le code.

Comment la nouvelle fonction de Cantata intitulée Code Change Analysis peut-elle réduire ce coût de maintenance des tests ?

Grâce à l'analyse des changements du code source et de leurs dépendances avec les tests :

- Comprendre dans un premier temps les changements du code testé (en comparant la dernière version du code testé avec la version courante contenant des modifications)
- En identifiant l'ensemble des scripts de test impactés par les changements
- En listant de manière synthétiques tous les changements affectant les tests
- En comprenant quels changements du code peuvent impacter la couverture de code obtenue pour les tests existants

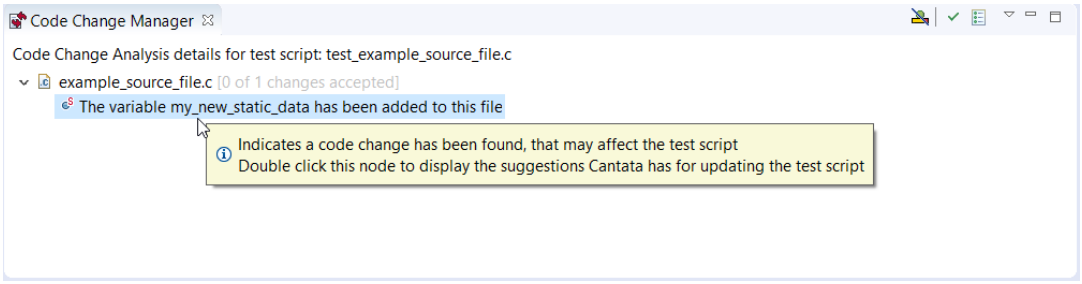
La liste des changements détectés pouvant affecter les scripts de tests comprend :

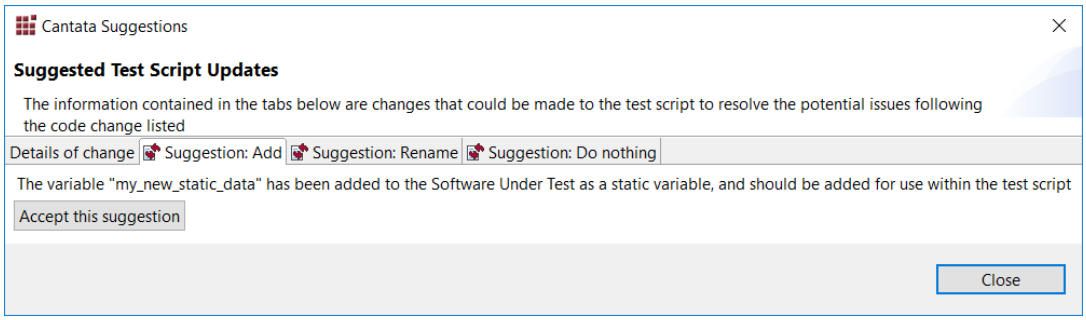
Fichiers Source	• Ajouté / effacé / renommé
Fichiers Header	• Ajouté / effacé / renommé
Fonctions & Méthodes	• Ajouté / effacé • Signature modifiée (type retourné, nommage & paramètres) • Séquence d'appel pouvant affecter les bouchons / wrappers
	• Corps de fonction ajouté / effacé • Corps de fonction modifié – avec impact sur la couverture de code

Variables	• Ajouté / effacé • Modifié (scope, type et nommage, etc)

Le développeur logiciel dispose d'une vue Code Change Manager offrant des choix de mise à jour automatique des tests permettant leur resynchronisation avec le code :

- Pour chaque changement, fournir plusieurs alternatives plausibles de mise à jour du script de test
- Procéder ensuite à la reprogrammation automatique du script de test sans effort de codage supplémentaire

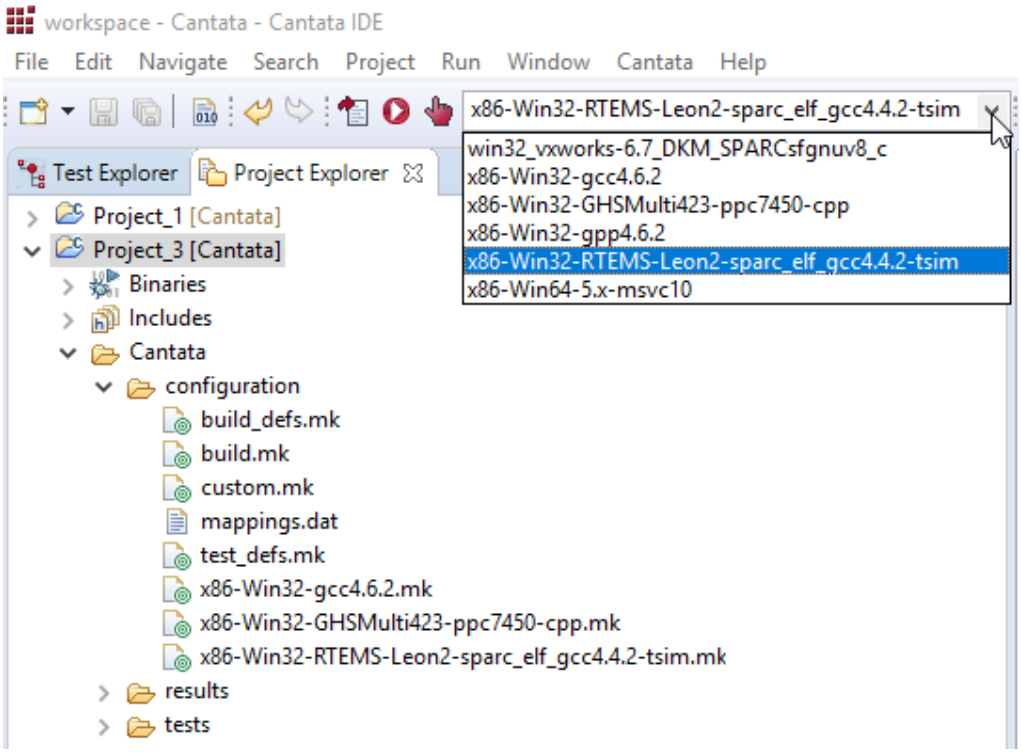




Target Deployment Switching

La nouvelle fonction de bascule entre vos différentes librairies de gestion des cibles embarquées permet de réutiliser simplement vos tests avec vos différents processeurs et chaines de compilation. Cette fonction est particulièrement intéressante si votre code est conçu pour différentes variantes d'un système s'utilisant sur différentes plateformes.

Des makefiles Cantata spécifiques sont automatiquement générés pour chaque environnement cible. La bascule entre ces makefiles se fait simplement par sélection d'une librairie de déploiement dans une liste déroulante présente au niveau de la barre d'icônes principale de l'outil.



Réutiliser les mêmes tests pour différents environnements cible accélère vos efforts de vérification, et évite de dupliquer des projets entiers de tests. L'effort de maintenance est également réduit par le fait de n'avoir qu'à gérer un seul projet de tests de référence. Les risques associés à la gestion de différentes copies de projets disparaissent.

Le fichier de résultats de tests Cantata (.ctr) contient l'enregistrement de l'environnement de déploiement pour lequel les tests ont été exécutés, les preuves de tests liées à la certification sont donc bien différentiées pour chaque environnement.

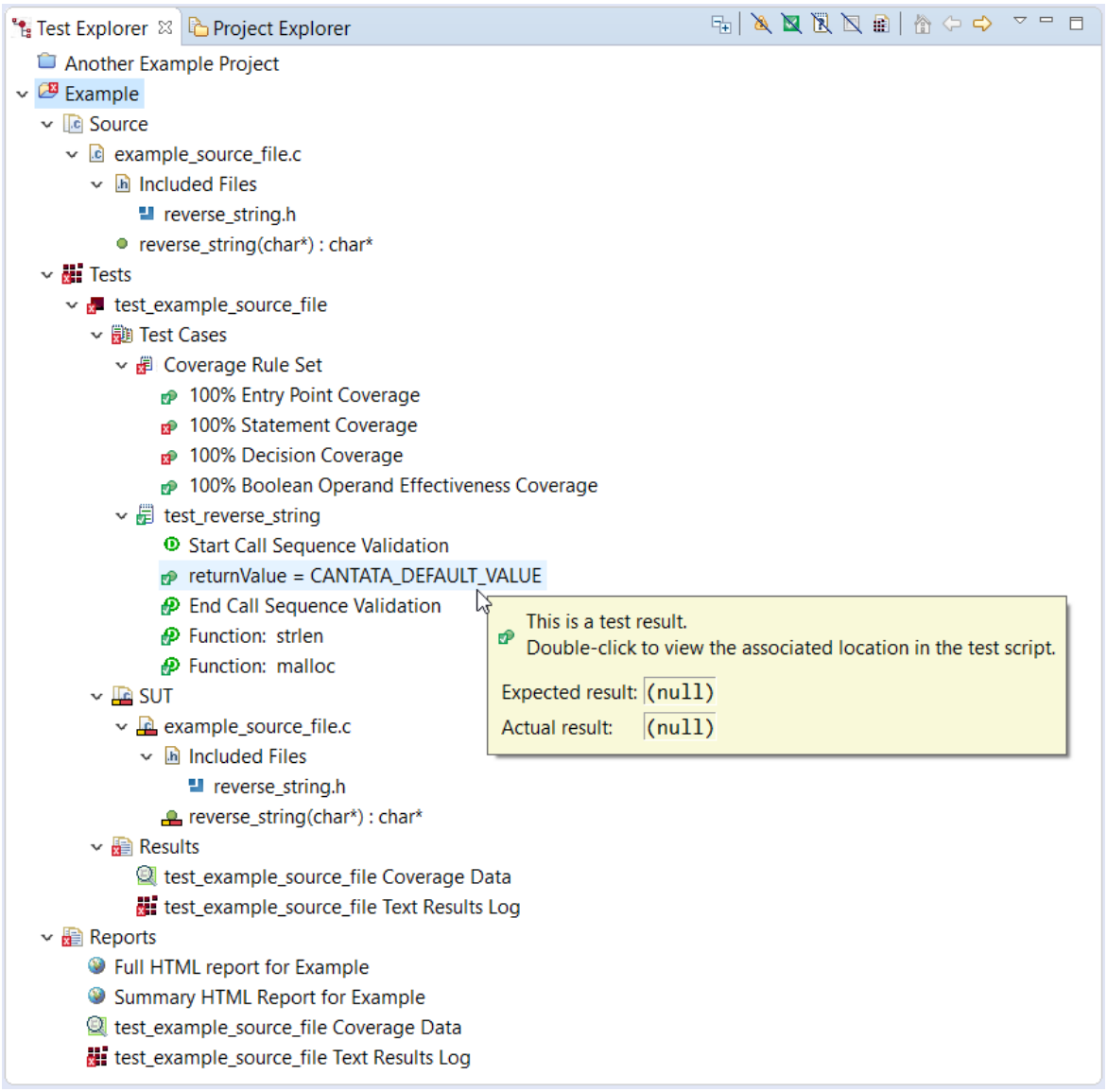
Test Explorer

Cantata 8.0 propose une nouvelle vue graphique intitulée Test Explorer afin de centraliser et simplifier l'accès et la gestion de votre code, tests, résultats et rapports. Cette vue est totalement dédiée aux tests contrairement au précédent explorateur de projet. Le rendu et la navigation sont plus synthétiques car ne proposant que les informations pertinentes aux tests gérés par Cantata.

Test Explorer est le point de départ de toutes les fonctionnalités et autres vues de Cantata, en combinant et améliorant les fonctions de Project Explorer avec les éléments de diagnostique précédemment affichés dans les vues Coverage Results et Test Results Explorer.

Les nouvelles fonctions de Test Results Explorer permettent :

- La génération et la gestion des tests Cantata tests avec des menus contextuels
- L'utilisation de filtres pour se concentrer sur les fichiers sources, tests et résultats pertinents
- L'affichage d'info-bulles explicatives proposant des résultats à différents niveaux de l'arborescence des tests

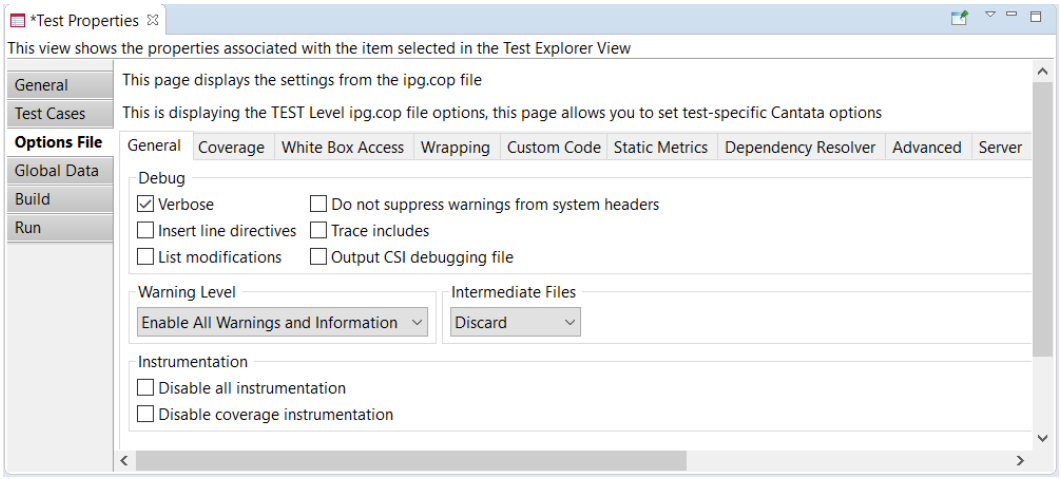


Test Properties

La nouvelle vue de l'IHM intitulée Test Properties centralise tous les éléments de contrôle de votre environnement de test. Cette vue affiche et permet de choisir les propriétés associées à chaque élément géré par Cantata depuis la vue Test Explorer. Elle permet de regrouper et améliorer les points de contrôle précédemment gérés par Test Script Manager.

La vue Test Properties permet à l'utilisateur de gérer :

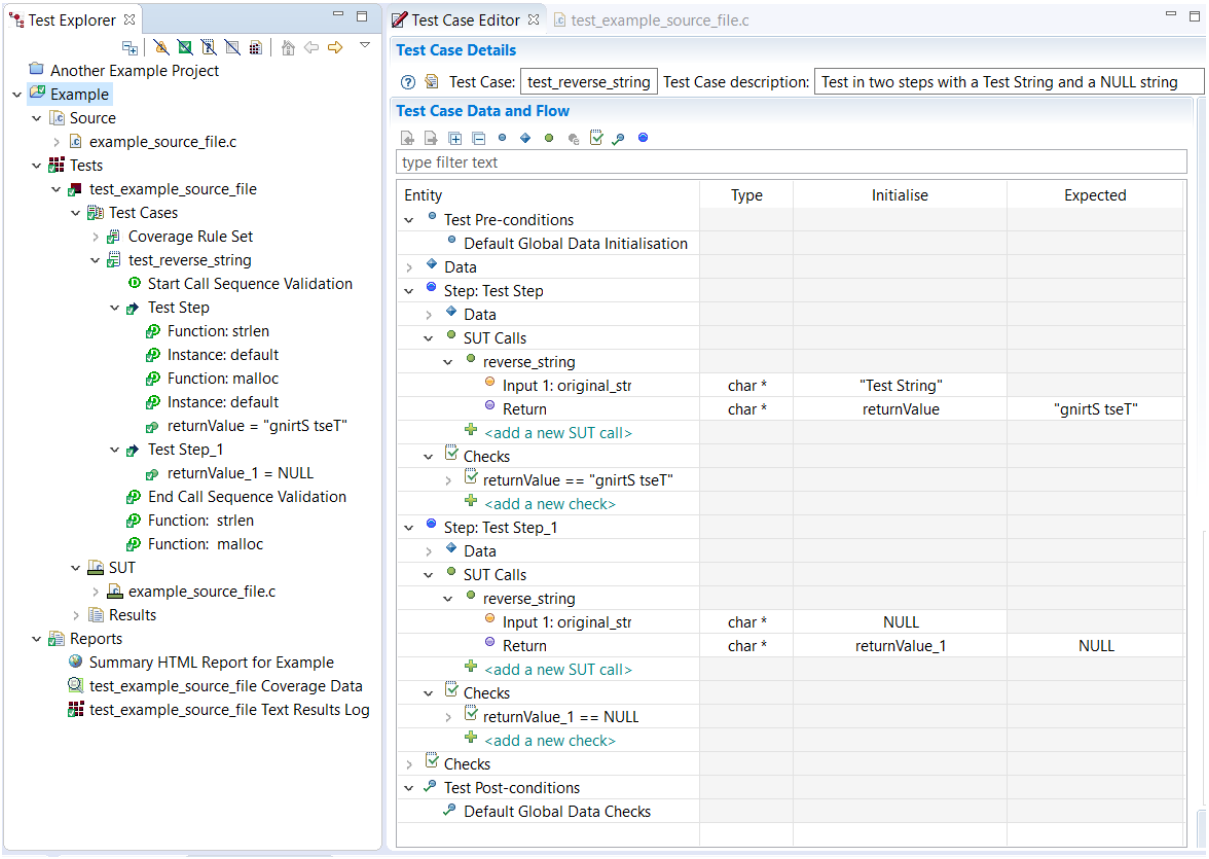
- Les options des Projets et Scripts de Test Cantata (fichiers ipg.cop)
- Les paramètres liés aux Variables Globales
- Le nommage des fonctions de test ainsi que l'insertion de commentaires additionnels
- Les paramètres de build (make file, chemin d'include, couverture de code, gestion mémoire)
- Les configurations d'exécution des tests (ainsi que l'activation/désactivation de certains tests)



Test Case Editor

La vue Test Case Editor a été améliorée en introduisant un nouvel élément appelé Test Steps, afin d'ajouter un niveau de granularité supplémentaire au sein d'un cas de test Cantata. Les cas de test peuvent désormais contenir un ou plusieurs Test Steps, contenant chacun leurs propres appels au code testé et la vérification de données. Par défaut, chaque nouveau cas de test est généré avec un Test Step initial, et l'utilisateur peut simplement en rajouter d'autres. La séquence d'ordonnancement des Test Steps peut être modifiée facilement depuis l'IHM en utilisant le drag & drop.

Cette amélioration de Test Case Editor vise à enrichir vos scenarios de tests, en permettant de tester et préserver les états intermédiaires du code appelé. Cela se révèle particulièrement utile pour le code orienté objet ou les fonctions interdépendantes gérant des états logiciels.



D'autres améliorations ont été apportées à Test Case Editor telles que : l'agencement général du panneau Call Interface Control, ainsi qu'une simplification de la vérification des variables globales.

Custom Code Injection

Il est désormais possible d'instrumenter votre code de manière fine avec des instructions supplémentaires saisies par l'utilisateur.

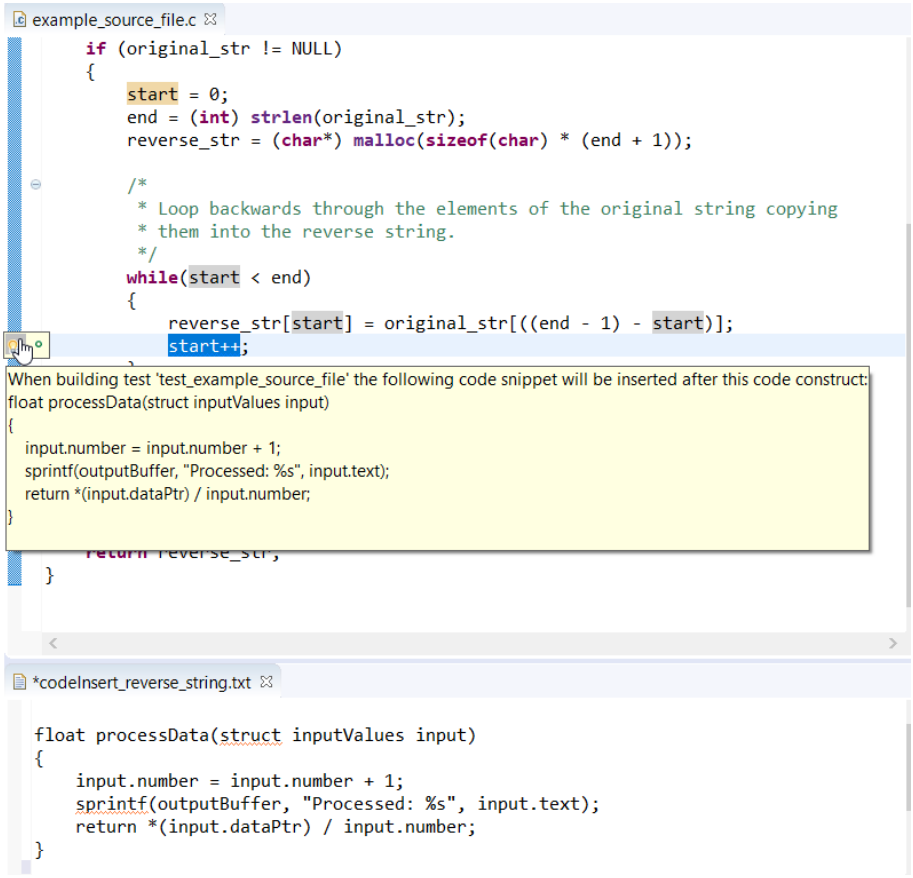
Cela permet d'éviter le recours à différentes macros de préprocesseur pour les besoins de tests, qui peuvent causer :

- Des effets de bord comportementaux au sein de votre environnement de production
- Une augmentation de la base de code en raison du code de vérification supplémentaire
- Une maintenance moins aisée du code due à une lisibilité réduite

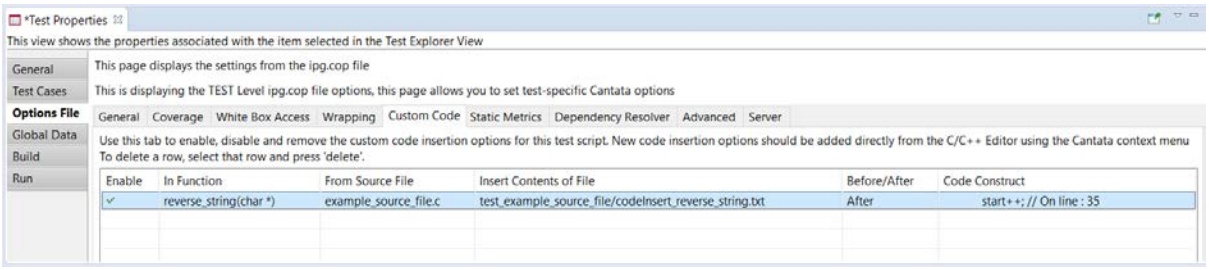
Les Wrappers Cantata offrent cette capacité également, mais uniquement au niveau des appels du code. Custom Code Injection peut être utilisé n'importe où dans le corps des fonctions afin de :

- Modifier les valeurs des variables locales et registres
- Injecter des erreurs logicielles
- Sortir des boucles infinies
- Lever des interruptions
- Appliquer un patch sans avoir à directement modifier le code de production

Custom code Injection se gère directement depuis l'éditeur C/C++ editor via un menu contextuel accessible depuis vos lignes de code. Le code injecté peut être inséré avant ou après une ligne donnée.



Le code injecté est stocké dans un fichier indépendant qui peut être réutilisé avec plusieurs scripts de test Cantata. Ce fichier est contrôlé à la fois par la vue Cantata Test Properties et l'éditeur C/C++.



Extension des plateformes supportées

Comme pour toute nouvelle version de Cantata, le support des environnements a été étendu pour la version 8.0.

Cantata est intégrable sous forme de plug-ins avec les plateformes de développement utilisant la base Eclipse®, et Cantata 8.0 propose désormais par défaut le support de la release Eclipse Neon (4.6.3), tout en livrant les plug-ins couvrant les releases Eclipse depuis Galileo (3.5) jusqu'à Oxygen (4.7), vous offrant ainsi l'accès à tout un écosystème d'outils (ALM, CI, SCM, etc.).

Le support des compilateurs GNU GCC et g++ a été étendu jusqu'aux versions 7.1 (Windows) et 7.3 (Linux).

Le support de C++ 11 et 14 a été amélioré pour couvrir des constructions du langage non disponibles dans la version 7.2 de Cantata. Si vous souhaitez obtenir la liste complète des constructions C++ 11 et 14 supportées, merci de contacter sales@qa-systems.com.