CANTATA

Quali Sono Le Novita' Di Cantata 8.0?

Cantata 8.0, disponibile da maggio 2018, è una nuova importante release che offre nuove significative funzionalità e un'interfaccia utente migliorata.

Questo documento delinea i principali cambiamenti di Cantata versione 8.0

Introduzione

Cantata 8.0, disponibile da maggio 2018, è una nuova importante release che offre nuove e significative funzionalità. Questa versione automatizza la manutenzione dei test, semplifica l'esecuzione dei test su target embedded molteplici e offre una maggiore flessibilità durante l'instrumentazione del codice.

La versione 8.0 continua anche il viaggio iniziato nella versione 7.2 per fornire un'interfaccia utente che sia accessibile e intuitiva per gli utenti alle prime armi, pur mantenendo le funzionalità avanzate che gli utenti esistenti si aspettano.

Cantata 8.0 contiene anche molti altri miglioramenti e correzioni. Il set completo di modifiche è documentato nelle note di rilascio che tracciano tutte le modifiche in Cantata dalla versione 4.1. Le modifiche più importanti sono evidenziate nelle sezioni seguenti.

Analisi delle modifiche del codice

Il test tempestivo nel ciclo di vita dello sviluppo del software consente di migliorare la qualità del sistema e riduce gli sforzi di recupero e rilavorazione del codice. Tuttavia, anche i progetti meglio organizzati devono affrontare molte modifiche nel codice sorgente, dopo le fasi di test iniziali. I test di unità e integrazione dipendono molto dalla struttura del codice sorgente e sono quindi indeboliti dalle modifiche allo stesso codice. L'aggiornamento dei test può richiedere uno sforzo di tempo eccessivo per le molte iterazioni e i continui tentativi di build, al fine di determinare quali modifiche devono essere apportate ai test per risincronizzarle con il codice.

La nuova funzionalità di analisi del codice di Cantata identifica le modifiche al codice che influenzano i test e aiuta ad automatizzare la manutenzione del test. Sono fornite indicazioni dettagliate sulla risoluzione di questi problemi e possono essere selezionati suggerimenti che attivano il refactoring automatico del test per molti tipi di modifica del codice.

Un'analisi approfondita delle modifiche al codice e delle dipendenze del test identifica:

- Modifiche al codice sorgente che influiscono sugli script di test esistenti
- Quali script di test sono interessati dalle modifiche al codice
- Modifiche al codice che potrebbero influire sulla copertura del codice precedentemente raggiunta dai test esistenti

I risultati dell'analisi delle modifiche al codice sono chiaramente identificati in un nuovo gestore dedicato di modifica del codice. Le informazioni vengono conservate quando il codice è stato testato l'ultima volta e confrontato con il codice corrente.

Le modifiche rilevate nel codice sorgente che influiscono sugli script di test esistenti includono:

File sorgenti	• aggiunti / rimossi / rinominati
File di intestazione	• aggiunti / rimossi / rinominati
Funzioni e metodi	• aggiunti / rimossi

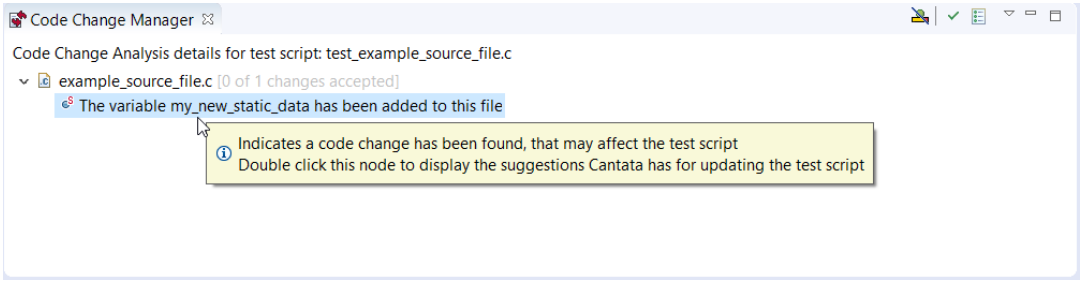
- firma modificata (scopo, tipo restituito, nome e parametri)
- Chiamata cambiata – che può influenzare gli stub / wrapper
- Corpo aggiunti / rimossi
- Corpo modificato – che potrebbe influire sulla copertura del codice

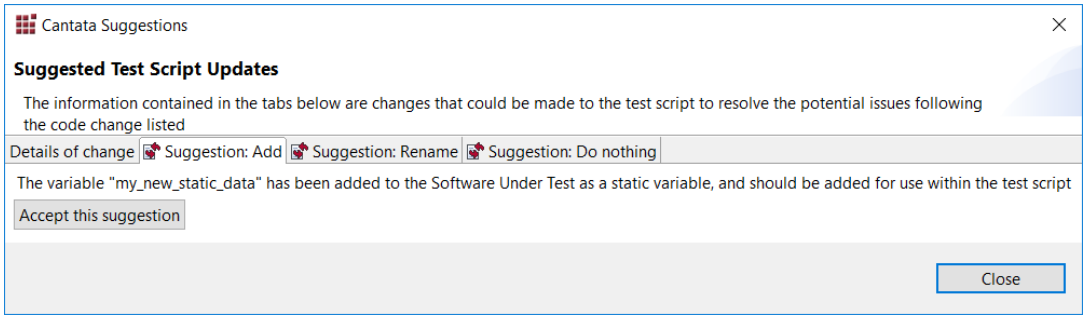
Variabili

- Aggiunte / rimosse
- Modificate (scopo, tipo e nome etc.)

Lo sviluppatore del software dispone di un gestore delle modifiche del codice che offre opzioni di aggiornamento automatico e di sincronizzazione dei test con il codice modificato:

- Per ogni modifica, fornisce diverse alternative plausibili per aggiornare lo script di test
- Fornisce suggerimenti per aggiornare il test con una guida sugli effetti dell'accettazione del suggerimento
- Procedere quindi alla riprogrammazione automatica dello script di test senza ulteriore sforzo di codifica

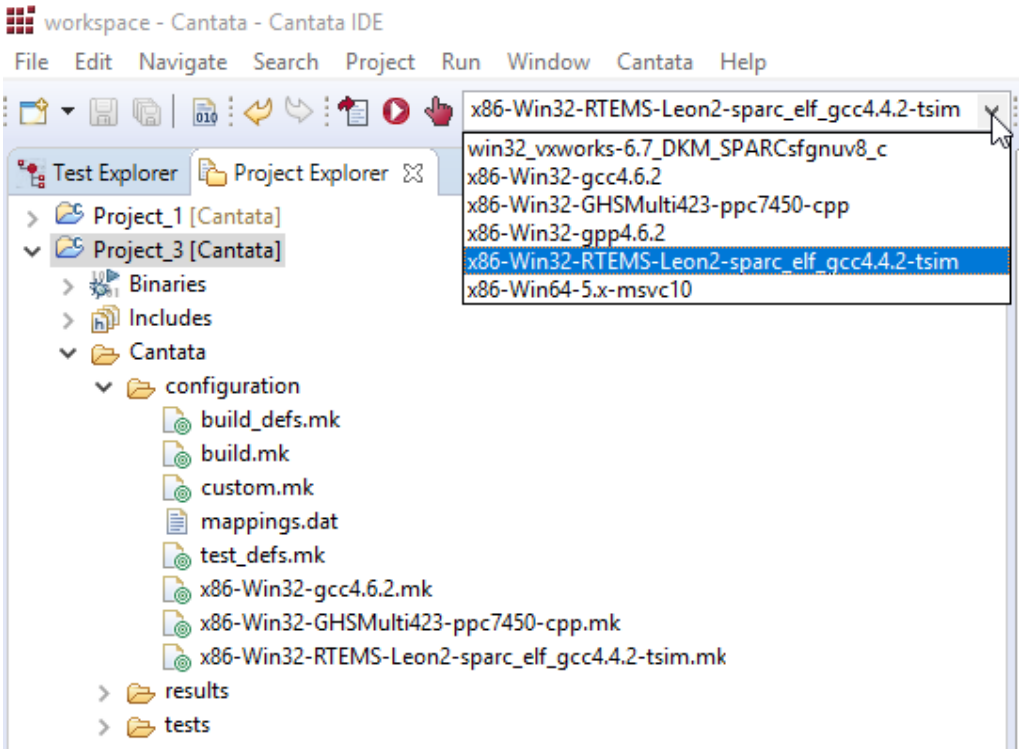




Target Deployment Switching

La nuova funzione di commutazione tra le diverse librerie di gestione dei target embedded consente facilmente di riutilizzare gli stessi test con i diversi processori e catene di compilatori. Questa funzione è particolarmente interessante se il codice è progettato per diverse varianti di un sistema che viene utilizzato su piattaforme diverse.

I makefile Cantata specifici vengono generati automaticamente per ciascun deployment. La commutazione tra questi makefile avviene semplicemente selezionando una deployment da un menù a discesa nella barra degli strumenti principale.



Riutilizzare gli stessi test con diversi deployment velocizza la verifica eseguita su ambienti diversi ed evita la duplicazione di interi progetti di test. Anche lo sforzo di manutenzione viene ridotto, dovendo solo gestire un singolo progetto di riferimento. I rischi associati alla gestione di diverse copie di progetti scompaiono. Il file dei risultati del test di Cantata (.ctr) contiene lo specifico deployment per il quale sono stati eseguiti i test, quindi le prove dei test relativi alla certificazione sono ben differenziate per ciascun ambiente.

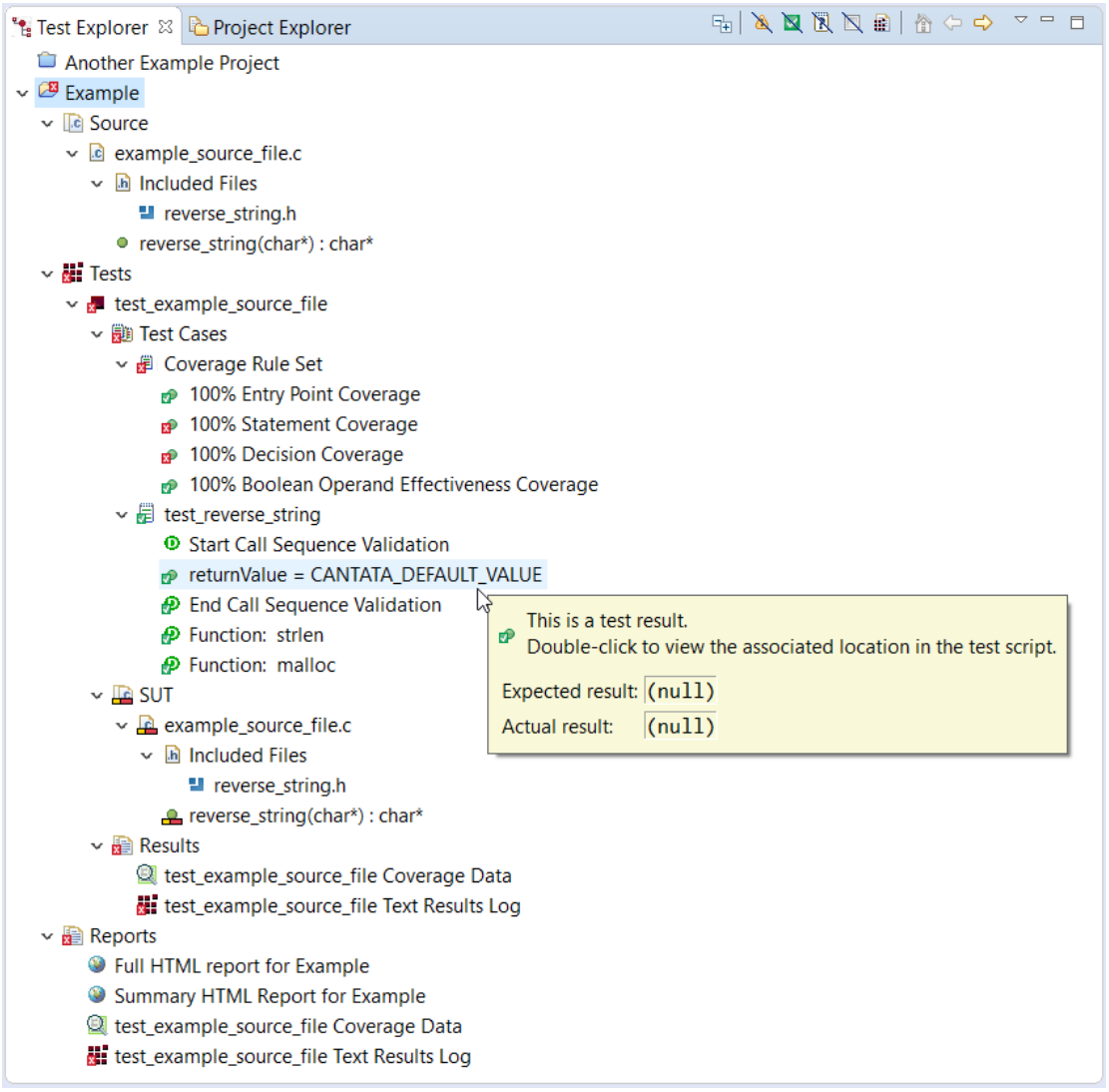
In aggiunta ai built-in target deployment integrati in Cantata per la personalizzazione dell'ambiente da parte dell'utente, man mano che tra le diverse release vengoni completati nuovi deployment, gli stessi sono resi disponibili per il download da QA Systems.

Test Explorer

Cantata 8.0 offre una nuova visualizzazione grafica denominata Test Explorer per centralizzare e semplificare l'accesso e la gestione del codice, dei test, dei risultati e dei report. Questa visione è totalmente dedicata ai test a differenza del precedente project explorer. Rendering e navigazione sono più sintetici perché forniscono solo informazioni rilevanti per i test gestiti da Cantata. Test Explorer è il punto di partenza per tutte le funzionalità e le altre viste di Cantata, combinando e migliorando le funzionalità del Project Explorer con gli elementi diagnostici visualizzati in precedenza nelle finestre Coverage Results e Test Results Explorer.

Le nuove funzionalià di Test Results Explorer consentono:

- La generazione e la gestione di test Cantata con menu contestuali
- L'uso di filtri per focalizzarsi su file sorgente, test e risultati rilevanti
- Suggerimenti degli strumenti con informazioni aggiuntive e risultati rilevanti per ciascun nodo dell'albero

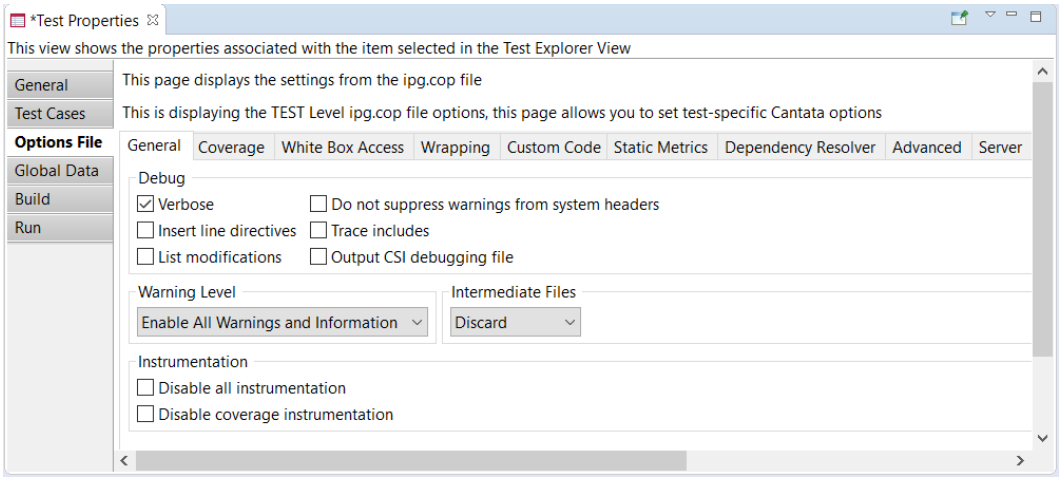


Proprietà del test

La nuova vista HMI denominata Test Properties centralizza tutti gli elementi di controllo nel proprio ambiente di test. Essa visualizza e consente di scegliere le proprietà associate a ciascun elemento gestito da Cantata in Test Explorer; permette di raggruppare e migliorare i punti di controllo precedentemente gestiti dal Test Script Manager.

Test Properties consente all'utente di gestire:

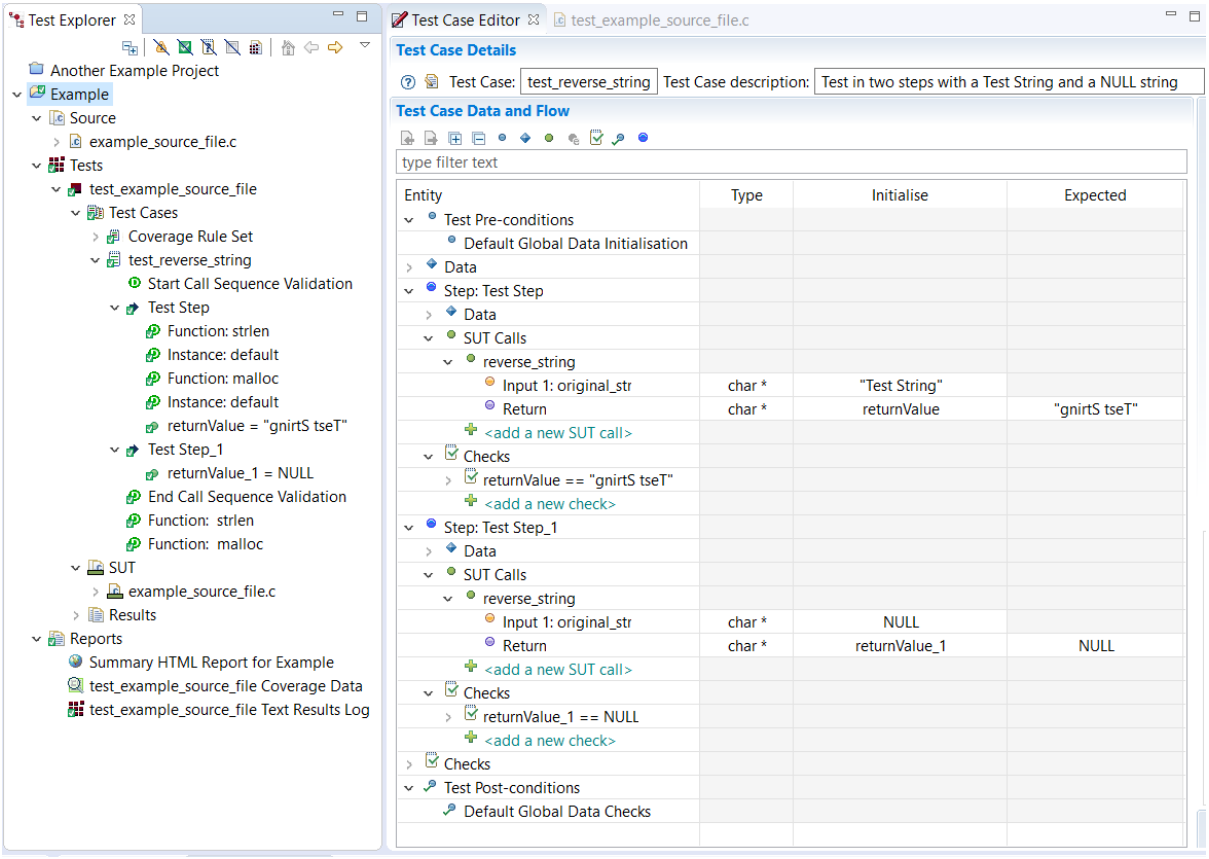
- Opzioni per progetti di test Cantata e script (file ipg.cop)
- Parametri relativi alle variabili globali
- Denominazione delle funzioni di test e inserimento di commenti aggiuntivi
- Controllo dei parametri di costruzione (make file, include path, code coverage, gestione della memoria)
- Controllo delle configurazioni di esecuzione del test (così come l'attivazione / disattivazione di alcuni test)



Editor dei Test Case

La vista del Test Case Editor è stata migliorata introducendo un nuovo elemento chiamato Test Steps, per aggiungere un ulteriore livello di granularità all'interno di un caso di test di Cantata. I test case possono ora contenere uno o più passaggi del test, ciascuno contenente le proprie chiamate al codice testato e alla verifica dei dati. Per impostazione predefinita, ogni nuovo caso di test viene generato con un Test Step iniziale e l'utente può semplicemente aggiungerne altri.

La sequenza dei Test Step può essere facilmente modificata dall'interfaccia operatore tramite drag & drop. Questo miglioramento del Test Case Editor mira ad arricchire gli scenari di test, consentendo di testare e preservare gli stati intermedi del codice chiamato. Ciò è particolarmente utile per il codice orientato agli oggetti o per le funzioni interdipendenti che supportano gli stati del software.



Altri miglioramenti sono stati apportati al Test Case Editor come: miglioramenti al layout generale del pannello Call Interface Control, nonché una semplificazione del controllo delle variabili globali.

Iniezione di codice personalizzato

Ora è possibile perfezionare il codice con istruzioni aggiuntive inserite dall'utente.

Ciò evita la necessità di ricorrere a diverse macro del preprocessore a scopo di test, che possono causare:

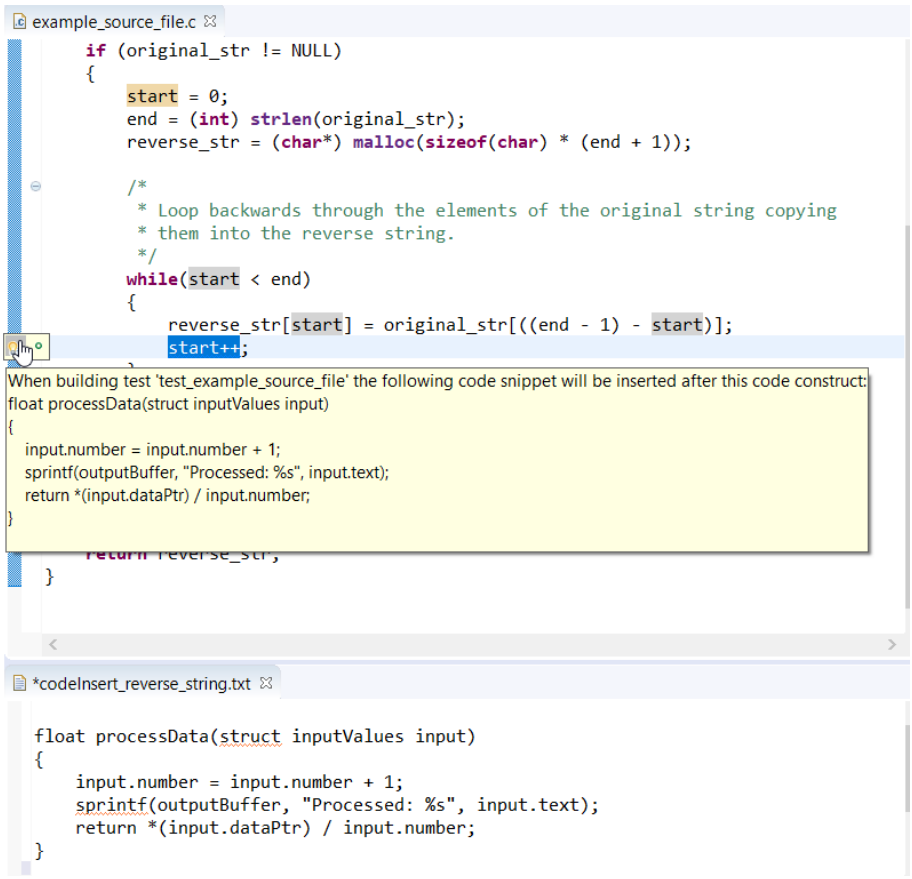
- Possibili comportamenti involontari nell'ambiente di produzione
- Un aumento della base di codice a causa del codice di verifica aggiuntivo
- Manutenzione del codice meno semplice dovuta alla ridotta leggibilità

Cantata Wrapping fornisce la capacità automatizzata di modificare il comportamento del codice, ma solo a livello di chiamate di codice, tuttavia è possibile utilizzare Custom Code Injection ovunque.

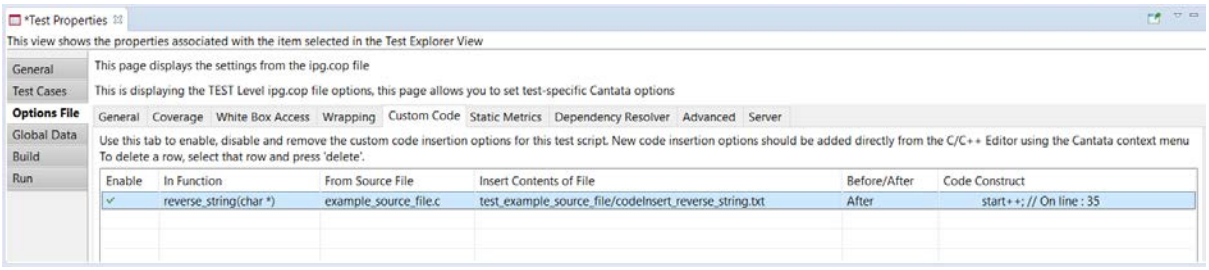
L'iniezione di codice personalizzato può essere utilizzata in qualsiasi parte del corpo delle funzioni al fine di:

- Modificare i valori delle variabili locali e dei registri
- Iniettare errori software
- Uscire da cicli infiniti
- Rimuovere interrupt
- Applicare una patch senza dover modificare direttamente il codice di produzione.

L'iniezione viene gestita direttamente dall'editor C / C ++ tramite un menu di scelta rapida accessibile dalle linee di codice. Il codice iniettato può essere inserito prima o dopo una determinata linea.



Il codice iniettato viene archiviato in un file separato e il suo utilizzo per uno o più script di test può essere controllato dalla nuova vista Cantata Test Properties e dall'editor C / C ++.



Estensione delle piattaforme supportate

Come con qualsiasi nuova versione di Cantata, il supporto degli ambienti è stato ampiamente esteso.

Cantata integra i principali IDE, ambienti di sviluppo integrati che sono Built-onEclipse® e le toolchain disponibili come plug-in Eclipse® Ready. Cantata 8.0 è costruito sulla versione Neon (4.6.3) di Eclipse ed è anche disponibile per l'installazione come set di plug-in Eclipse-Ready per le versioni Galileo (3.5) fino a Oxygen (4.7), consentendo di accedere a un intero ecosistema di strumenti (ALM, CI, SCM, ecc.).

Il supporto per i compilatori GNU GCC e g ++ è stato esteso alle versioni 7.1 (Windows) e 7.3 (Linux).

Il supporto per C ++ 11 e 14 è stato migliorato per coprire i costrutti del linguaggio non disponibili in Cantata versione 7.2. Per un elenco completo delle funzionalità supportate in C ++ 11 o 14, contattare sales@qa-systems.com.