

Couverture de code « simple » avec Visual Studio

par François-R Boyer

Ce document contient

1. Introduction à la couverture de code 1
2. Ajout d'un « outil externe » à l'interface Visual Studio..... 2
3. Configuration du projet pour la couverture de code 2
4. Recueillir et visualiser la couverture de code 3

1. Introduction à la couverture de code

Le compilateur peut trouver des erreurs de compilation, et parfois nous avertir d'erreurs possibles, mais ne peut pas détecter les erreurs de logique. Pour tester qu'un programme fonctionne, il faut minimalement au moins l'avoir exécuté une fois. Par la même idée, mais de manière plus détaillée, on peut dire que pour tester qu'une ligne de programme fonctionne, il faudrait minimalement l'avoir exécutée une fois. Donc, pour tester un programme, il est généralement souhaitable d'avoir minimalement exécuté chaque ligne au moins une fois. La *couverture de code* d'un ensemble de tests effectués sur un programme indique quelles lignes ont été couvertes par ces tests, c'est-à-dire quelles lignes ont été exécutées.

Dans ce petit exemple :

The screenshot shows a Visual Studio window with a C++ file named `ex4_1.cpp`. The code is as follows:

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      cout << "Entrer un nombre: ";
6      double nombre;
7      cin >> nombre;
8
9      if (10 < nombre && nombre < 30)
10         cout << "oui";
11     else
12         cout << "non";
13 }
14
```

Below the code editor, the 'Résultats de la couverture du code' (Code Coverage Results) window is open. It shows a table with the following data:

Hiérarchie	Non couvertes (% Lignes)	Couvertes (% Lignes)	Partiellement couvertes (% Lignes)
couverture2018-02-20_20_58_16,26.coverage	25,00 %	62,50 %	12,50 %
x.exe	25,00 %	62,50 %	12,50 %
Fonctions globales	25,00 %	62,50 %	12,50 %
main	25,00 %	62,50 %	12,50 %

The table indicates that 25% of the lines were not covered, 62.5% were covered, and 12.5% were partially covered. The code in the editor is color-coded: lines 5, 6, 7, 10, 11, and 12 are blue (covered), line 9 is yellow (partially covered), and line 10 is red (not covered).

On a exécuté le programme une seule fois, en entrant le nombre « 5 ». La couverture de code nous montre que la partie où la condition serait vraie n'a jamais été exécutée, indiqué en surligné rouge, car le nombre est inférieur à 10. La ligne de la condition est surlignée en jaune car une partie de cette ligne n'a pas été exécutée : en C/C++, dans un « && », si la partie de gauche est fausse, la partie de droite n'est pas vérifiée, donc dans le cas où le nombre est inférieur à 10, il ne vérifie pas s'il est inférieur à 30 (cette comparaison n'est pas exécutée). Les lignes en bleu ont été exécutées entièrement au moins une fois.

Le reste de ce document montre comment voir la couverture de code dans Visual Studio après une exécution simple du programme.

2. Ajout d'un « outil externe » à l'interface Visual Studio

Nous allons ajouter un élément « Couverture de code » au menu « Outil ».

Sélectionner le menu Outils > Outils externes...

Cliquer sur le bouton « Ajouter » (voir figure de gauche ci-dessous), puis entrer les informations :

Titre :

Couverture de code

Commande :

C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise\Team Tools\Dynamic Code Coverage Tools\CodeCoverage.exe

(Note: Si vous utilisez la version « Preview » de VS2019, vous pouvez remplacer « \Enterprise\ » par « \Preview\ » dans le chemin ci-dessus.)

Arguments :

collect /output:"couverture%DATE:/=-%_TIME::=%_.coverage" /session:INF1005C /verbose \$(TargetPath)

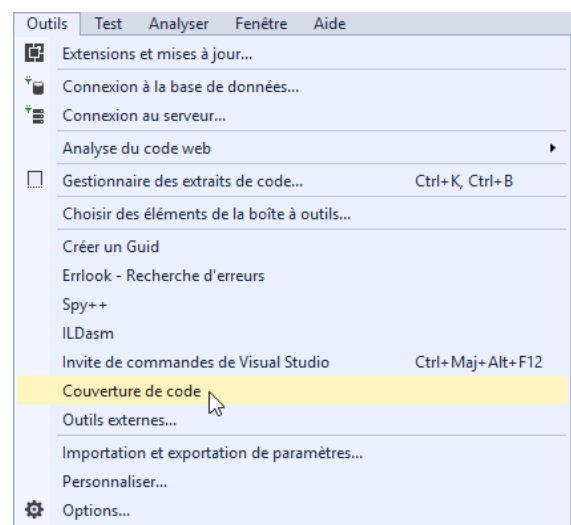
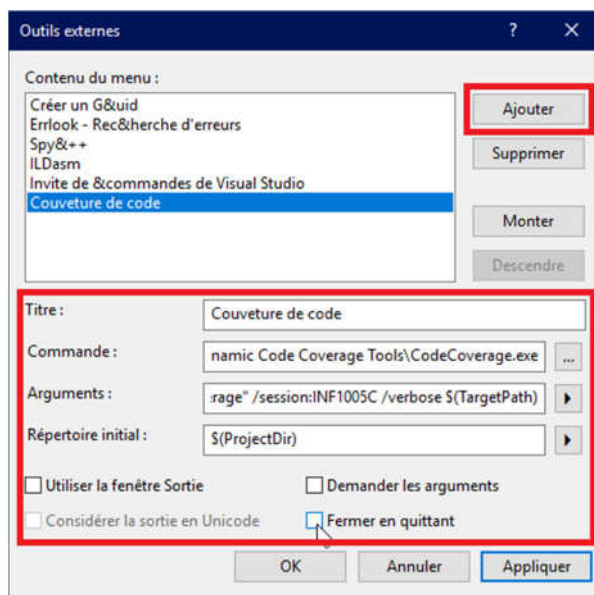
(Note: Une fois que tout fonctionne bien, vous pouvez enlever le « /verbose » pour éliminer les messages causés par cet outil, mais il n'affichera alors aucun message, même dans le cas où il y a une erreur qui l'empêche d'obtenir la couverture du code.)

Répertoire initial:

\$(ProjectDir)

Décocher « Fermer en quittant »

Puis cliquer sur « OK ». Vous devriez maintenant avoir le nouvel item dans le menu (voir figure de droite ci-dessous).



3. Configuration du projet pour la couverture de code

Note : si vous utilisez le projet fourni pour le TD6, ces opérations ont déjà été faites et vous pouvez sauter cette section.

Dans l'« Explorateur de solutions » faites un clic-droit sur le nom du projet, puis « Propriétés ».

Dans « Propriétés de configuration > C/C++ > Général », changer le « Format des informations de débogage » pour « Base de données du programme (/Zi) ». (Note : si vous aimez utiliser le mode « Modifier et Continuer » pendant le débogage de votre programme, commencez par voir la sous-section « Créer une nouvelle configuration de projet » ci-dessous avant de faire cette modification.)

Dans « Propriétés de configuration > Éditeur de liens > Général », changer l'« Activation des liens incrémentiels » par « Non ».

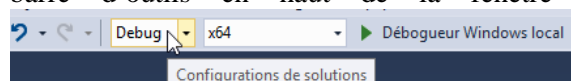
Dans « Propriétés de configuration > Éditeur de liens > Avance », changer le « Profil » pour « Oui (/PROFILE) ».

Cliquer sur « OK ».

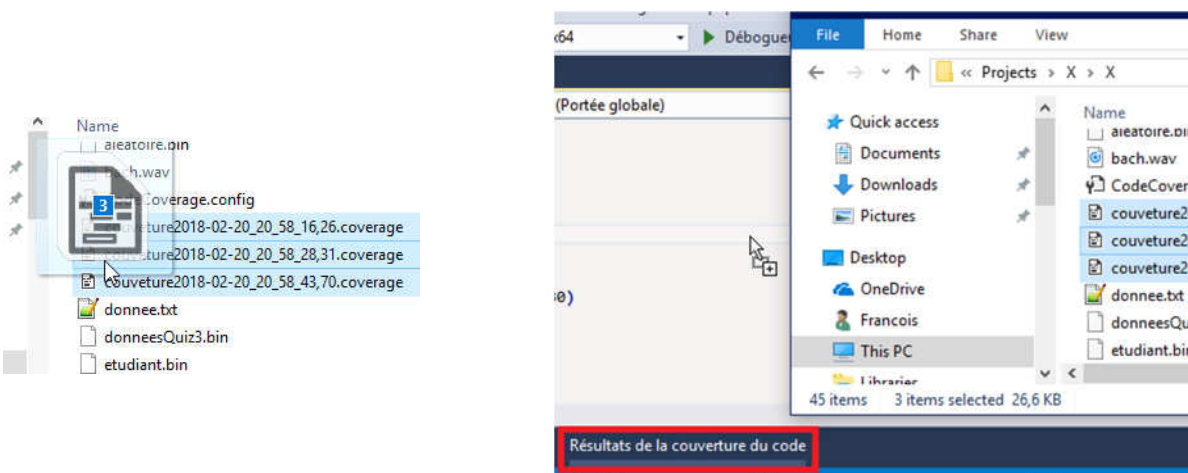
Créer une nouvelle configuration de projet : Dans le cas où vous voulez pouvoir utiliser le mode « Modifier et Continuer » pendant le débogage, vous devez utiliser des options différentes pendant le débogage et pendant la vérification de la couverture de code. Pour créer une nouvelle configuration du projet, c'est-à-dire donner un nom à un ensemble d'options, dans la fenêtre des propriétés du projet, cliquer sur « Générer > Gestionnaire de configurations... ». Comme « Configuration de la solution active : », choisir « <Nouveau...> », lui donner un nom comme « Couverture », choisir de « Copier les paramètres de : » « Debug », laisser coché « Créer des configurations de projet », puis « OK ». Et « Fermer » le gestionnaire de configuration. Dans la fenêtre des propriétés du projet, choisir la « Configuration : » (en haut complètement) ayant le nom que vous venez de créer, avant de modifier les paramètres tel qu'indiqué ci-dessus.

4. Recueillir et visualiser la couverture de code

Une fois le projet bien configuré et l'outil ajouté au menu, il faut s'assurer de sélectionner la bonne configuration (dans la barre d'outils en haut de la fenêtre de Visual Studio) que vous avez modifiée à la section 3,



s'assurer que le programme est bien compilé (F7), puis utiliser le menu « Outils > Couverture de code », une ou plusieurs fois, pour générer des fichiers couverture*.coverage (où * est une date et heure d'exécution du programme) dans le même répertoire que le projet. Dans l'explorateur de fichiers de Windows, vous pouvez sélectionner tous les fichiers de couverture voulus (avec un rectangle de sélection, ou en cliquant sur le premier fichier et shift-clic sur le dernier) puis glisser-déposer ces fichiers dans la fenêtre de Visual Studio. Vous verrez une sous-fenêtre « Résultats de la couverture de code » (si votre fenêtre est configurée pour minimiser automatiquement les sous-fenêtres, son nom sera indiqué sans qu'elle s'ouvre; cliquer sur son nom pour l'ouvrir).




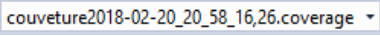

En cas où elle ne s'ouvrirait pas, on peut y accéder par le menu « Test > Fenêtres > Résultats de la couverture du code ». Par défaut, la fenêtre ressemble à ceci :

Résultats de la couverture du code				
couverture2018-02-20_20_58_16,26.coverage				
Hiérarchie	Non couverts (blocs)	Non couverts (% blocs)	Couverts (blocs)	Couverts (% blocs)
<ul style="list-style-type: none"> couverture2018-02-20_20_58_16,26.coverage 	3	33,33 %	6	66,67 %

La couverture est en termes de « blocs » plutôt que de lignes; un bloc étant une séquence d'instructions machine sans saut (chapitre 1 « Jeu d'instructions élémentaires »), sauf possiblement la dernière instruction de la séquence. On ne vous demande pas dans ce cours de comprendre comment seront générés ces blocs à partir du programme C++, et vous devriez donc changer les colonnes du tableau pour afficher en « % lignes ».

Faire un clic droit sur un titre de colonne et sélectionner « Ajouter/Supprimer des colonnes... », puis cocher les % lignes couvertes, partiellement couvertes et non couvertes, et décocher les colonnes que vous ne voulez pas.

La flèche ▸ dans la colonne « Hiérarchie » permet d'ouvrir cette hiérarchie (on peut aussi double cliquer sur l'élément plutôt que la flèche) et éventuellement voir les noms des fonctions. Double cliquer sur une fonction pour la voir et afficher sa couverture en surlignage coloré. Le bouton  permet aussi d'activer/désactiver la coloration.

Si vous avez ouvert plusieurs fichiers de couverture, il est possible de passer d'un à l'autre en le choisissant dans la sélection  couvverture2018-02-20_20_58_16,26.coverage ▾, mais ce qu'on veut généralement faire est de combiner les couvertures. Cliquer sur le bouton ; dans la fenêtre pour « Fusionner des séries de tests », cliquer sur le nom du premier fichier, shift-cliquer sur le nom du dernier, cocher sa case à gauche (ceci permet de cocher tous les fichiers d'un coup), puis « OK ».

Après fusion des couvertures de vos différentes exécutions, il serait bon d'avoir 0% de lignes non couvertes; dans ce cours, c'est correct d'avoir des lignes partiellement couvertes.