
UTILISATION DE L'AUTOVALIDATION

S2-APP4

[v.99]

D. Palao Munoz

C.-A. Brunet

Étape 1 : Récupérer l'archive d'autovalidation

1. Télécharger l'archive de l'autovalidation, un fichier zip, à partir de la page web de l'unité.
2. Extraire les documents de l'archive dans le même dossier que vos fichiers de code (fichiers .c et .h) de la problématique.
3. Vérifier que les nouveaux éléments suivants sont bien présents :
 - `libAutoValidation.a` : la librairie d'autovalidation.
 - `AutoValidation.h` : la définition des fonctions d'autovalidation.
 - `autovalidation.pdf` : ce fichier.
 - `readme.txt` : fichier qui contient la commande à mettre dans Geany à l'étape 3.
 - Le dossier `images` qui contient des images de tests pour l'autovalidation.

Étape 2 : Utiliser l'autovalidation

Il faut modifier le fichier `gestion_images.c` de la manière suivante (voir l'exemple de code un peu plus loin) :

1. Inclure le fichier `AutoValidation.h` avec la directive `#include`. (ligne 3 de l'exemple)
2. Ajouter l'appel à la fonction `AutoValidation` dans votre `main`. (ligne 6 de l'exemple)
3. Si désiré, mettre en commentaire le reste du code du `main`.

Il est possible de ne pas tester toutes les fonctionnalités d'un seul coup et de tester individuellement chaque opération. Il suffit alors de ne pas appeler la fonction `AutoValidation` dans le `main`, mais plutôt d'appeler les fonctions de tests individuels. Les noms de ces fonctions sont donnés dans le fichier `AutoValidation.h`.

```
#include <stdio.h>
#include "bibliotheque_images.h"
#include "AutoValidation.h"

int main() {
    AutoValidation();
    // Pour tester individuellement, ne pas appeler AutoValidation(), mais
    // appeler des fonctions comme test_pgm_lire() et test_pgm_ecrire().
    // Consultez Autovalidation.h.
    return 0;
}
```

Étape 3 : Ajuster la commande de construction de Geany

1. Cliquer sur la flèche pour ouvrir le menu déroulant de Construire et choisir *Définir les commandes de construction*. Ceci ouvre le dialogue suivant :

Définir les commandes de construction

#	Étiquette	Commande	Dossier de travail	Remettre à zéro
Commandes pour C				
1.	Compile	gcc -Wall -c "%f"		
2.	Build	gcc -Wall -o "%e" "%f"		
3.	Lint	cppcheck --language=c --enable=		
		Expression régulière pour les erreurs :		
Commandes indépendantes				
1.	Make	make		
2.	Make Custom Target...	make		
3.	Make Object	make %e.o		
4.				
		Expression régulière pour les erreurs :		
<i>Note : le deuxième élément ouvre une boîte de dialogue et ajoute la réponse à la commande.</i>				
Commandes d'exécution				
1.	Exécute	"./%e"		
2.				
%d, %e, %f, %p et %l sont remplacés dans les dossiers et les commandes, voir le manuel pour plus de détails.				
		Annuler Valider		

- Changer l'instruction associée à Construire (*Build*) afin d'inclure la librairie d'autovalidation. La nouvelle instruction est donnée dans le fichier `readme.txt`, un des fichiers extraits lors de l'étape 1. Cliquer sur Valider pour terminer.

Définir les commandes de construction

#	Étiquette	Commande	Dossier de travail	Remettre à zéro
Commandes pour C				
1.	Compile	gcc -Wall -c "%f"		
2.	Build	gcc -static gestion_images.c bibli		
3.	Lint	cppcheck --language=c --enable=		
		Expression régulière pour les erreurs :		
Commandes indépendantes				
1.	Make	make		
2.	Make Custom Target...	make		
3.	Make Object	make %e.o		
4.				
		Expression régulière pour les erreurs :		
<i>Note : le deuxième élément ouvre une boîte de dialogue et ajoute la réponse à la commande.</i>				
Commandes d'exécution				
1.	Exécute	"./%e"		
2.				
%d, %e, %f, %p et %l sont remplacés dans les dossiers et les commandes, voir le manuel pour plus de détails.				
		Annuler Valider		

Étape 4 : Construire et exécuter

- Cliquer sur Construire et vérifier que tout compile sans erreurs. Une erreur courante est la signature de vos fonctions qui ne respecte pas le format exigé dans le guide étudiant. Si c'est le cas, il faut les corriger et reconstruire.
- Lancer l'exécution comme d'habitude en cliquant sur Exécuter. Si l'exécution de l'autovalidation se termine normalement, il y aura un fichier nommé `validation.log` qui est généré. Il est normal que ce fichier ne soit pas lisible dans un éditeur de texte. Votre version finale de ce fichier fait partie des livrables. Consultez la section des livrables dans le guide étudiant à ce sujet.