# **Nouveautés en C++23**

# **std::expected**

**C’est quoi :**

Dans c++ 23 ils a une nouvelle fonctionnalité qui est la classe modèle expected qui permet de mieux gérer les erreurs et rend notre code plus lisible. À la place d’utiliser des exceptions, d’arrêter le programme ou d’utiliser d’autre méthode de la classe std on peut utiliser expected. En utilisant Expected, on peut gérer les erreurs tout en continuant l’exécution du programme. De plus Expected nous donne plus d’information sur l’erreur rencontrer et est optimiser donc ne ralenti pas l’application. Expected est similaires à Either dans Haskell ou result dans Rust qui tout permette de gérer des résultats d’opération.

**Composition** de ***std::expected<T, E> :***

Tout d’abord, expected a deux paramètres soit **T** qui est le type de la valeur attendue et **E** qui est le type d’erreur.**s****td::exepected** a aussi un constructeur qui va nous permettre de construire des objets (valide ou erreur). std::exepected contient aussi les méthodes ***has\_value()*** qui est une méthode qui retourne true si l’objet est valide ou false, ***value()***qui retourne la valeur de l’objet, et ***error()*** qui retourne l’erreur de l’objet.

**Où l’utiliser :**

On peut utiliser expected quand on a des fonctions ou des méthodes qui peuvent échouer et produire des erreurs par exemple quand on décide de lire un fichier.

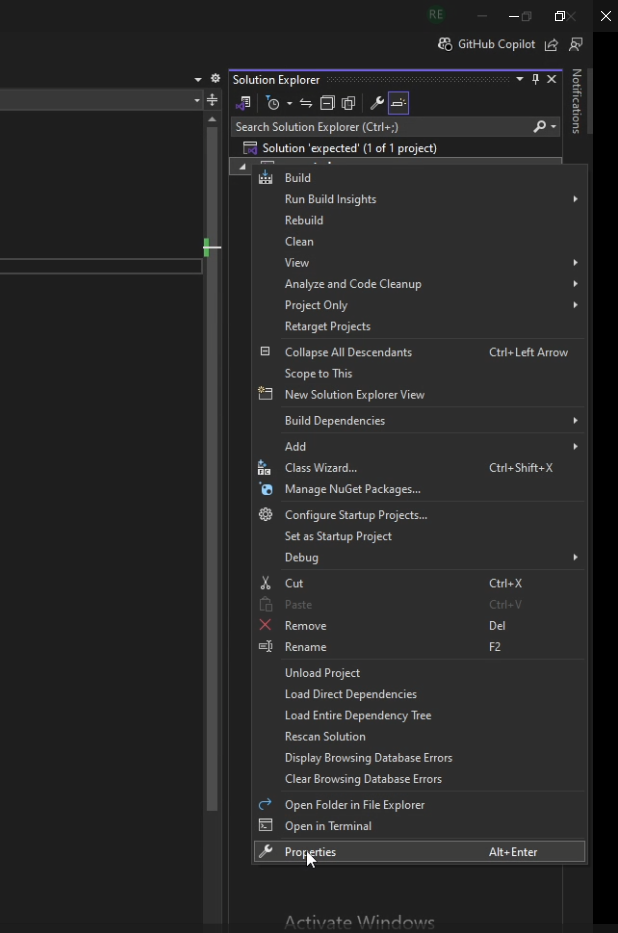
On peut utiliser dans des processus complexe incluant plusieurs étapes par exemple une application qui permet de créer des utilisateurs.

**Comment l’utiliser :**

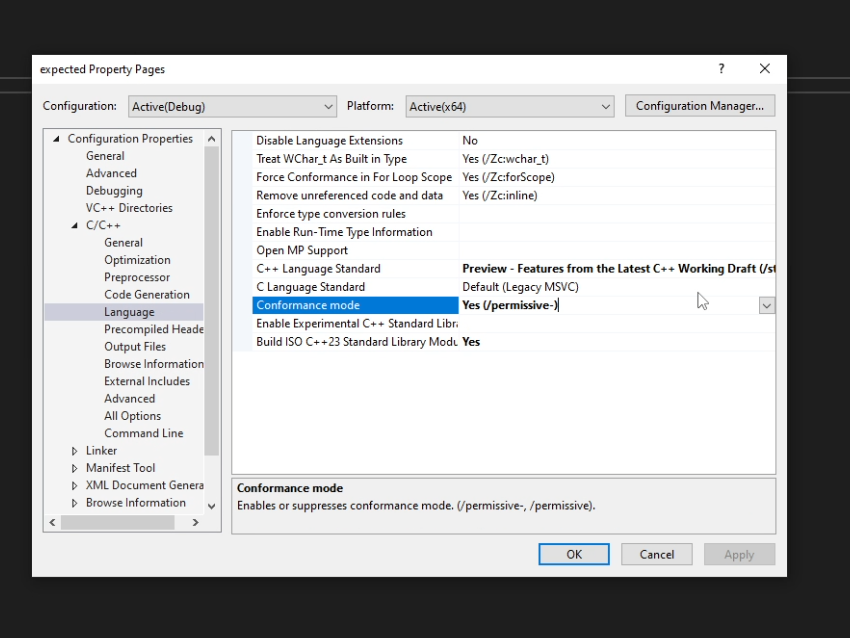
**1. Passer En C++23**

1.1 Vérifier que notre version de c++ est bien la dernière sortie

Clic droit sur le projet puis aller dans properties



1.2Dans configuration Properties Aller dans l’Onglet C/C++

1.3

Dans C++ Language Standard Sélectionner soit ISO C++23 ou Features from the latest c++

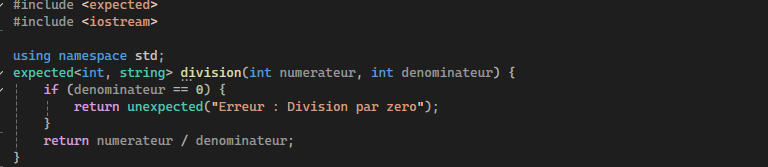
Build ISO C++ 23 Standard Library Models doit être YES.

**2. Coder**

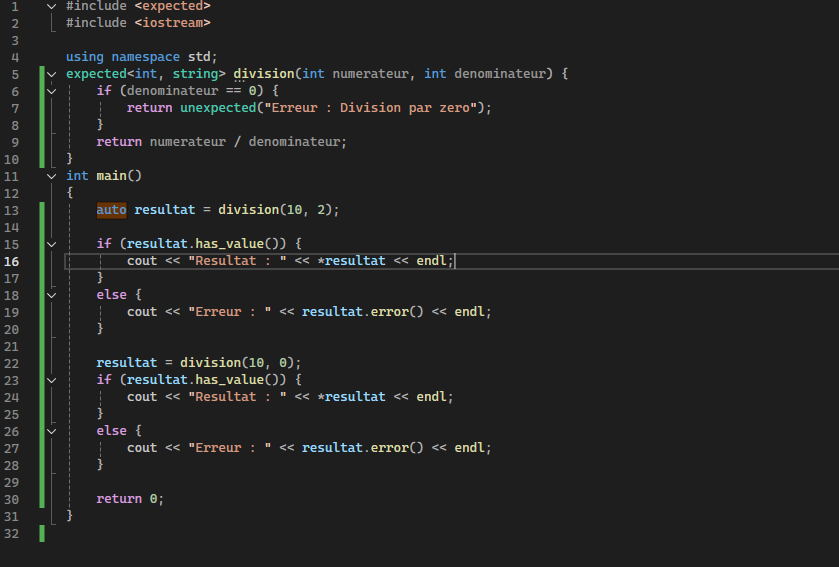
1.Tout d’abord l’inclure dans le projet :



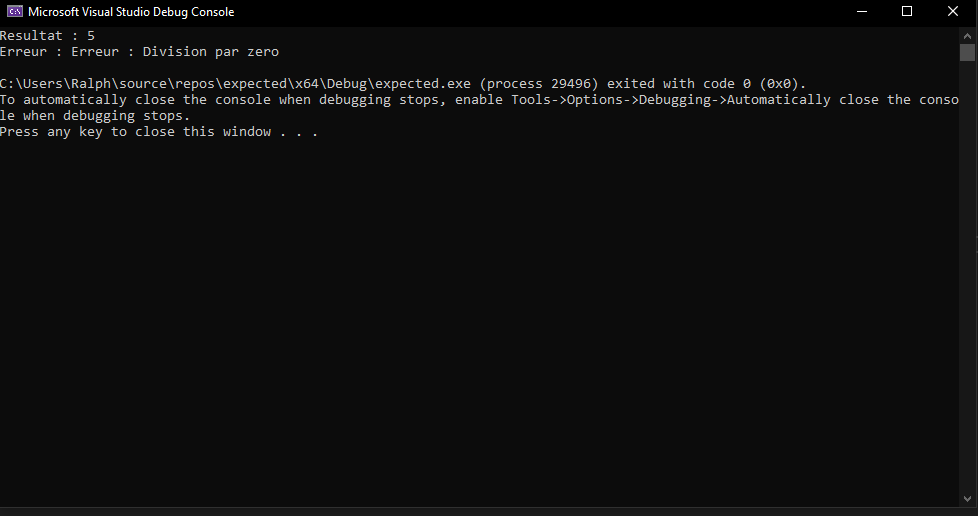
2. créer la fonction division en incluant expected<int, string> devant



3. créer le main (utiliser auto afin de déduire la variable que la fonction division car elle peut renvoyer un int mais aussi un string)



Résultat :

  
Ressource : <https://www.cppstories.com/2024/expected-cpp23/>

<https://en.cppreference.com/w/cpp/utility/expected>