

Warnungen an- und abschalten

Der Foliensatz gibt Antworten zu ein paar Problemen, die in Zusammenhang mit Visual Studio während der vergangenen Semester aufgetreten sind

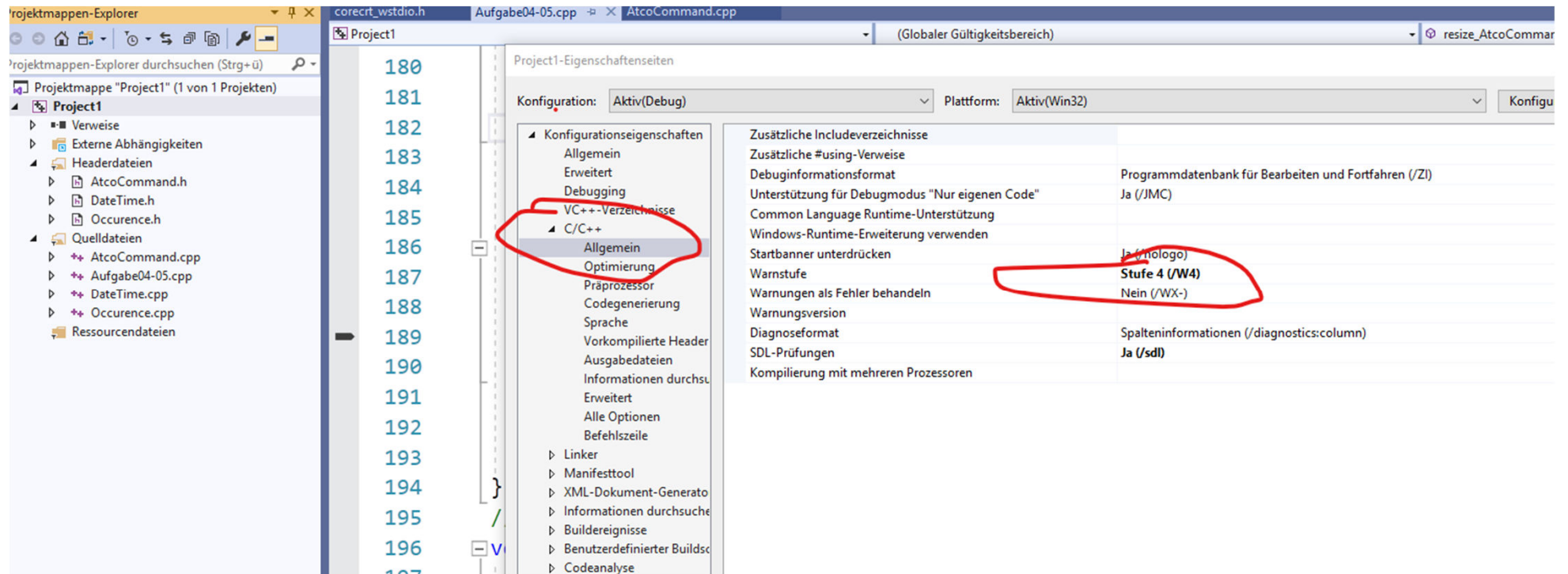
Inhalt

Der Foliensatz gibt Antworten zu ein paar Problemen, die in Zusammenhang mit Visual Studio während der vergangenen Semester aufgetreten sind.

Themen sind z.B.:

- Compilerwarnungen an- und abschalten
 - Warnung zur Option D9035 "Gm"
 - Warnung D8016: Die Optionen „/ZI und /Gy- sind inkompatibel,,
 - Warnung Compiler-Warnung C6262
- Konsolen-Fenster am Programmende NICHT schließen
- C++-Sprach-Standard einstellen (C++-20, C++-22 etc.)

Warnungs-Level erhöhen



Das Projekt (hier Project1) mit rechter Maustaste anklicken
Eigenschaften und dann C/C++ → Allgemein und Warnstufe von Stufe 3 auf
Stufe 4 setzen, nicht all wählen, dann gibt es zu viele

Bestimmte Warnungen abschalten

```
#if _MSC_VER > 1900 /* if Visual Studio 2019 or higher */  
/* Nicht dem Standard entsprechende Erweiterung: "Argument":  
   Konvertierung von "std::string" in "std::string &"/  
# pragma warning( disable : 4239 )  
#endif
```

Die Warnung 4239 wird für die Übersetzung dieser Datei ab dieser Codestelle abgeschaltet.

Die Klammerung mit `#ifdef` oder hier mit `#if` sorgt dafür, dass diese Pragma nicht jeder Compiler beachtet.

Bestimmte Warnungen als Error ansehen

// Report warning C4164 as an error.

pragma warning(error : 4164)

// Report warning C4164 as an error.

pragma warning(error : 164)

// Warnings starts with a 4

// Report warning C4064 as an error.

pragma warning(error : 64)

Bestimmte Warnungen nur einmal melden

// Issue warning C4385 only once.

#pragma warning(once : 4385)

#pragma warning(disable : 4507 34; once : 4385; error : 164)

Entspricht

// Disable warning messages 4507 and 4034.

#pragma warning(disable : 4507 34)

// Issue warning C4385 only once.

#pragma warning(once : 4385)

// Report warning C4164 as an error.

#pragma warning(error : 164)

Bestimmte Warnungen kurzzeitig abschalten

```
#pragma warning(disable:4700)
```

```
void Test() {  
    int x;  
    int y = x; // no C4700 here  
    #pragma warning(default:4700) // C4700 enabled after Test ends  
}
```

```
int main() {  
    int x;  
    int y = x; // C4700  
}
```

Bestimmte Warnungen kurzzeitig abschalten

```
#pragma warning( push )
```

```
#pragma warning( disable : 4705 )
```

```
#pragma warning( disable : 4706 )
```

```
#pragma warning( disable : 4707 )
```

```
// Some code
```

```
#pragma warning( pop )
```


Bestimmte Warnungen kurzzeitig abschalten

```
#pragma warning( push )  
#pragma warning( disable : 4705 )  
#pragma warning( disable : 4706 )  
#pragma warning( disable : 4707 )  
// Some code  
#pragma warning( pop )
```

Am Ende dieses Codes stellt *Pop* den Status jeder Warnung (einschließlich 4705, 4706 und 4707) an dem Anfang des Codes wieder her.

Siehe auch

<https://docs.microsoft.com/de-de/cpp/preprocessor/warning?view=msvc-160>

Bestimmte Warnungen kurzzeitig abschalten

```
#pragma warning( push, 3 )
```

```
// Declarations/definitions
```

```
#pragma warning( pop )
```

Wenn Sie Header Dateien schreiben, können Sie mithilfe von Push und Pop sicherstellen, dass von einem Benutzer vorgenommene Warnzustandsänderungen nicht verhindern, dass die Header ordnungsgemäß kompiliert werden. Verwenden Sie Push am Anfang des Headers und des Popups am Ende. Wenn Sie z. B. über einen Header verfügen, der nicht ordnungsgemäß auf Warnstufe 4 kompiliert, ändert der folgende Code die Warnstufe in 3 und stellt dann die ursprüngliche Warnstufe am Ende der Kopfzeile wieder her.

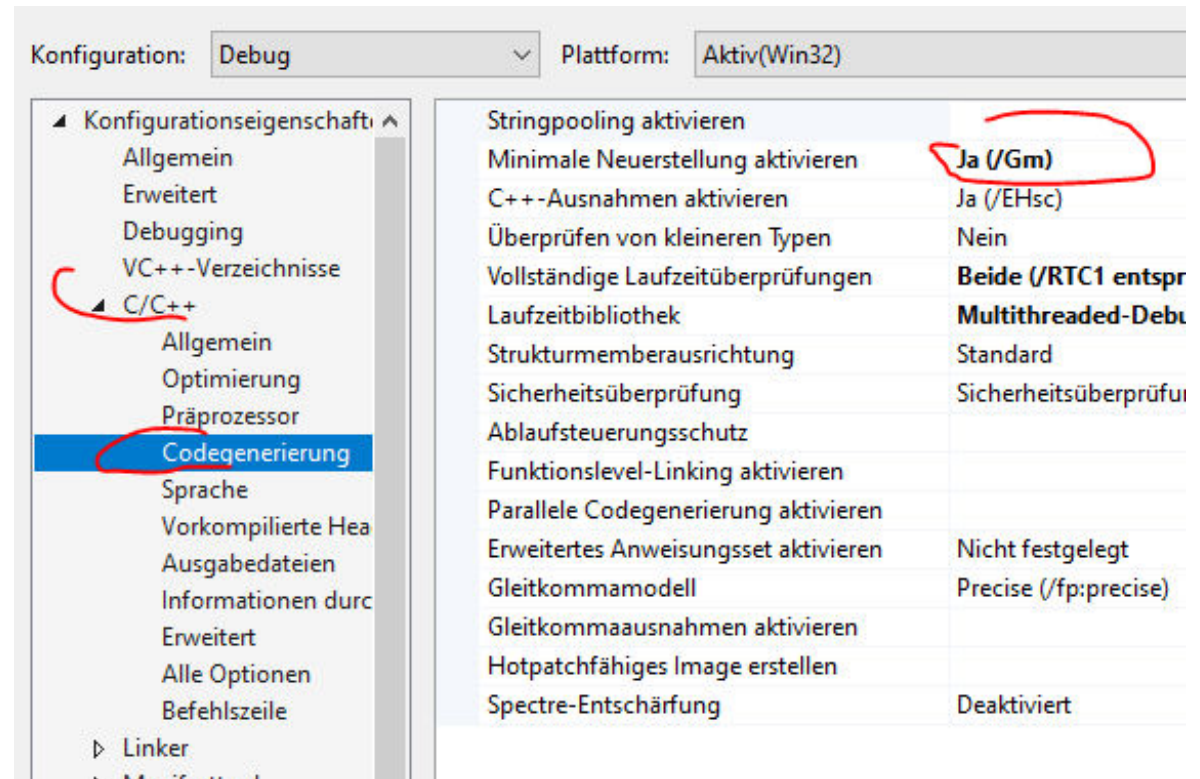
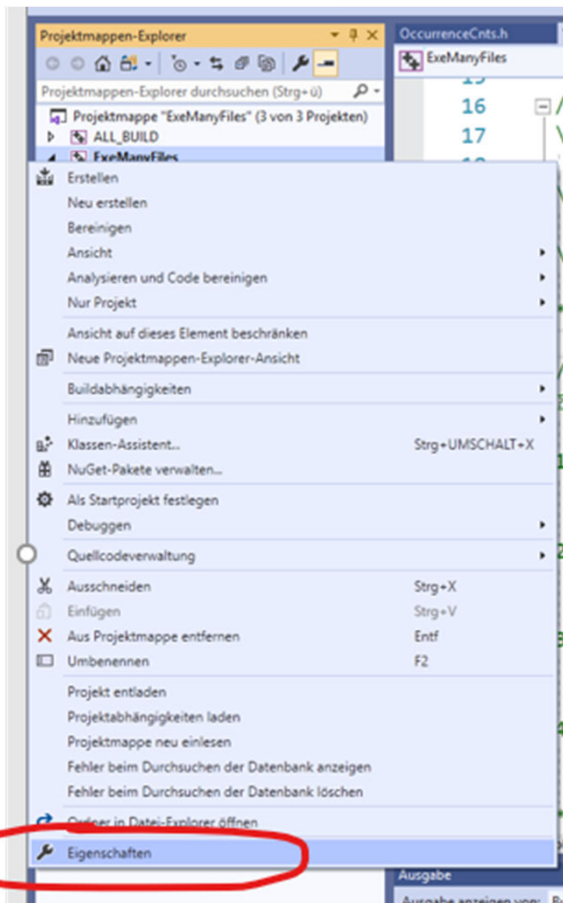
Siehe auch

<https://docs.microsoft.com/de-de/cpp/preprocessor/warning?view=msvc-160>

D9035: Die Option "Gm"

Fehler-Meldung der Gm Option

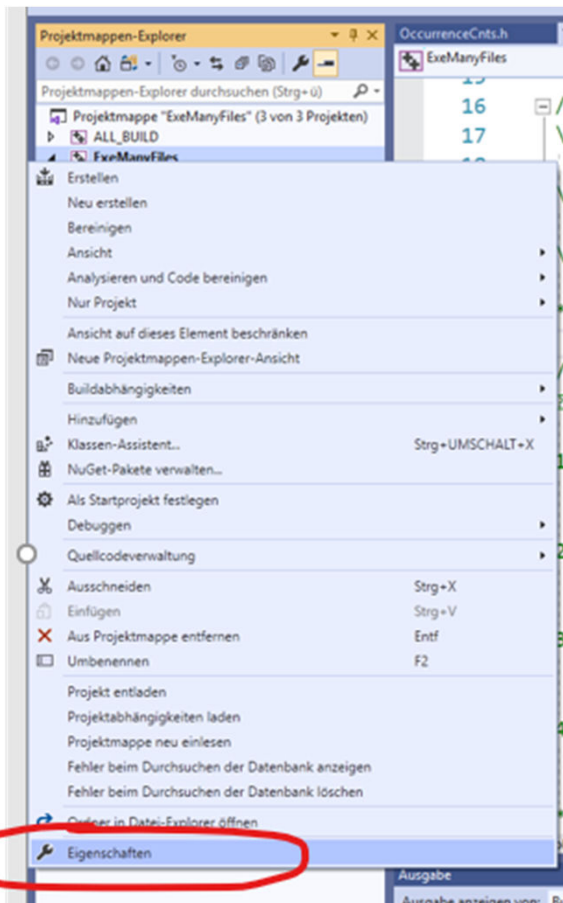
befehlszeile warning D9035: Die Option "Gm" ist veraltet und wird in einer der nächsten Versionen entfernt.



Eintrag komplett entfernen

Fehler-Meldung der Gm Option

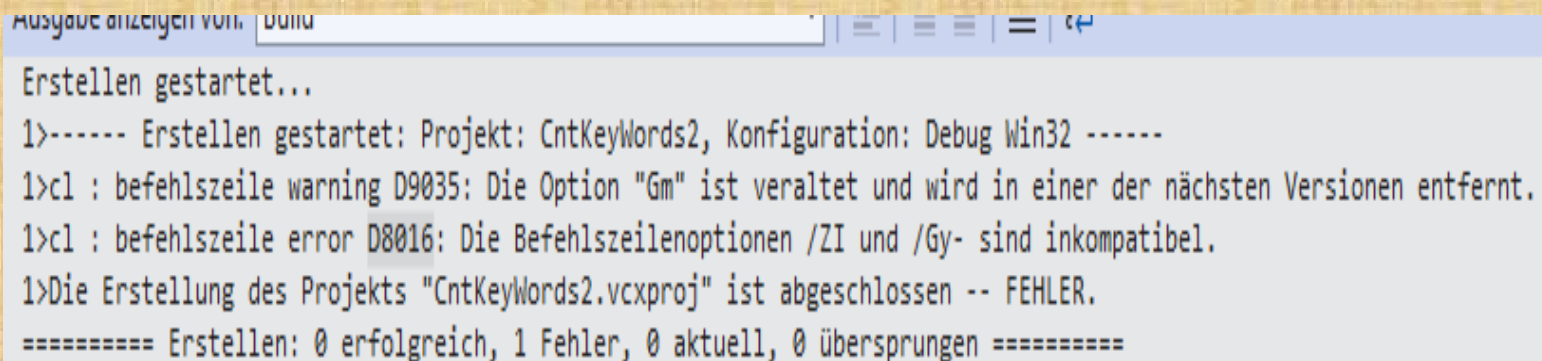
befehlszeile warning D9035: Die Option "Gm" ist veraltet und wird in einer der nächsten Versionen entfernt.



Stringpooling aktivieren	
Minimale Neuerstellung aktivieren	
C++-Ausnahmen aktivieren	Ja (/EHsc)
Überprüfen von kleineren Typen	Nein
Vollständige Laufzeitüberprüfungen	Beide (/RTC1 entspricht /RTCsu) (/RTC'
Laufzeitbibliothek	Multithreaded-Debug-DLL (/MDd)

Eintrag komplett entfernen

D8016: Die Optionen „/ZI und /Gy- sind inkompatibel“

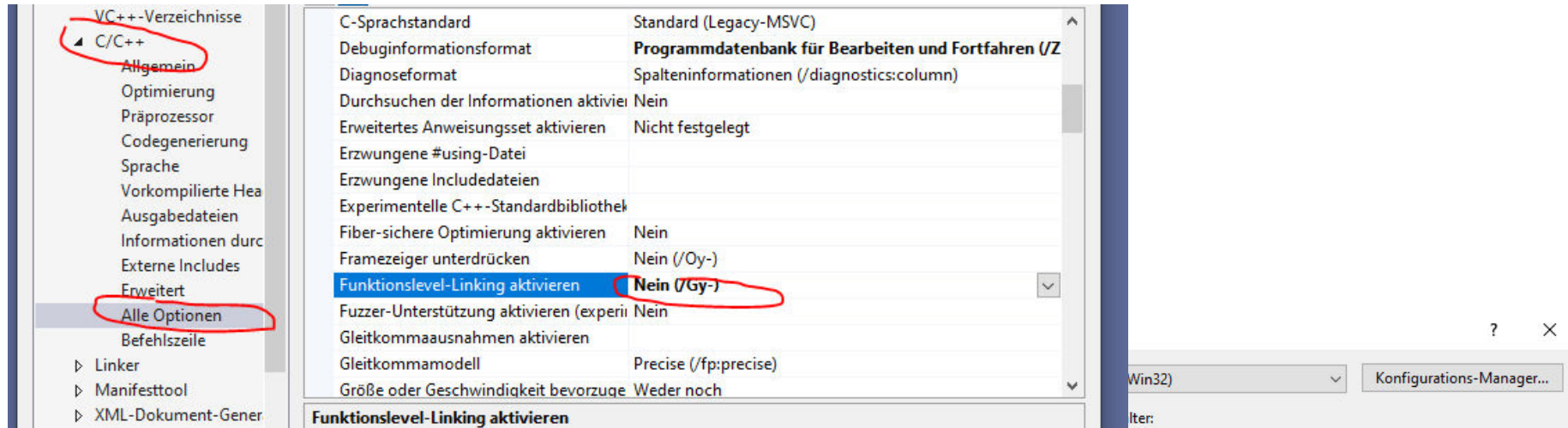


Ausgabe anzeigen von: Build |  |  |  |  | 

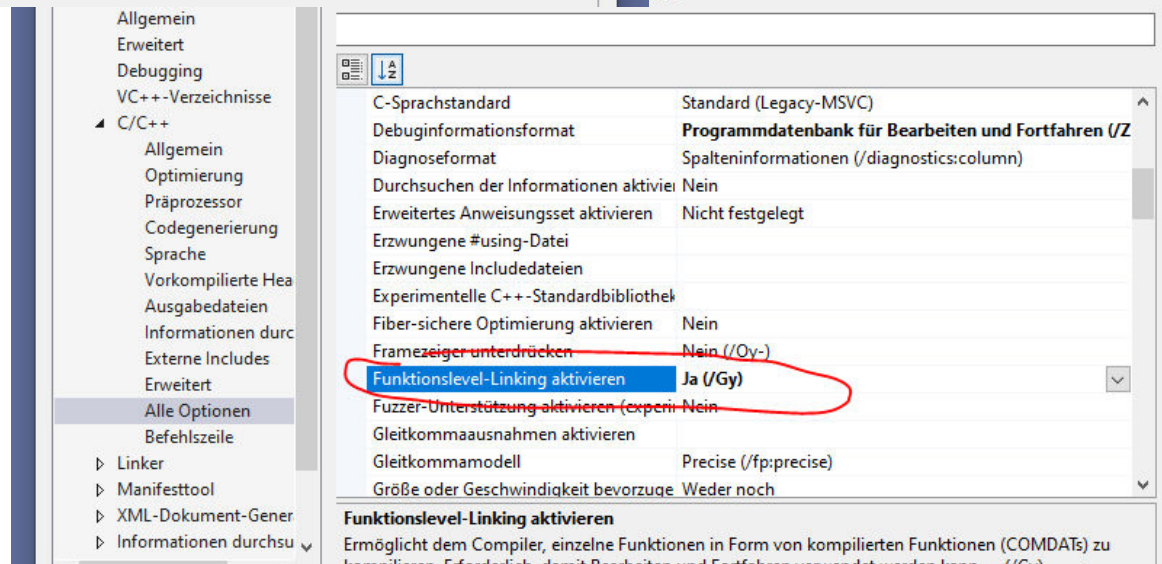
```
Erstellen gestartet...
1>----- Erstellen gestartet: Projekt: CntKeyWords2, Konfiguration: Debug Win32 -----
1>cl : befehlsszeile warning D9035: Die Option "Gm" ist veraltet und wird in einer der nächsten Versionen entfernt.
1>cl : befehlsszeile error D8016: Die Befehlszeilenoptionen /ZI und /Gy- sind inkompatibel.
1>Die Erstellung des Projekts "CntKeyWords2.vcxproj" ist abgeschlossen -- FEHLER.
===== Erstellen: 0 erfolgreich, 1 Fehler, 0 aktuell, 0 übersprungen =====
```

Fehler-Meldung der Gy- inkompatibel zu /ZI

befehlszeile warning D8016:



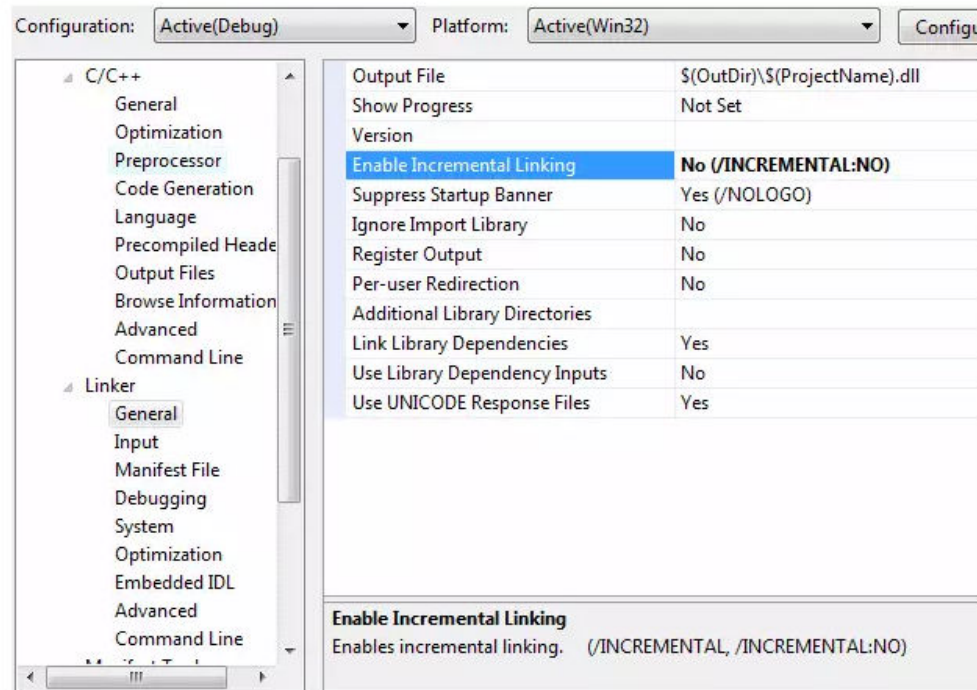
Eintrag ändern
nach /Gy sowie
nächste Folie



Fehler-Meldung der Gy- inkompatibel zu /ZI (2)

Siehe hierzu auch <https://gehirnwindung.de/categories/cpp/warning-Ink4075-editandcontinue-incrementalno>

Wenn der C++ Linker die Warnung **LNK4075** ausgibt, so kann man entweder die Einstellung **/INCREMENTAL** von **NO** auf **YES** stellen, oder - so kann man meinen - **/EDITANDCONTINUE** deaktivieren.

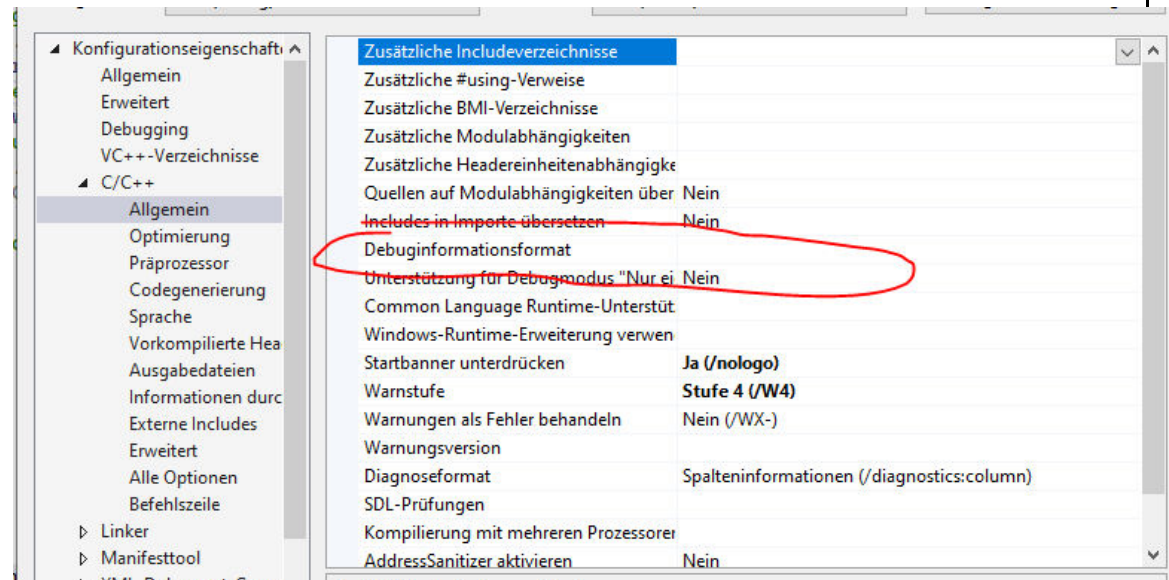
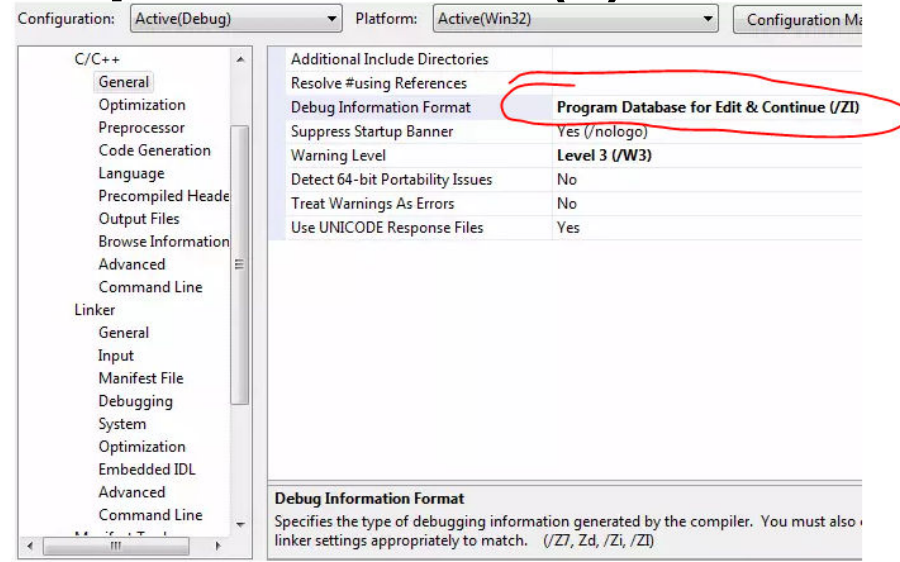


Fehler-Meldung der Gy- inkompatibel zu /ZI (3)

Während man die erstgenannte Einstellung in den Projekteigenschaften unter Linker ⇒ General ⇒ Enable Incremental Linking findet, ist die Einstellung für Edit And Continue allerdings nicht auffindbar.

Diese ist über den Visual Studio Dialog nicht direkt einstellbar. Stattdessen gibt es eine Einstellung unter C/C++ ⇒ General ⇒ Debug Information Format. Ist hier /ZI (also **Program Database for Edit & Continue**) eingestellt, so impliziert das die (interne) Linkeroption /EDITANDCONTINUE.

Eintrag entfernen



Compiler-Warnung C6262

Compiler-Warnung C6262 in Visual Studio 2019

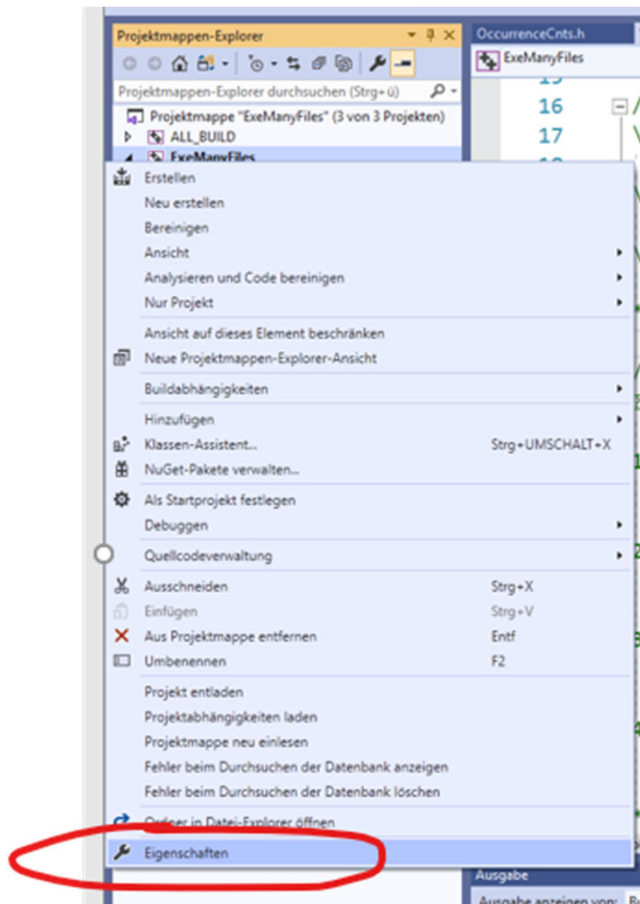
Siehe hierzu auch:

[https://docs.microsoft.com/de-de/cpp/code-quality/c6262?f1url=%3FappId%3DDev16IDEF1%26I%3DDE-DE%26k%3Dk\(C6262\)%26rd%3Dtrue&view=vs-2019](https://docs.microsoft.com/de-de/cpp/code-quality/c6262?f1url=%3FappId%3DDev16IDEF1%26I%3DDE-DE%26k%3Dk(C6262)%26rd%3Dtrue&view=vs-2019)

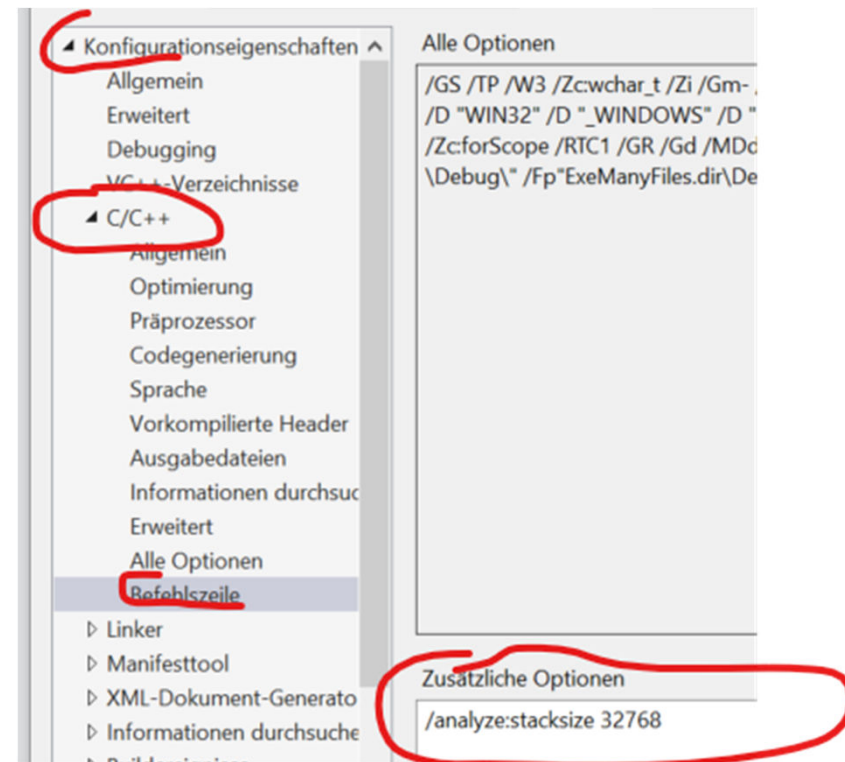
1. Wählen Sie in der Visual Studio-IDE das Projekt im Fenster Projektmappen-Explorer aus.
2. Klicken Sie in der Menüleiste auf Projekt > Eigenschaften.
3. Wählen Sie im Dialogfeld Eigenschaften Seiten die Eigenschaften Seite Konfigurations **Eigenschaften > C/C++- > Befehlszeile** aus.
4. Fügen Sie in zusätzliche Optionen hinzu `/analyze:stacksize <new-size>` , wobei `<new-size>` größer als `constant_1` ist. Wenn `constant_1` z. b. 27180 ist, können Sie eingeben `/analyze:stacksize 32768` . Klicken Sie auf OK, um die Änderungen zu speichern.

Compiler-Warnung C6262 in Visual Studio 2019 (2)

1. Wählen Sie in der Visual Studio-IDE das Projekt im Fenster Projektmappen-Explorer aus.

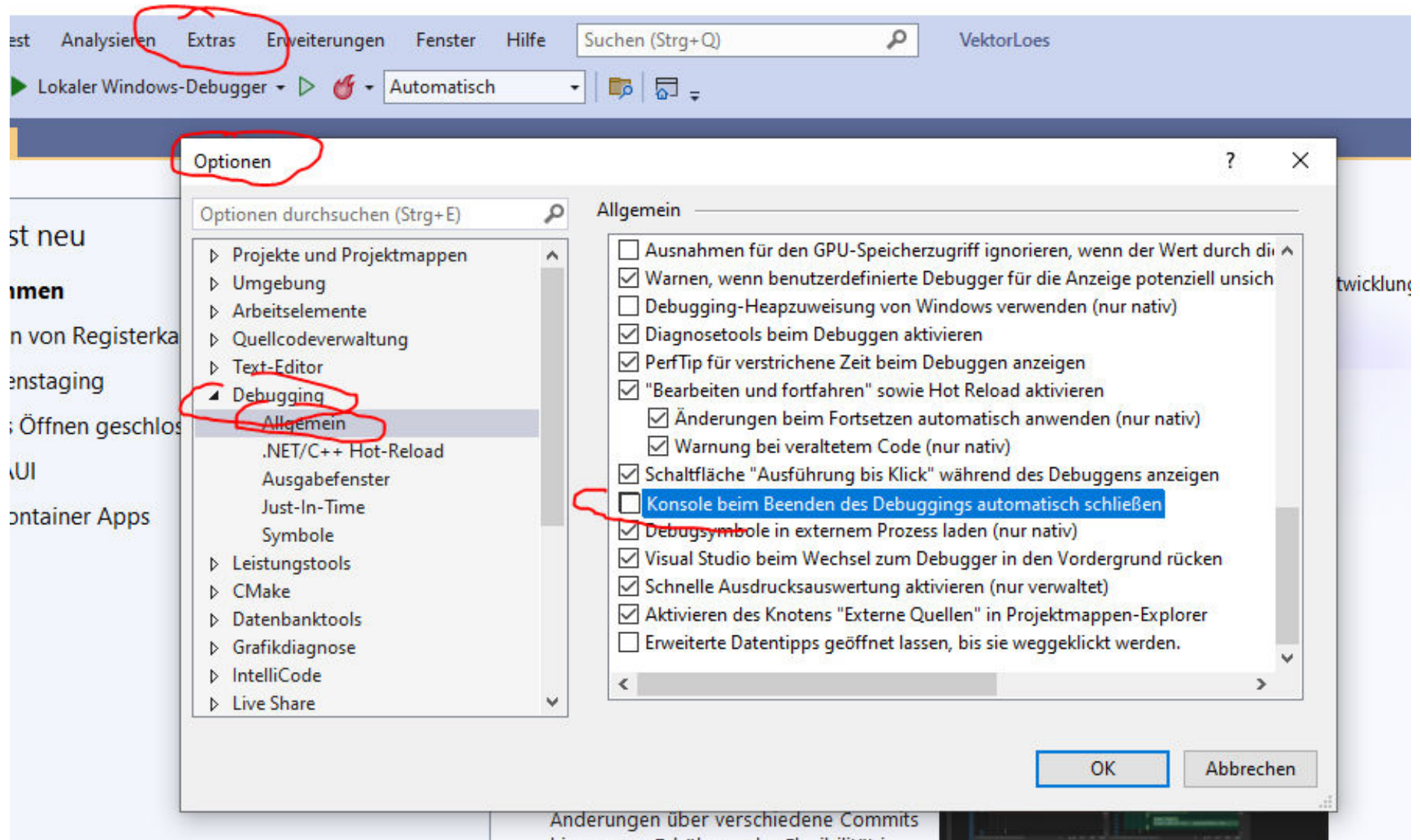


3. Wählen Sie im Dialogfeld Eigenschaften die Eigenschaften Seite Konfigurations Eigenschaften > C/C++- > Befehlszeile aus.



**Konsolen-Fenster am Programmende
NICHT schließen**

Extras→Optionen-→Debugging→Allgemein→



C++-Sprach-Standard einstellen

Eigenschaften (mit rechter Maus) → C/C++ → Sprache

