
Option Private Module

Autor & Copyright: Dipl.-Ing. Harald Nahrstedt

Version: 2016 / 2019 / 2021 / 365

Erstellungsdatum: 15.02.2012

Überarbeitung: 01.12.2023

Beschreibung:

Wenn wir Prozeduren als Makros aufrufen wollen, setzen wir ein *Public* für das *Sub*. Wenn wir sie verstecken wollen, dann setzen wir ein *Private* vor das *Sub*. Damit ist die Prozedur nicht mehr als Makro aufrufbar. Sie kann aber auch nicht von Prozeduren anderer Module genutzt werden. Eine Ausnahme ist der Aufruf aus einem Menü (CommandBar). Doch es gibt einen Ausweg aus dem Dilemma.

Anwendungs-Datei: AE-003_OptionPrivateModule.xlsm

Will man Prozeduren in einem Codemodul auch in Prozeduren anderer Codemodule nutzen, ohne dass sie als Makro aufrufbar sind, dann hilft die Anweisung *Option Private Module* weiter. Doch zunächst ein paar Vorüberlegungen.

Eine Prozedur in einem Codemodul hat die allgemeine Form

```
Sub Prozedurname
    Anweisungen
End Sub
```

Wie schon im Buch erläutert, ist sie damit als Makro aufrufbar. Dies kann man noch besser verdeutlichen, wenn man vor das *Sub* ein *Public* setzt, so wie in dem nachfolgenden Beispiel.

Codeliste 1. Prozedur Test1 in modModule1

```
Public Sub Test1()
    MsgBox "Prozedur Test1"
End Sub
```

Soll eine Prozedur nicht mehr aufrufbar sein, wird *Public* gegen *Private* ausgetauscht. Jetzt zeigt das Makro-Anwahlfenster die Prozedur *Test1* nicht mehr an (Bild 1).

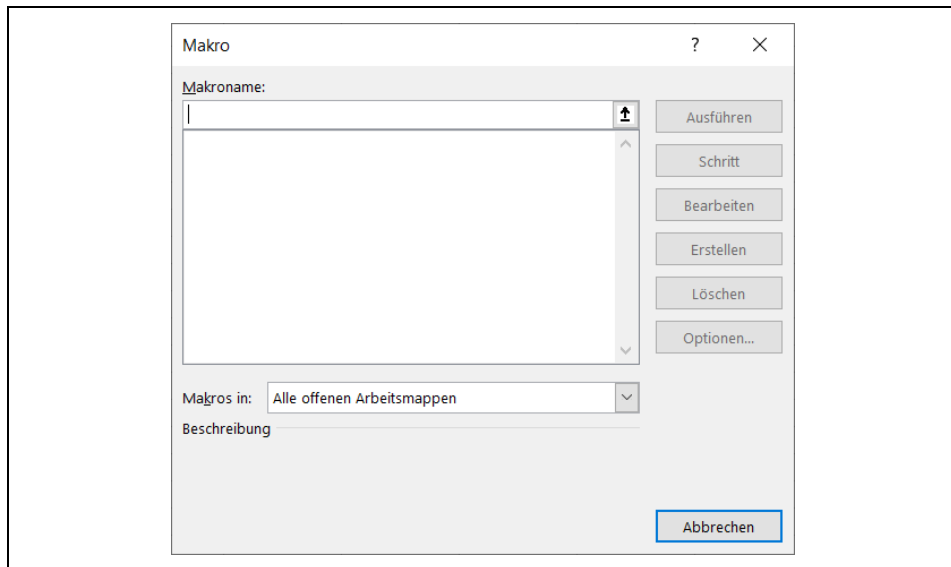


Bild 1. Keine Anzeige in der Makrosauswahl

Damit ist die Prozedur aber aus keinem anderen Codemodul aufrufbar. In einem zweiten Modul befindet sich die nachfolgende Prozedur.

Codeliste 2. Prozedur Aufruf_Test1 in modModule2

```
Public Sub Aufruf_Test1()
    Call Test1
End Sub
```

Diese Prozedur ist zwar als Makro aufrufbar, liefert aber folgendes Ergebnis (Bild 2).

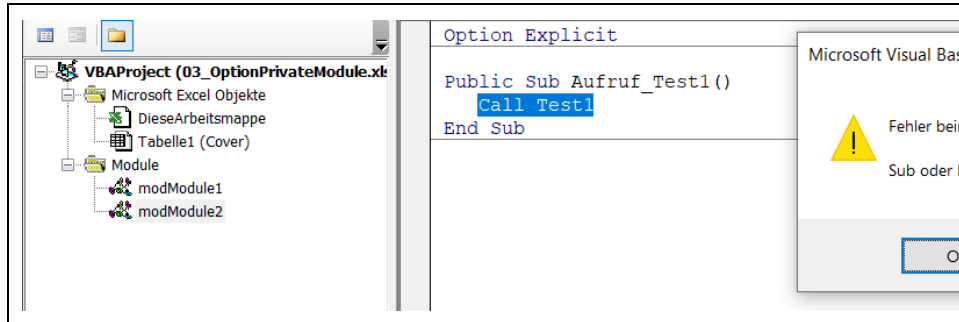


Bild 2. Fehlermeldung beim Kompilieren

Eine Möglichkeit ist, der Prozedur einen Parameter mitzugeben, denn Prozeduren mit Parametern lassen sich nicht als Makros aufrufen. Dazu wandeln wir beide Prozeduren in den unterschiedlichen Modulen wie folgt ab.

Codeliste 3. Prozedur Test2 in modModule1

```
Public Sub Test2(Temp As Boolean)
    MsgBox "Prozedur Test2"
End Sub
```

Codeliste 4. Prozedur Aufruf_Test2 in modModule2

```
Public Sub Aufruf_Test2()
    Call Test2(True)
End Sub
```

Oder noch raffinierter, wenn wir einen Parameter nur optional benutzen.

Codeliste 5. Prozedur Test3 in modModule1

```
Public Sub Test3(Optional Temp As Boolean)
    MsgBox "Prozedur Test3"
End Sub
```

Codeliste 6. Prozedur Aufruf_Test3 in modModule2

```
Public Sub Aufruf_Test3()
    Call Test3
End Sub
```

Doch es geht noch einfacher. In dem Codemodul, indem sich Public-Prozeduren befinden, die nicht als Makros aufgerufen werden dürfen, wird am Anfang die Anweisung *Option Private Module* gesetzt. Am besten direkt unter die Anweisung *Option Explicit*.

Codeliste 7. Prozedur Test4 in modModule3

```
Option Explicit
Option Private Module

Public Sub Test4()
    MsgBox "Prozedur Test4"
End Sub
```

Codeliste 8. Prozedur Aufruf_Test4 in modModule4

```
Option Explicit
```

```
Sub Aufruf_Test4()  
    Call Test4  
End Sub
```