Bauhaus-Universität Weimar

Parallele und Verteilte Systeme

Dritte Übung

Wintersemester 2020/2021

Computer Vision in Engineering – Prof. Dr. Rodehorst
M.Sc. Mariya Kaisheva
mariya.kaisheva@uni-weimar.de

Bauhaus-
Universität
Weimar

Einrichtung von OpenMPI & MS-MPI

Bauhaus-
Universität
Weimar

Einrichtung unter **LINUX**

- OpenMPI
  - **mpic++** *<Quellcode Datei>* **-o** *<Name des Programms>*
  - **mpirun -np** *<Anzahl der Prozessen>* *<Name des Programms>*

Bauhaus-Universität Weimar

Einrichtung unter **LINUX**

Schritt **1:**

Quellcode schreiben

Open | helloMPI.c ~/PVS/MPI | Save

```c
#include <mpi.h>
#include <stdio.h>

int main(int argc, char** argv) {

    MPI_Init(NULL, NULL);


    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);


    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);


    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);


    printf("Hello world from processor %s, rank %d out of
%d processors\n",
           processor_name, world_rank, world_size);


    MPI_Finalize();
}
```

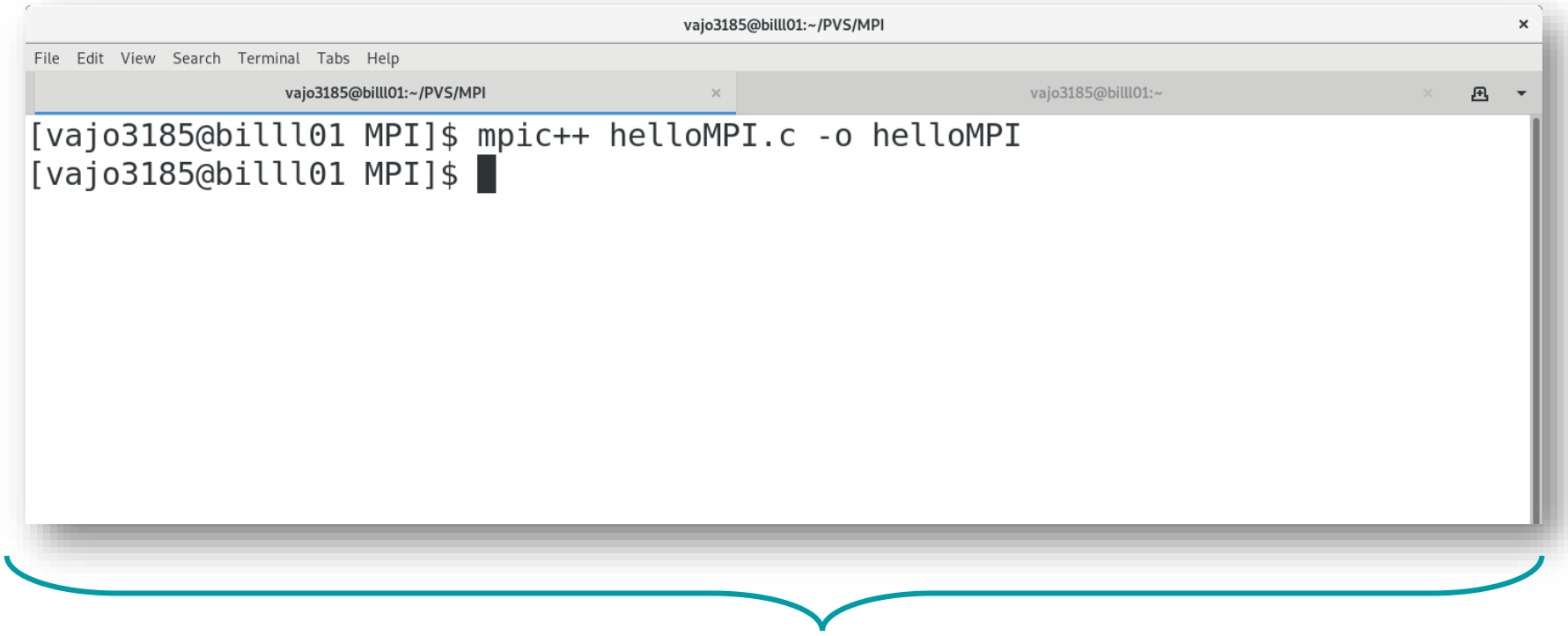
C | Tab Width: 8 | Ln 27, Col 2 | INS

Bauhaus-Universität Weimar

Einrichtung unter **LINUX**

Schritt **2:**

Quellcode kompilieren



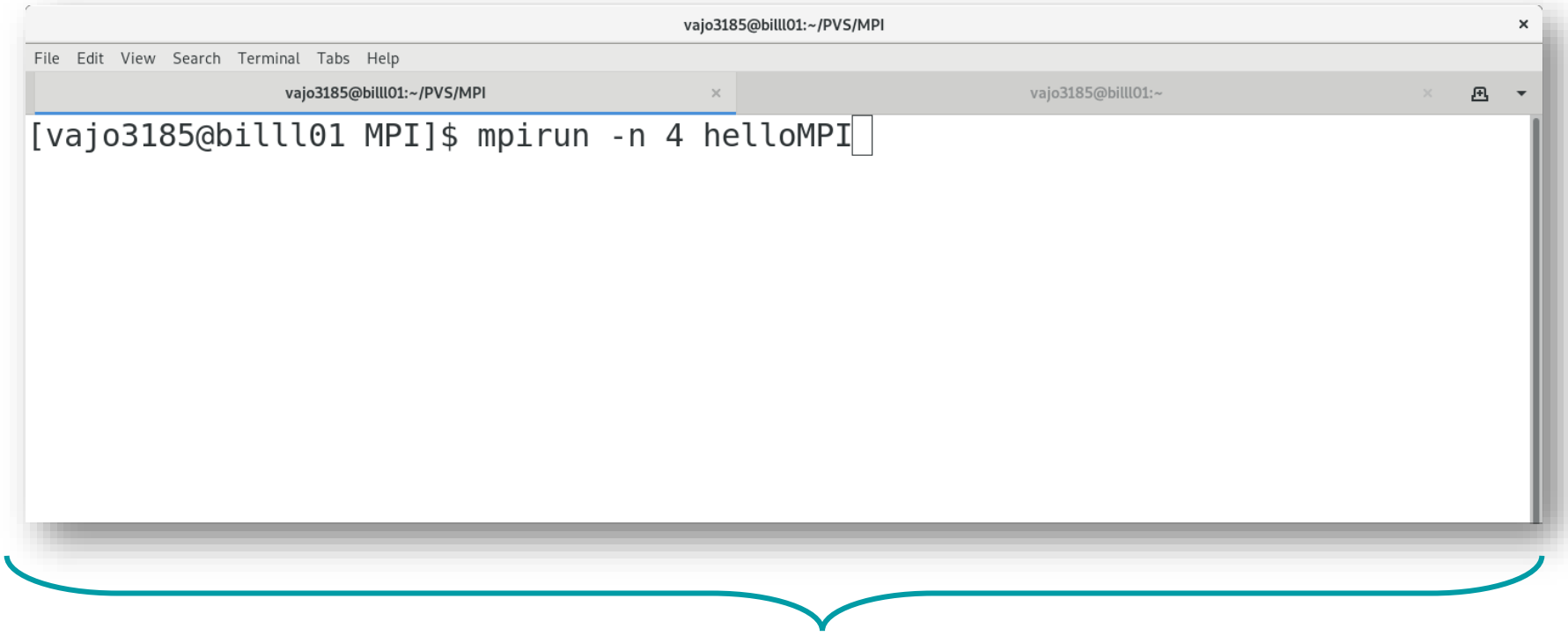
Konsole (Terminal)

Bauhaus-Universität Weimar

Einrichtung unter **LINUX**

Schritt **3:**

Programm ausführen



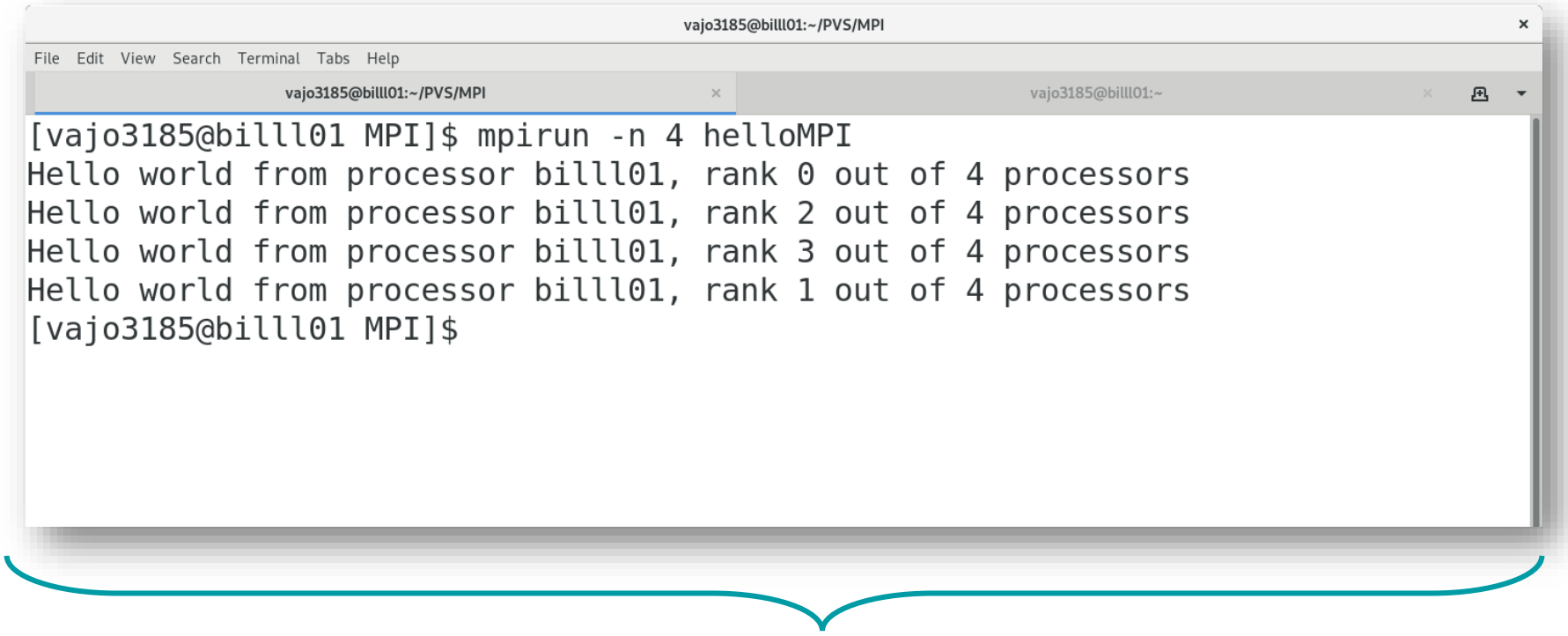
Konsole (Terminal)

Bauhaus-Universität Weimar

Einrichtung unter **LINUX**

Schritt **3:**

Programm ausführen

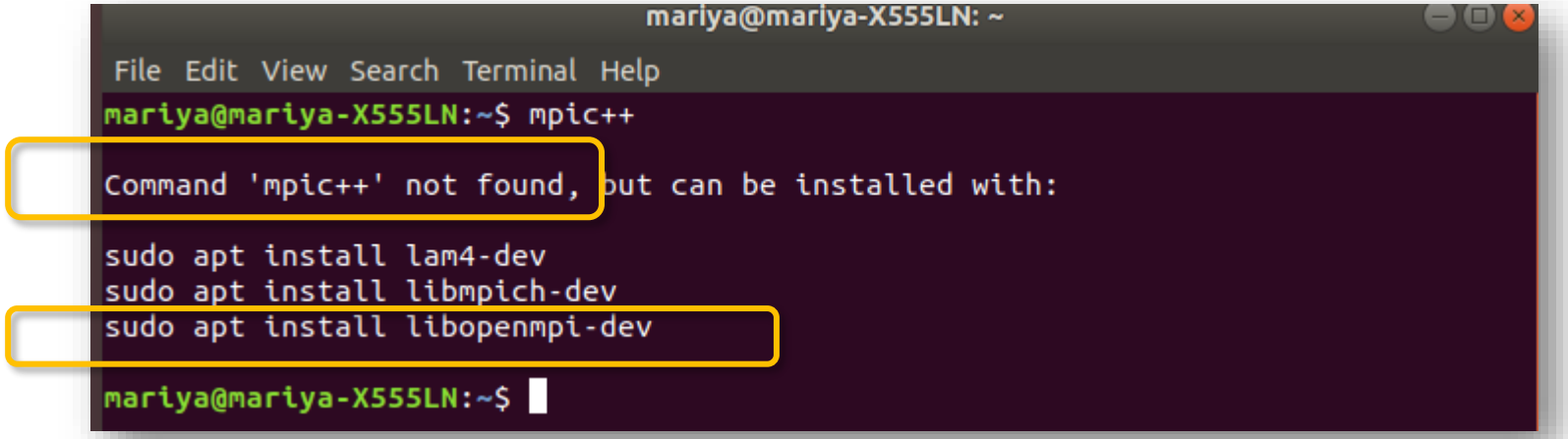


Konsole (Terminal)

Bauhaus-
Universität
Weimar

Einrichtung
unter **LINUX**

Falls der mpic++ Compiler fehlt



sudo apt install libopenmpi-dev

Bauhaus-
Universität
Weimar

Einrichtung
unter **LINUX**

Falls mpic++ Compiler fehlt

```
mariya@mariya-X555LN:~/work/PVS$ sudo apt install libopenmpi-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  autotools-dev ibverbs-providers libfabric1 libhwloc-dev libhwloc-plugins libhwloc5 libibverbs-dev libibverbs1 libltdl-dev
  libnl-route-3-200 libnuma-dev libopenmpi2 libpsm-infinipath1 librdmacm1 libtool ocl-icd-libopencl1 openmpi-bin openmpi-common
Suggested packages:
  libhwloc-contrib-plugins libtool-doc openmpi-doc autoconf automaken gfortran | fortran95-compiler gcj-jdk opencl-icd gfortran
The following NEW packages will be installed:
  autotools-dev ibverbs-providers libfabric1 libhwloc-dev libhwloc-plugins libhwloc5 libibverbs-dev libibverbs1 libltdl-dev
  libnl-route-3-200 libnuma-dev libopenmpi-dev libopenmpi2 libpsm-infinipath1 librdmacm1 libtool ocl-icd-libopencl1 openmpi-bin
  openmpi-common
0 upgraded, 19 newly installed, 0 to remove and 227 not upgraded.
Need to get 4.938 kB of archives.
After this operation, 20,5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
mariya@mariya-X555LN:~/work/PVS$ mpic++ helloMPI.cpp -o helloMPI
mariya@mariya-X555LN:~/work/PVS$ mpirun -n 4 helloMPI
```

Bauhaus-
Universität
Weimar

Einrichtung unter **Windows**

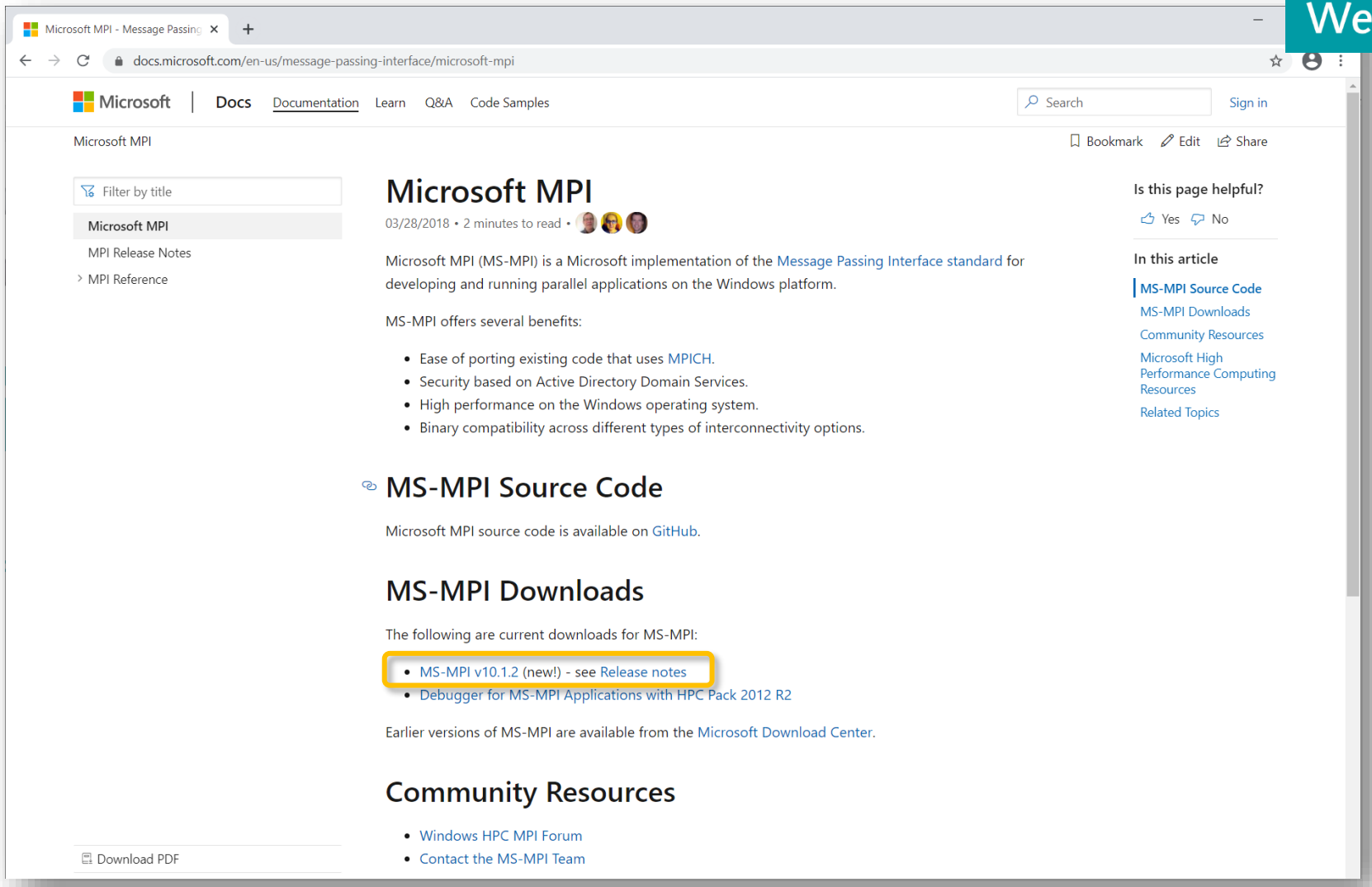
- OpenMPI ist zur Zeit unter Windows nur bis Version 1.6 unterstützt
- Windows bietet eigene Implementierung von MPI ⇨ **MS-MPI**

Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **1:**

MS-MPI herunterladen



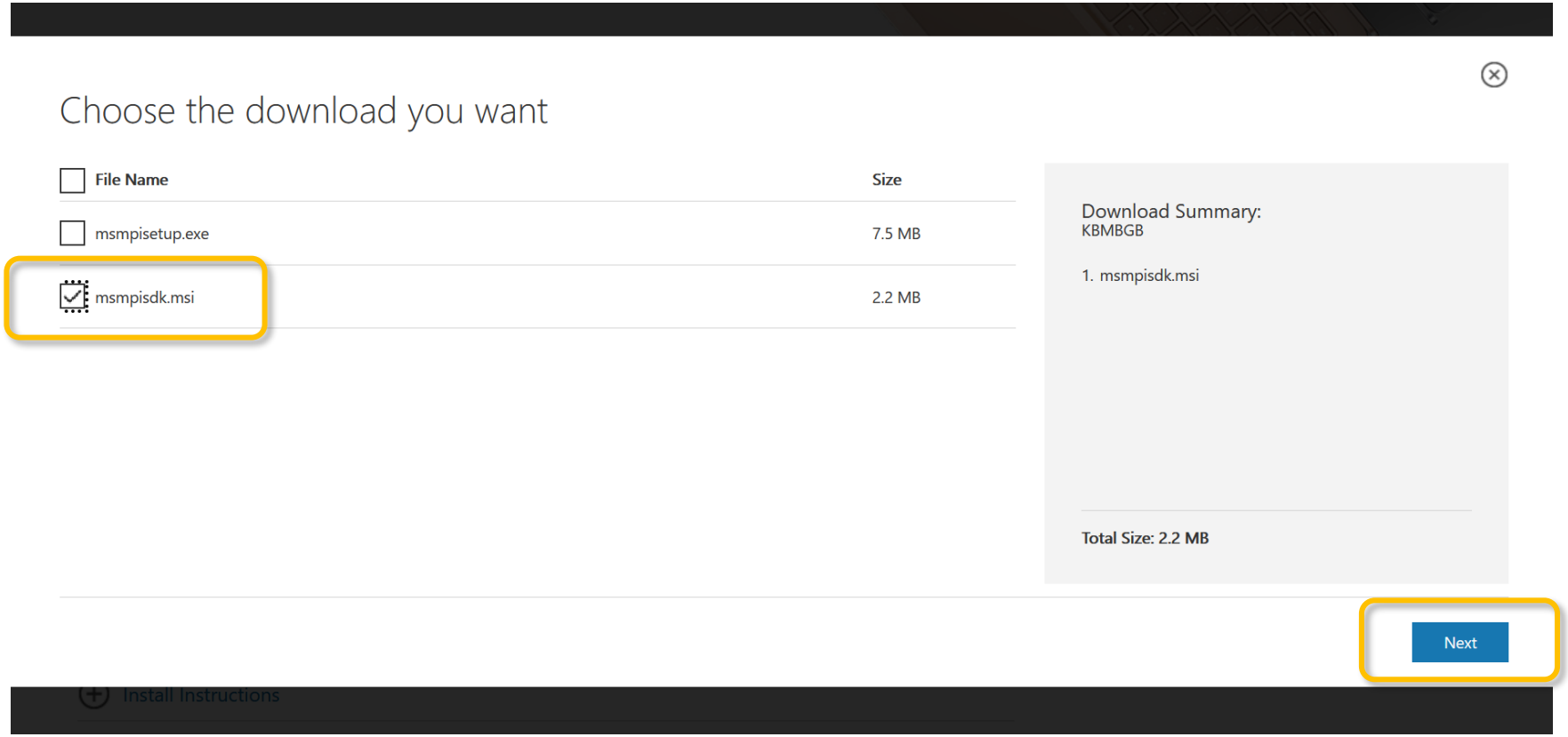
https://docs.microsoft.com/en-us/message-passing-interface/microsoft-mpi

Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **1:**

MS-MPI herunterladen

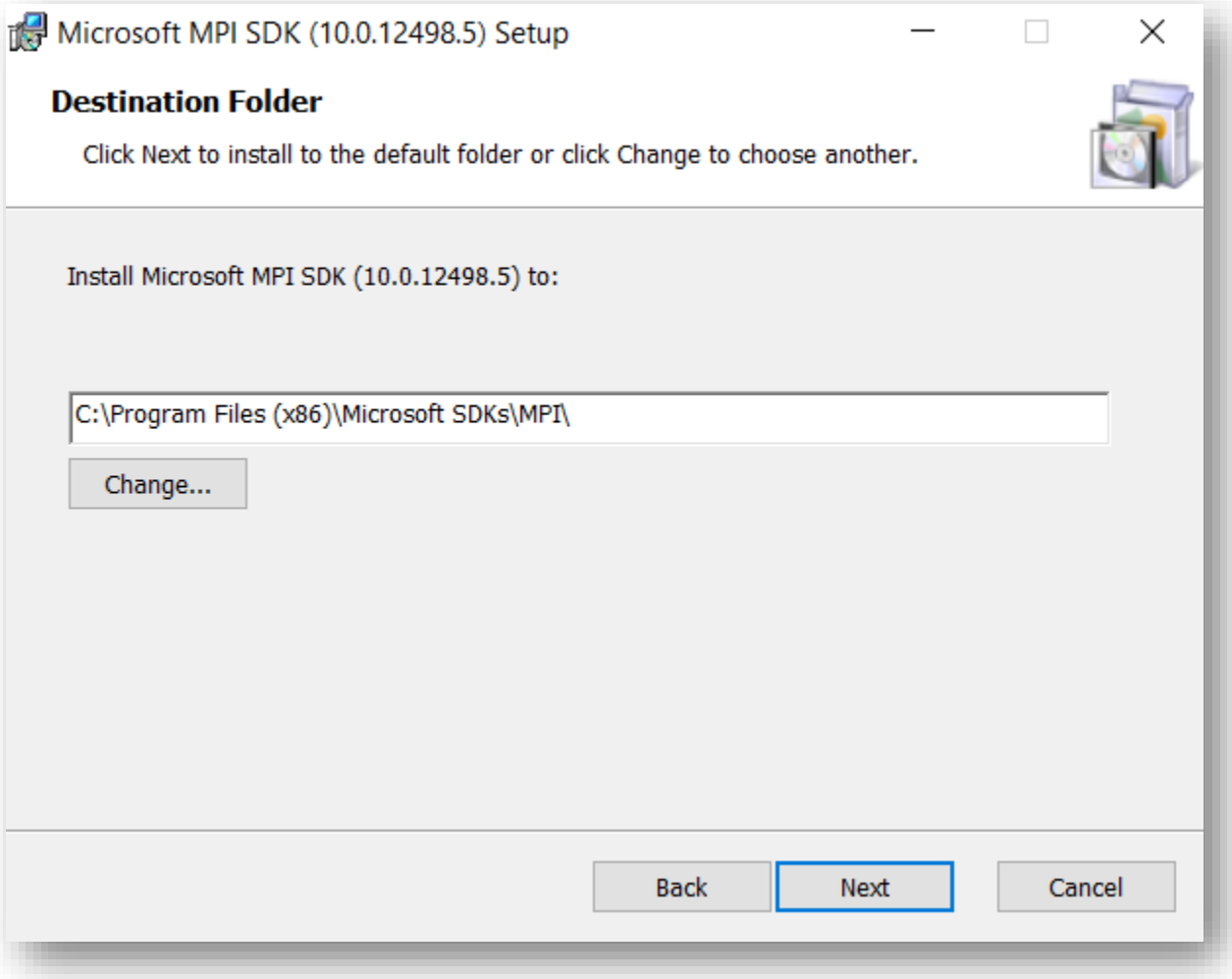


Bauhaus-
Universität
Weimar

Einrichtung unter **Windows**

Schritt **2:**

MS-MPI installieren



Bauhaus-
Universität
Weimar

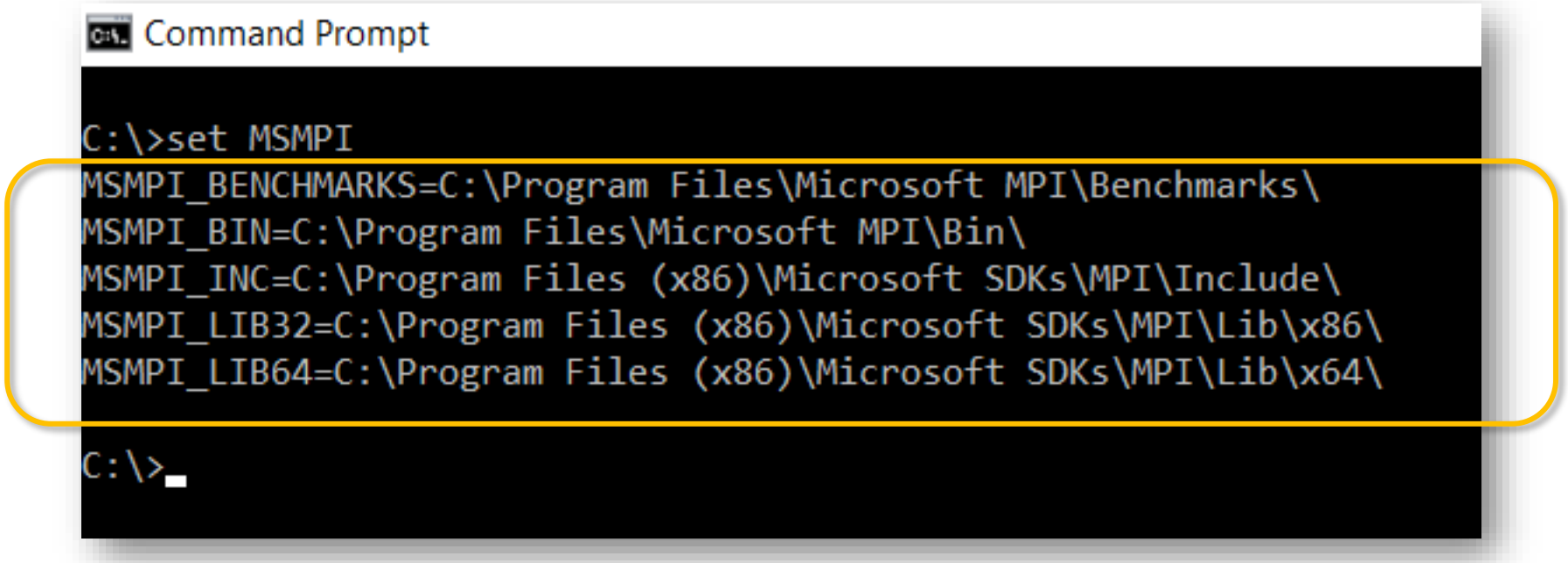
Einrichtung unter **Windows**

Test nach Schritt **2:**

Bauhaus-
Universität
Weimar

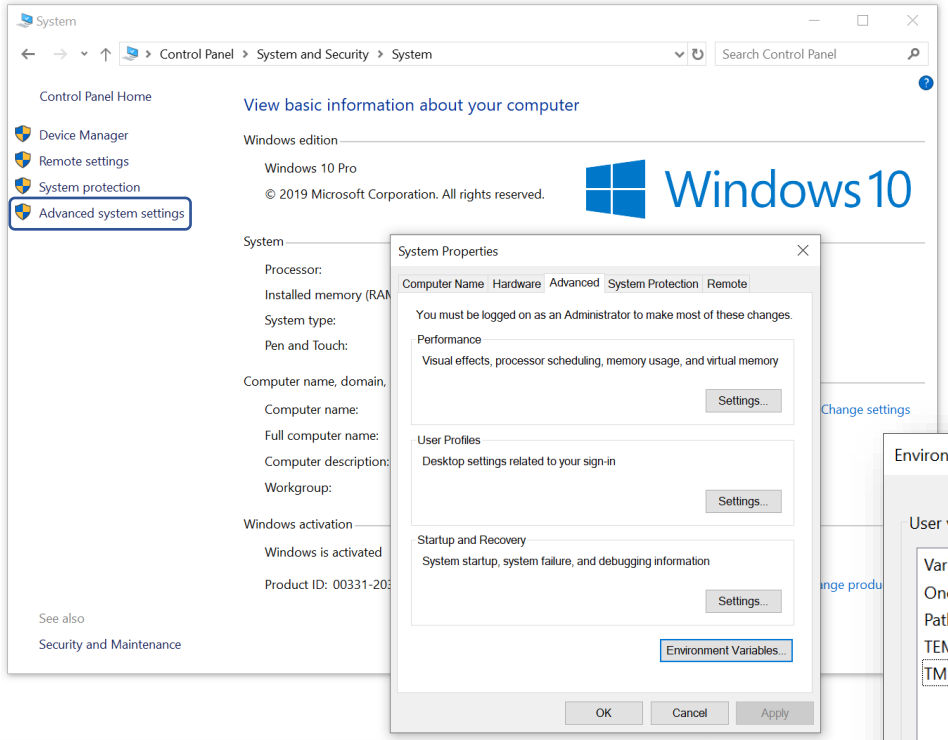
Einrichtung unter **Windows**

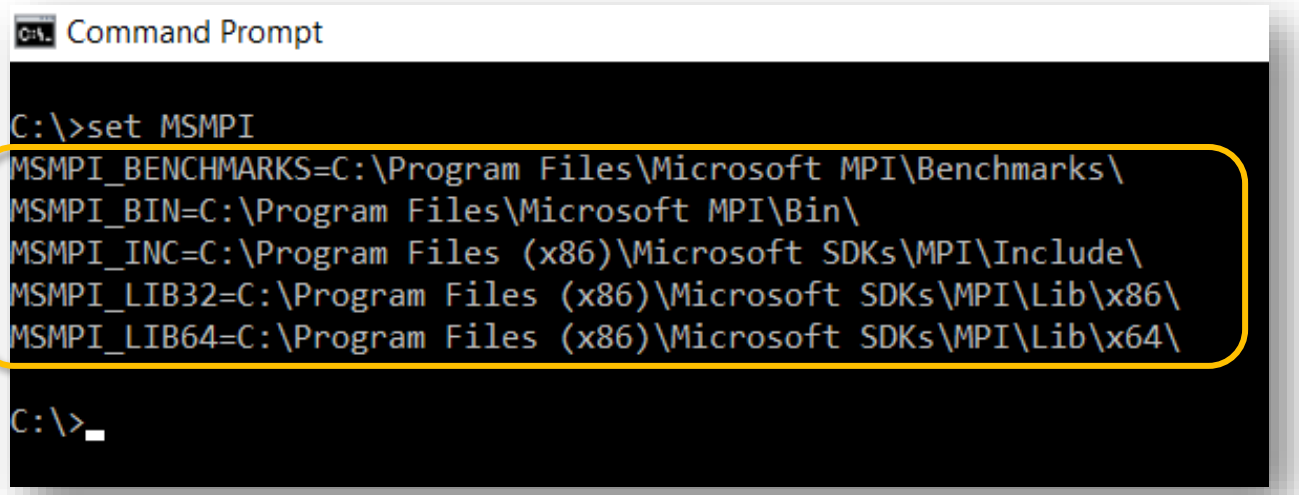
Test nach Schritt **2:**

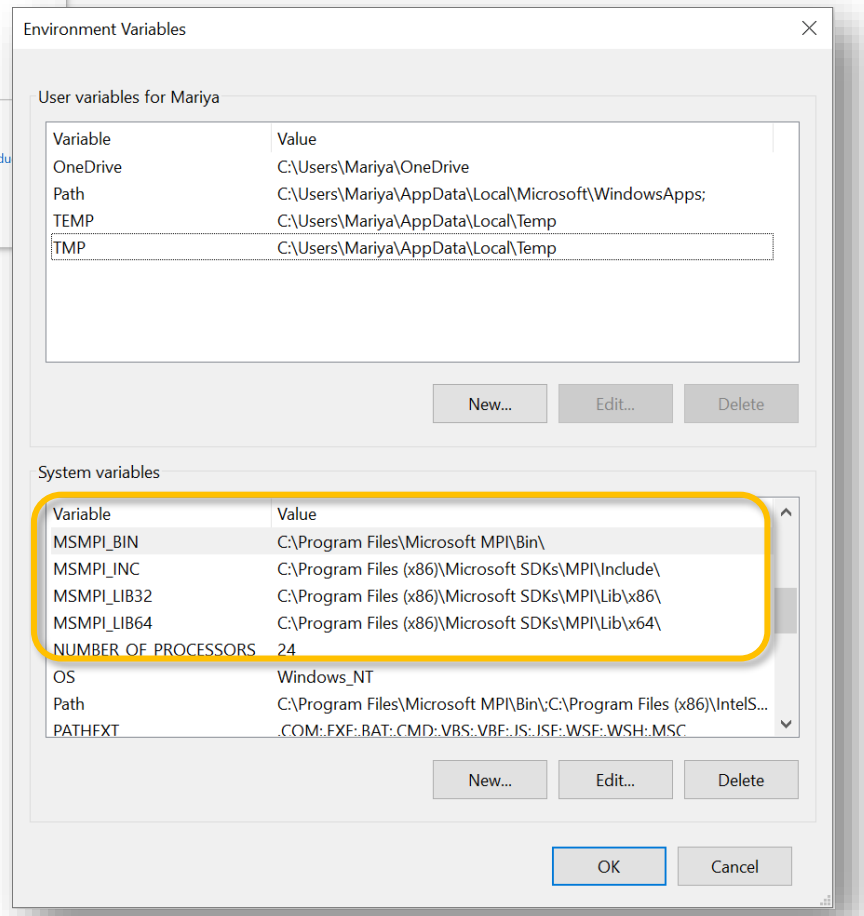


Einrichtung unter **Windows**

Test nach Schritt **2:**







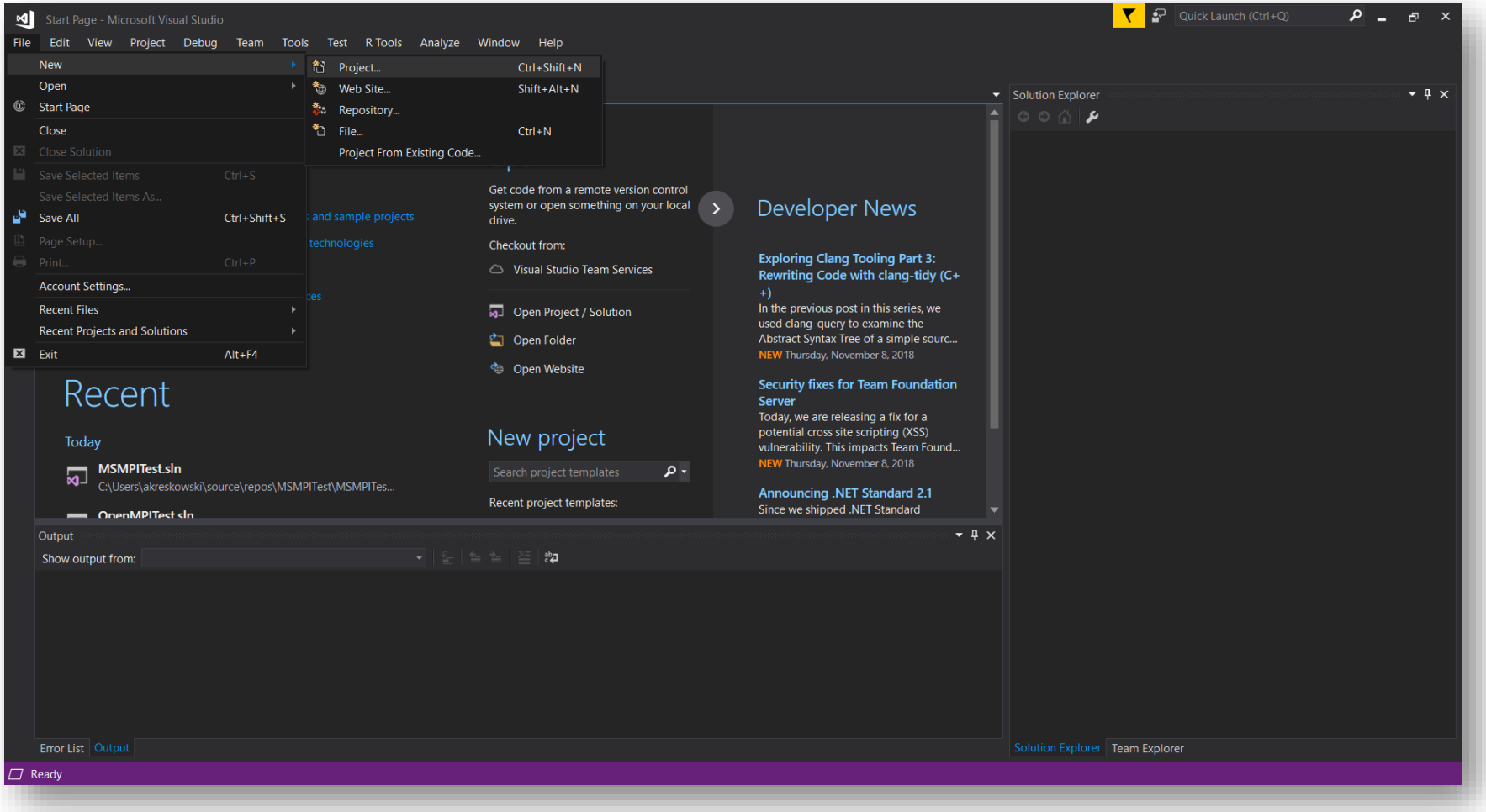
Bauhaus-
Universität
Weimar

Einrichtung unter **Windows**

Schritt **3:**

Projekt erstellen in **Visual Studio***

*Bitte nicht mit Visual Studio Code verwechseln



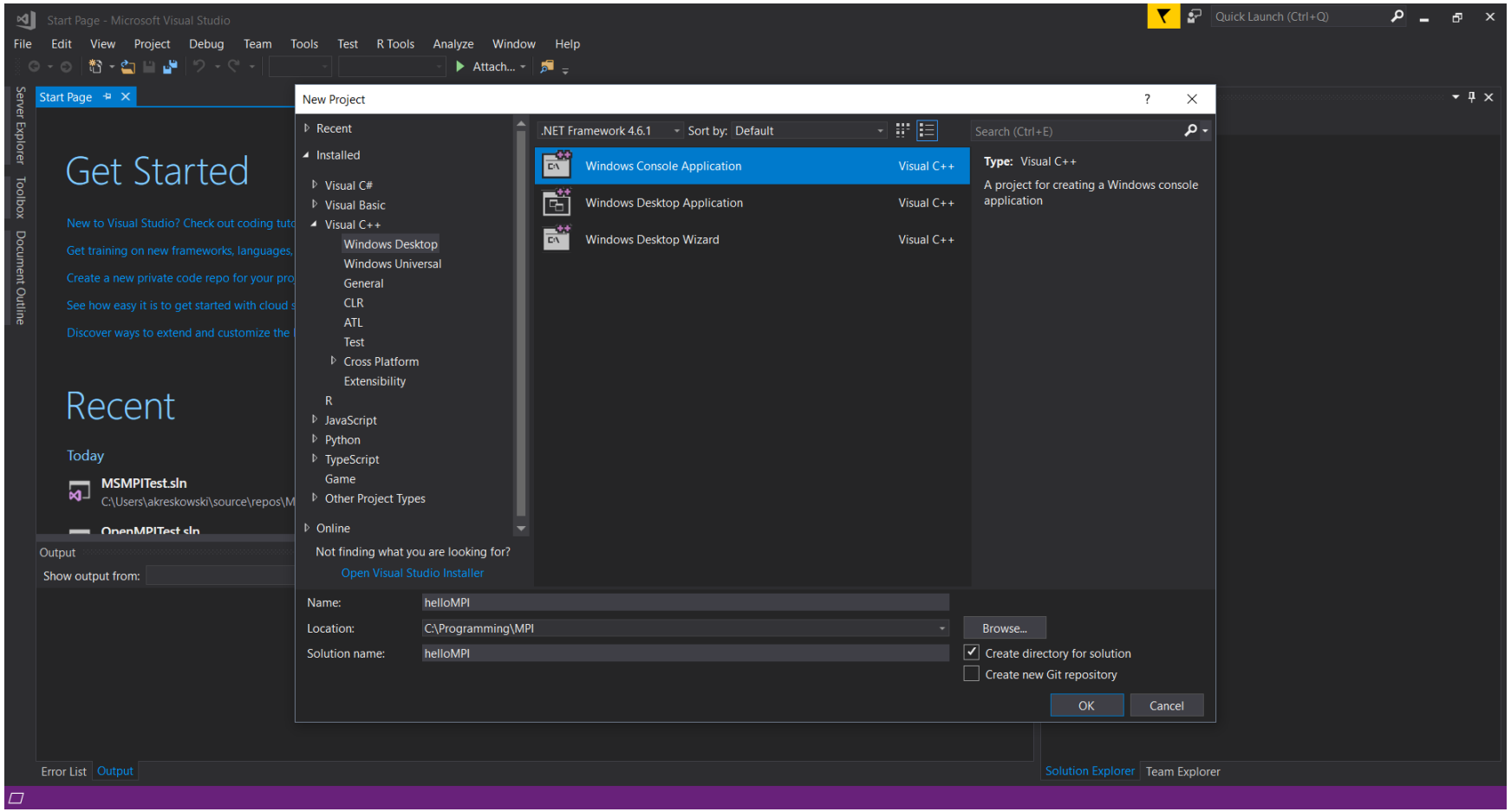
Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **3:**

Projekt erstellen in **Visual Studio***

*Bitte nicht mit Visual Studio Code verwechseln



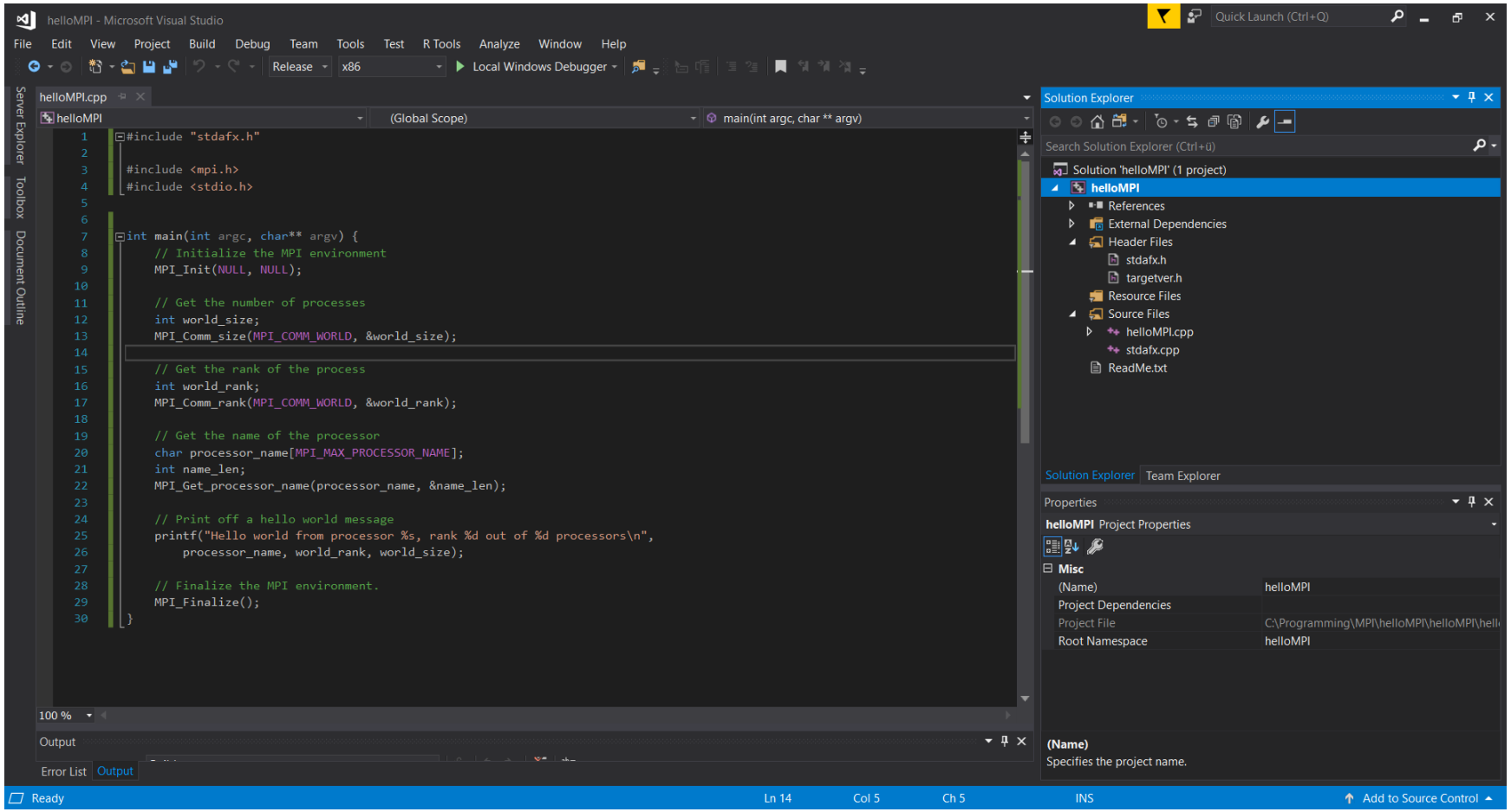
Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **3:**

Projekt erstellen in **Visual Studio***

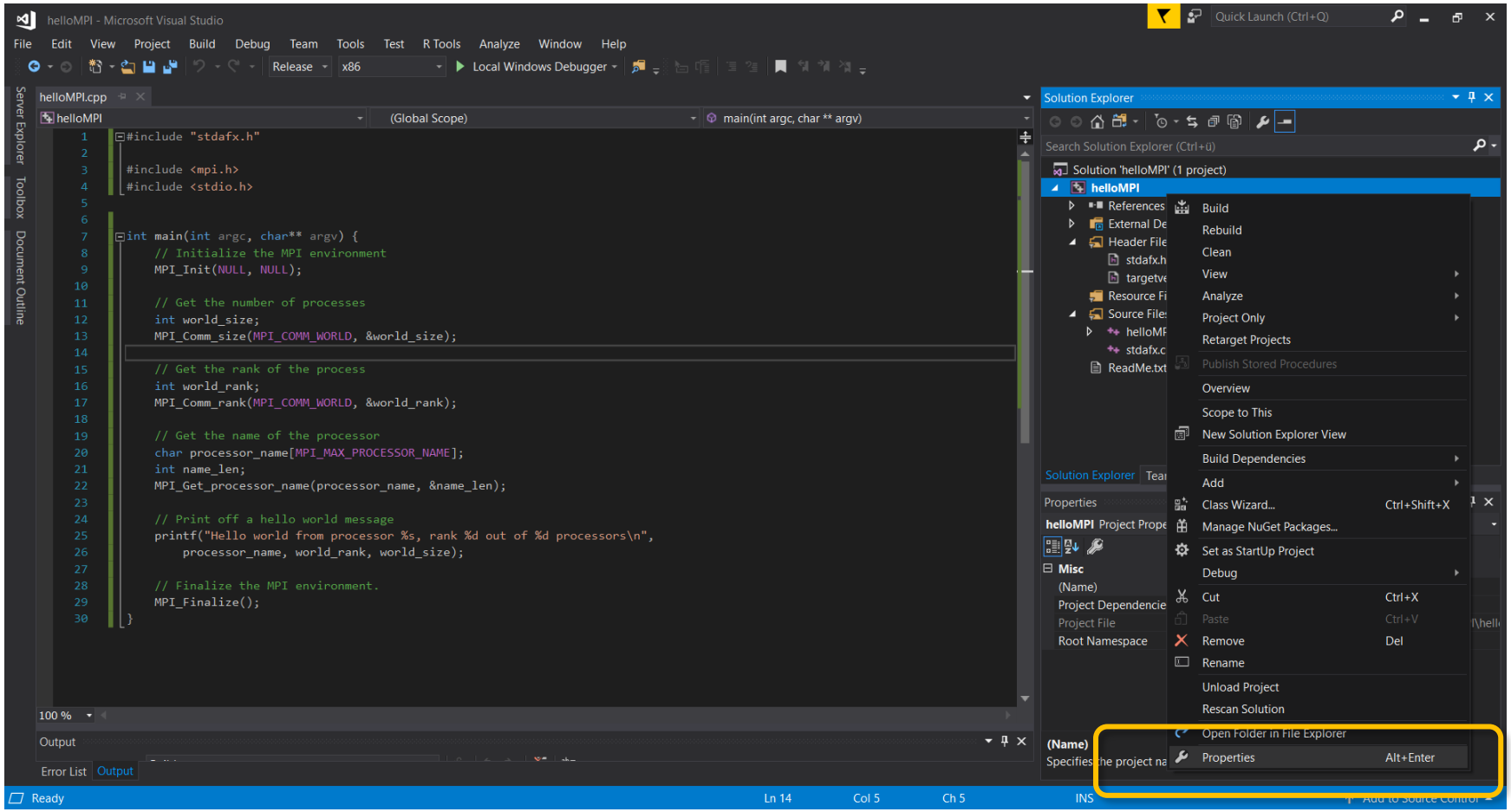
*Bitte nicht mit Visual Studio Code verwechseln



Einrichtung unter **Windows**

Schritt **4:**

MS-MPI für das erstellte **Visual Studio** Projekt einrichten



Project → Properties…

Project → Properties… → …

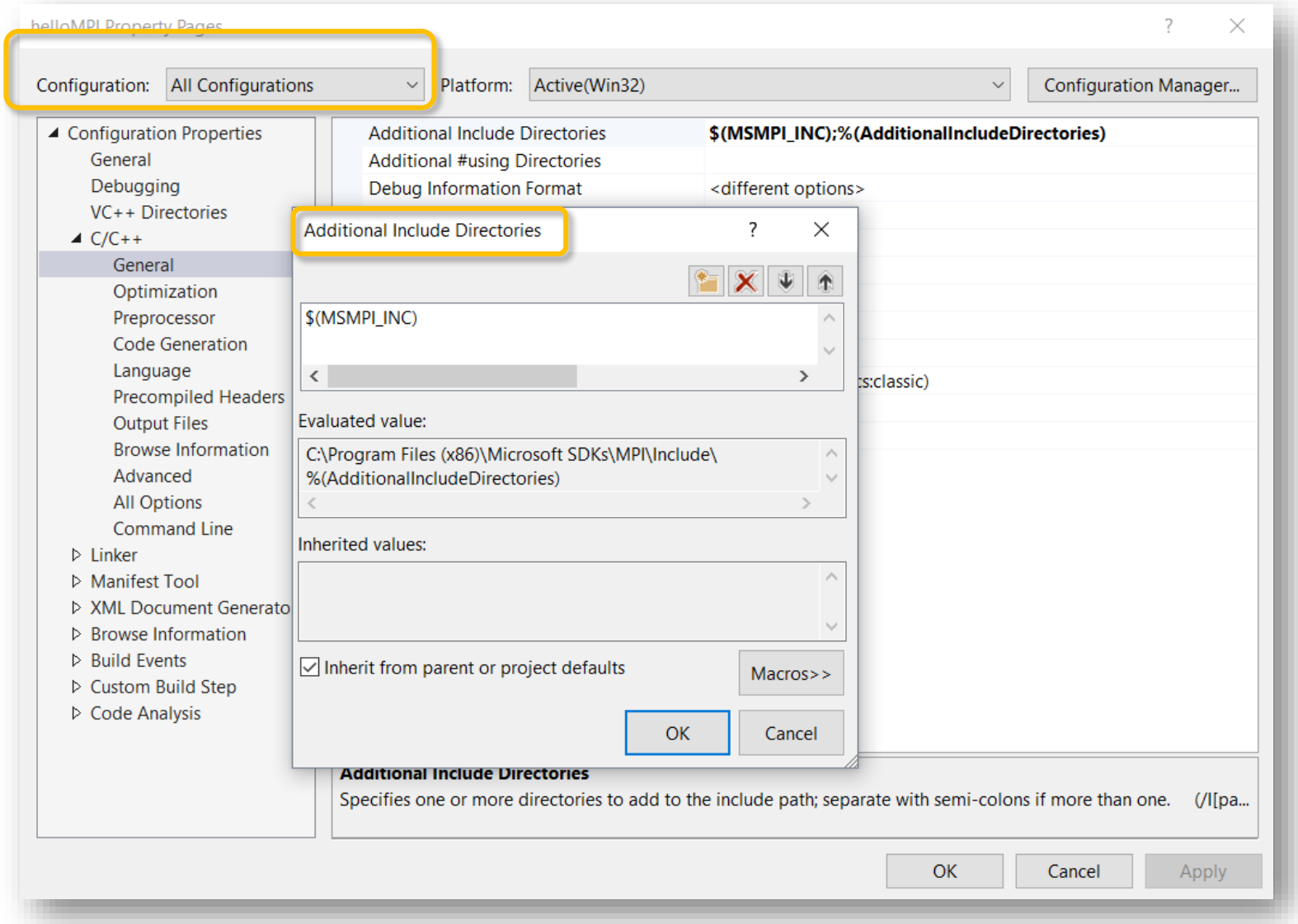
Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **4.1.:**

MS-MPI für das erstellte **Visual Studio** Projekt einrichten:

$(MSMPI_INC) unter *Additional **Include** Directories*



Project → Properties… → …

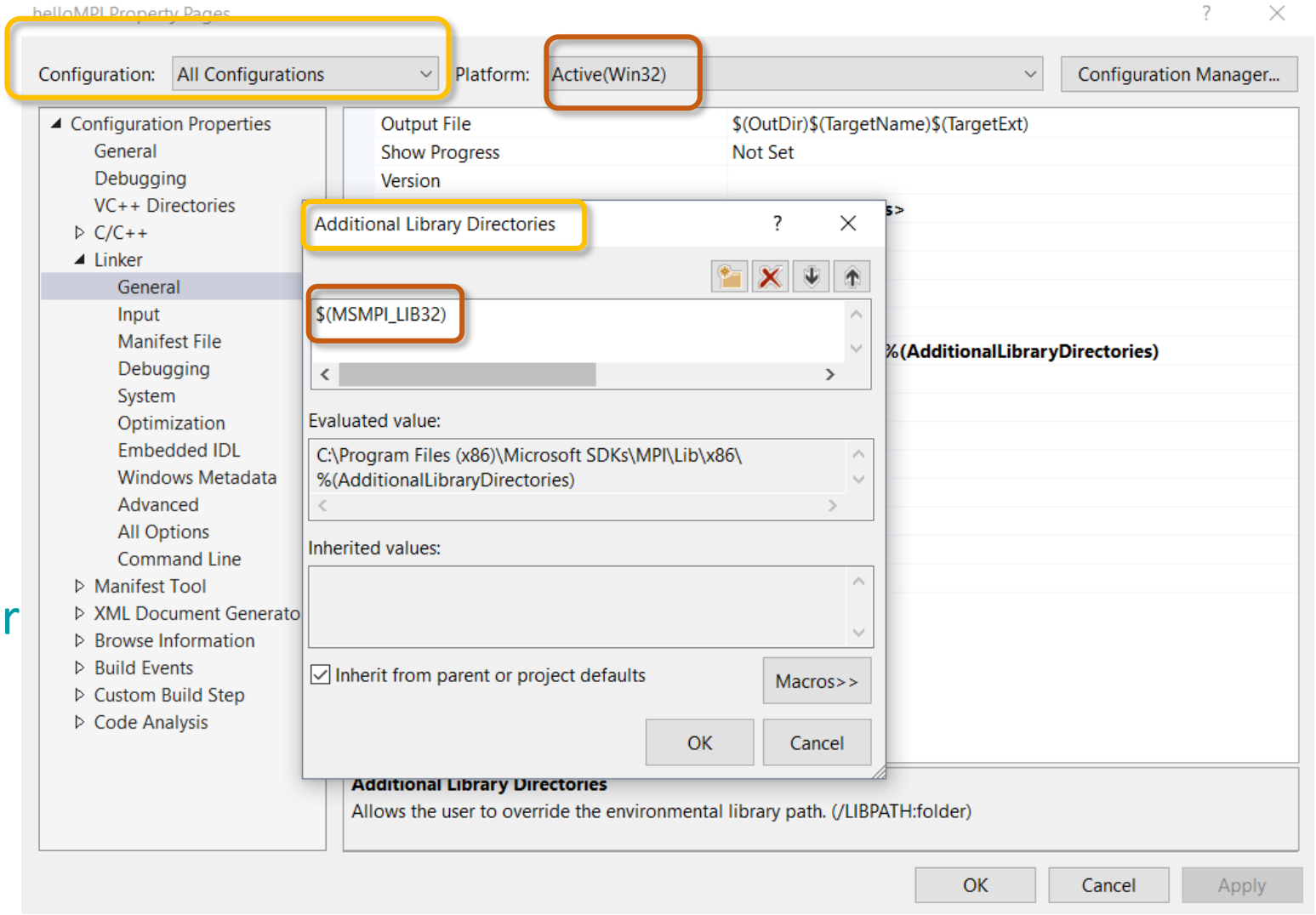
Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **4.2.:**

MS-MPI für das erstellte **Visual Studio** Projekt einrichten:

$(MSMPI_LIB32) unter *Additional **Library** Directories*



Project → Properties… → …

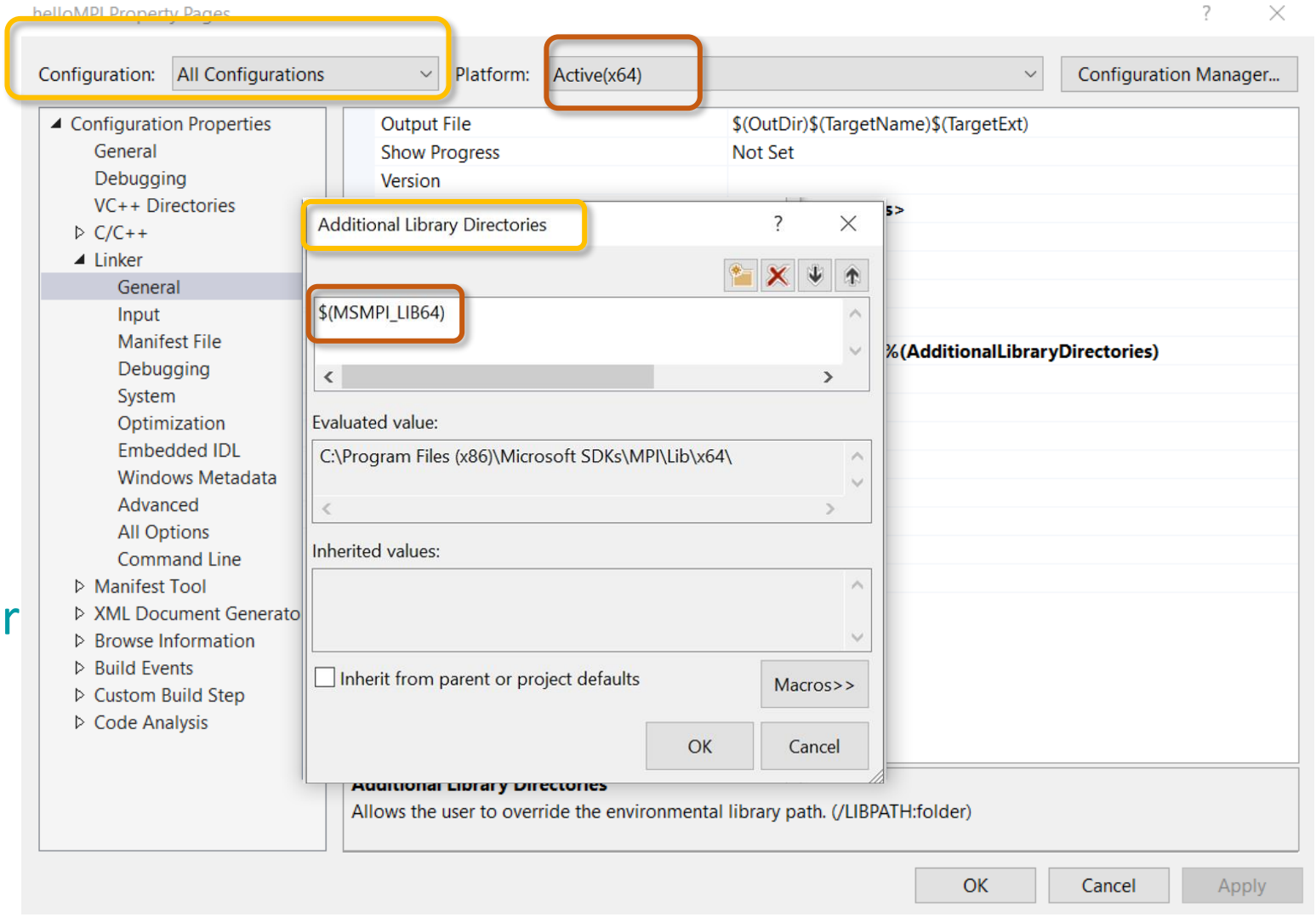
Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **4.2.:**

MS-MPI für das erstellte **Visual Studio** Projekt einrichten:

$(MSMPI_LIB64) unter *Additional **Library** Directories*



Project → Properties… → …

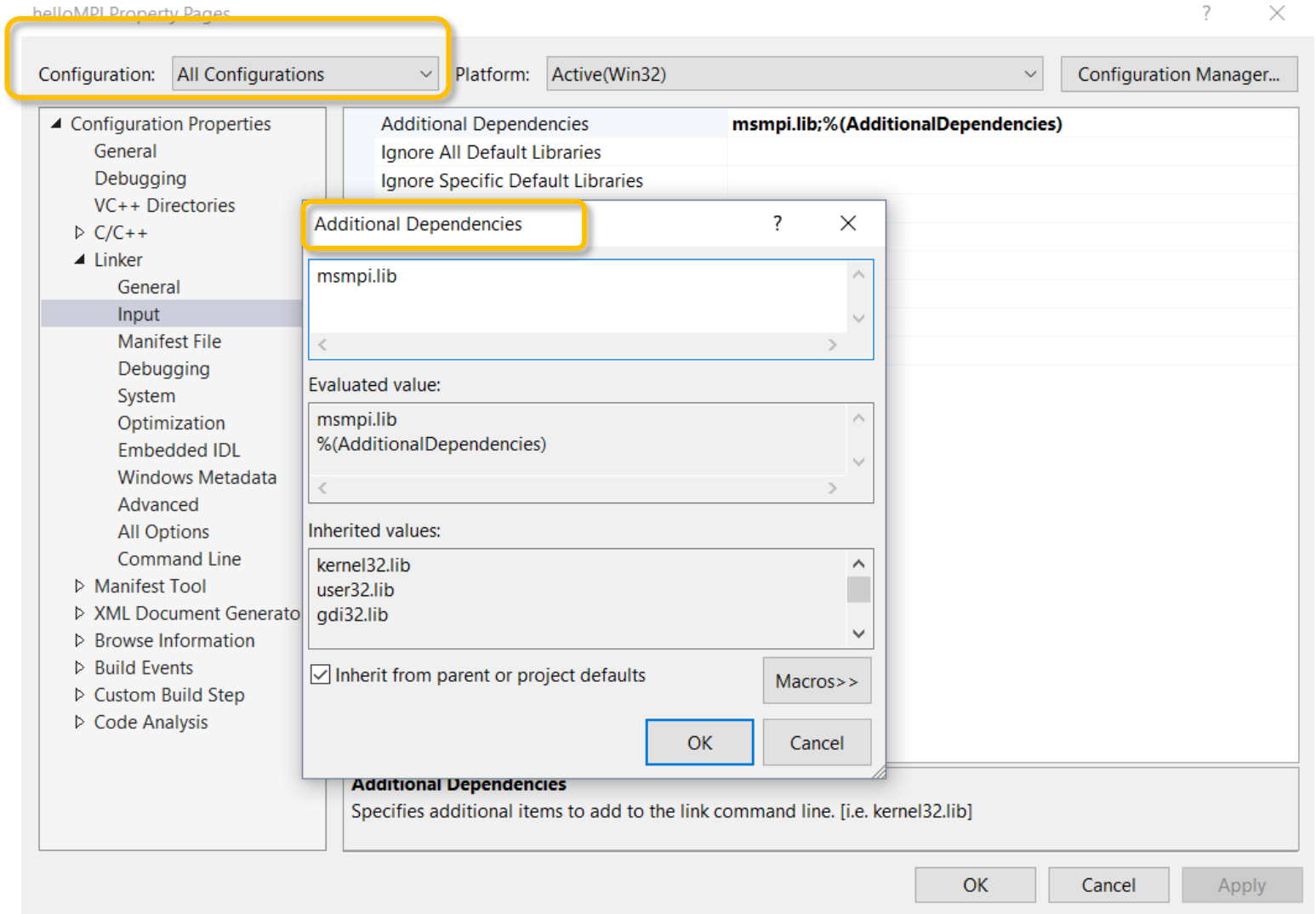
Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **4.3.:**

MS-MPI für das erstellte **Visual Studio** Projekt einrichten:

msmpi.lib unter *Additional* ***Dependencies***

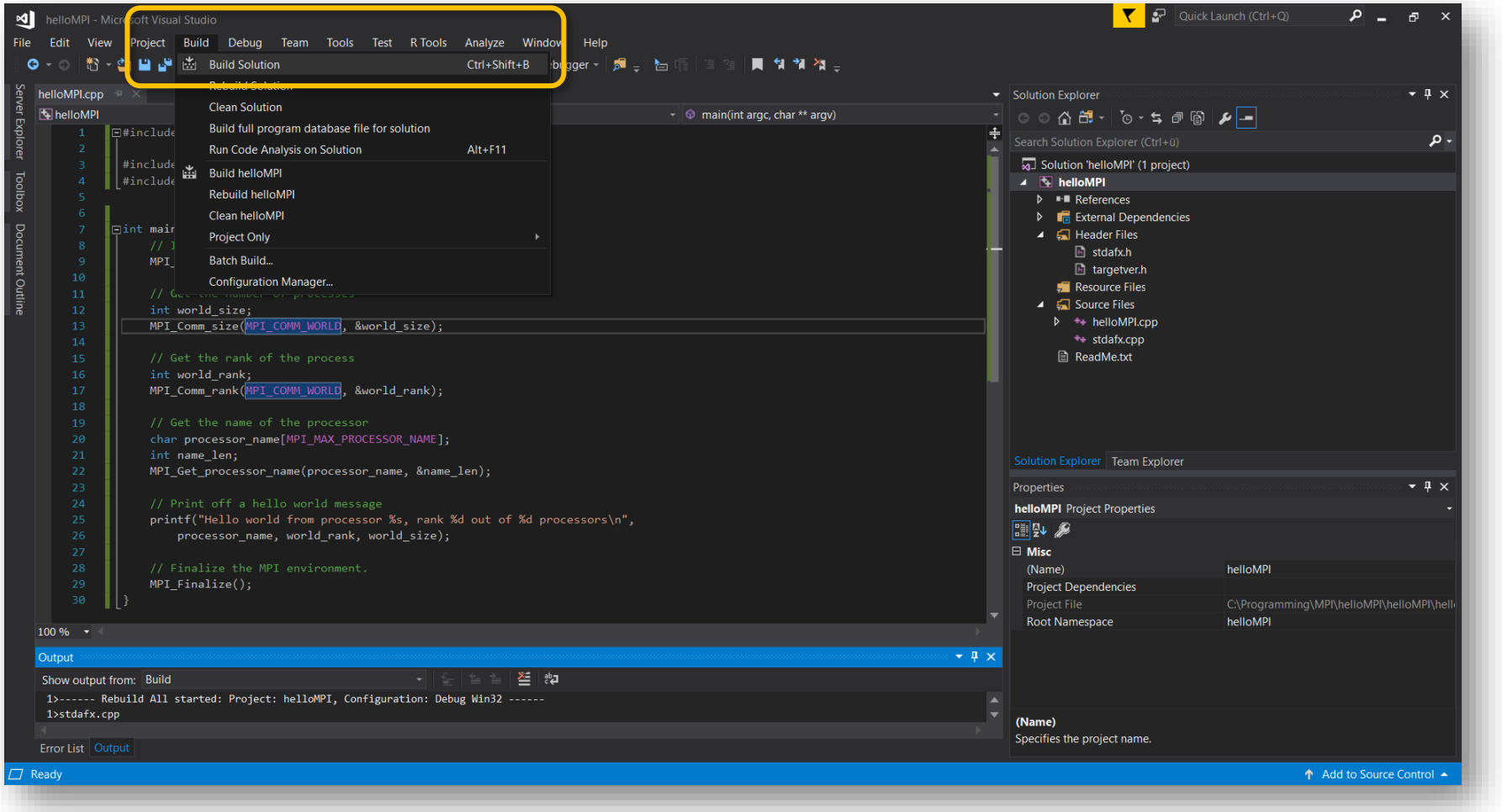


Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **5:**

Programm erstellen (Kompilieren von Quellcode)

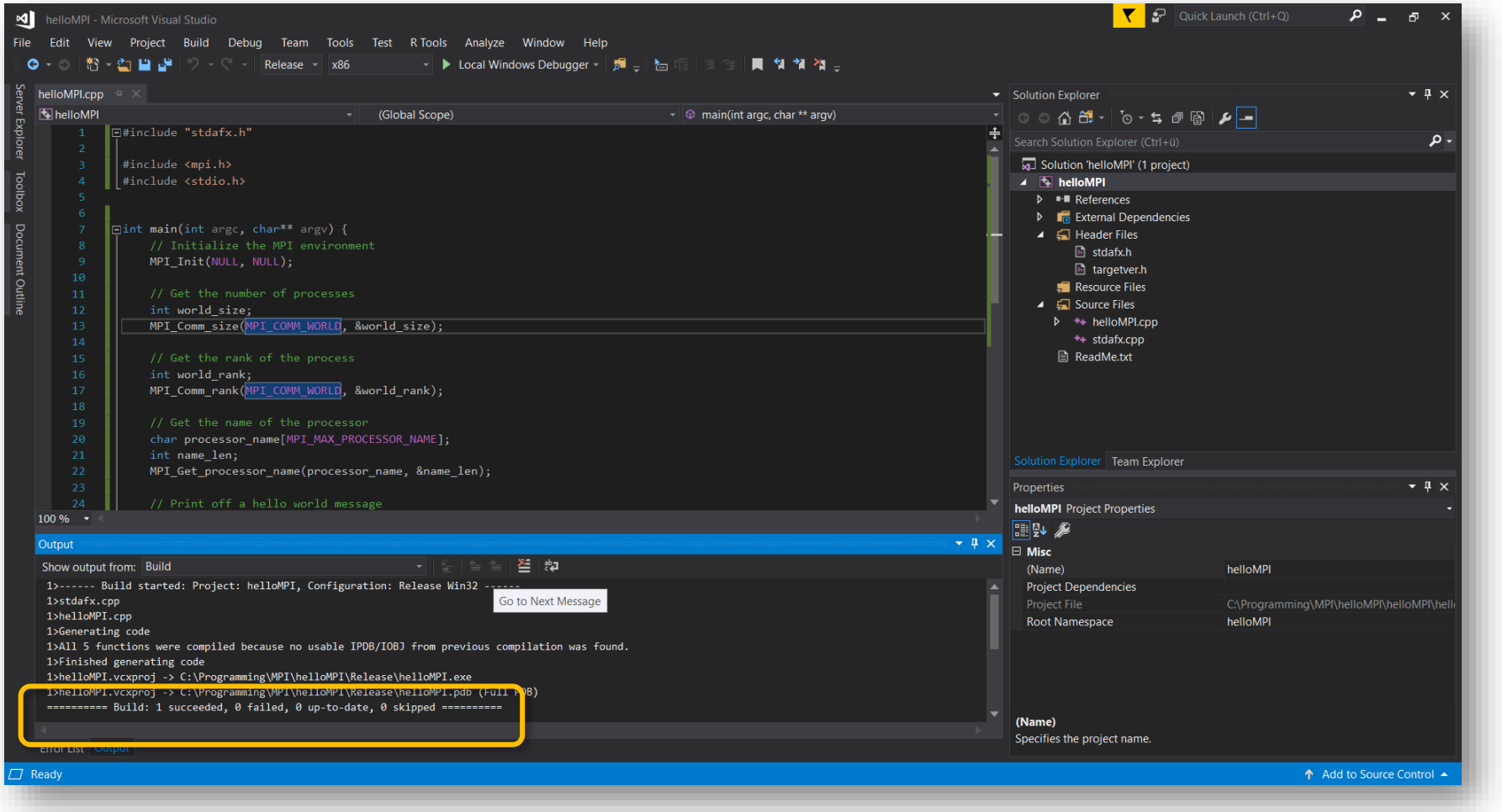


Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **5:**

Programm erstellen
(Kompilieren von Quellcode)

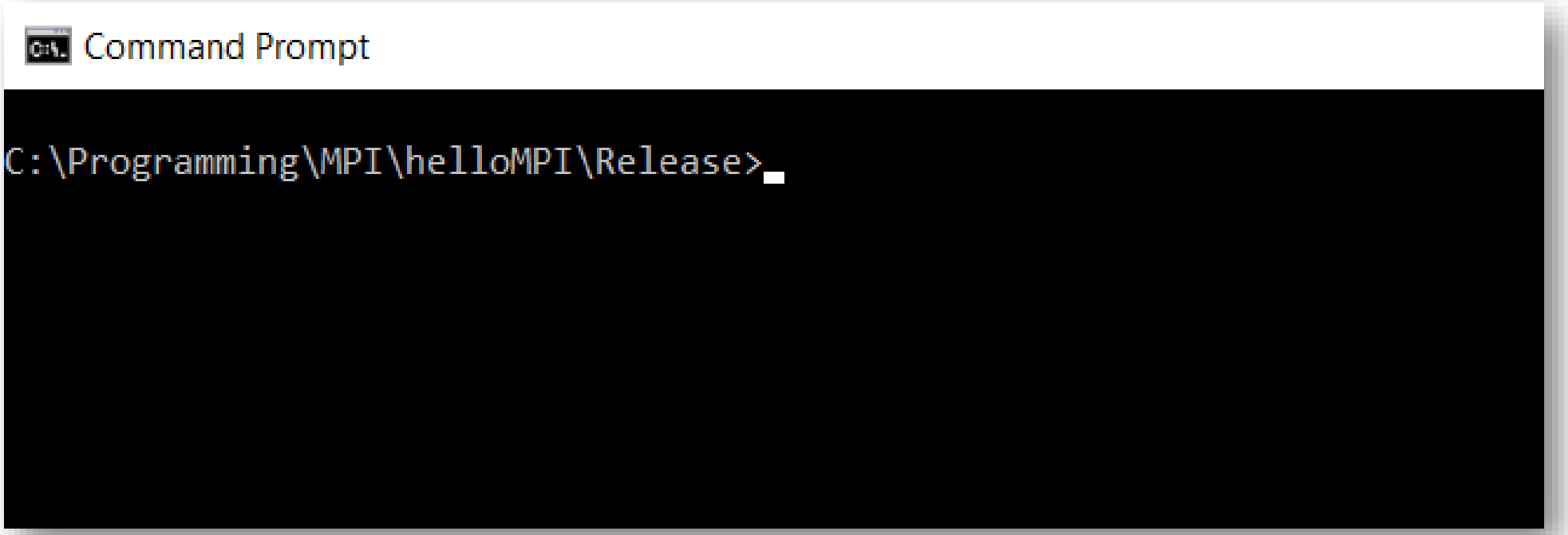


Bauhaus-
Universität
Weimar

Einrichtung unter **Windows**

Schritt **6:**

Programm ausführen

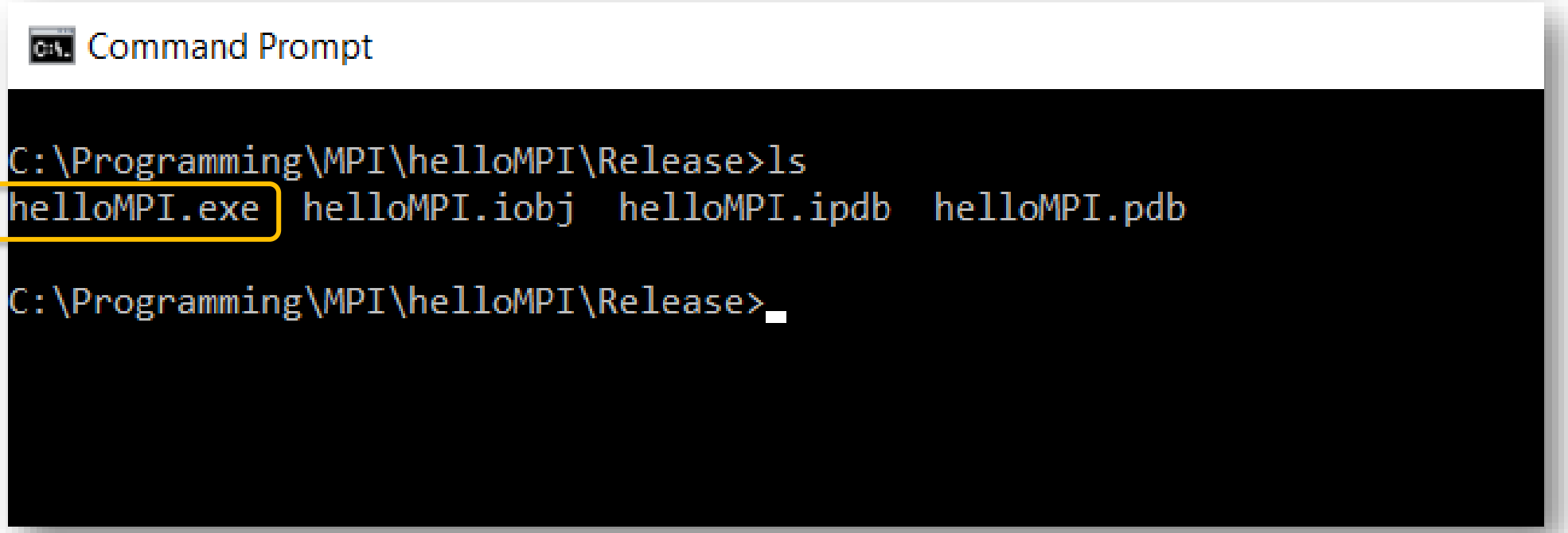


Bauhaus-
Universität
Weimar

Einrichtung unter **Windows**

Schritt **6:**

Programm ausführen



`ls`- zeigt den Inhalt eines Verzeichnisses bzw. Ordners an

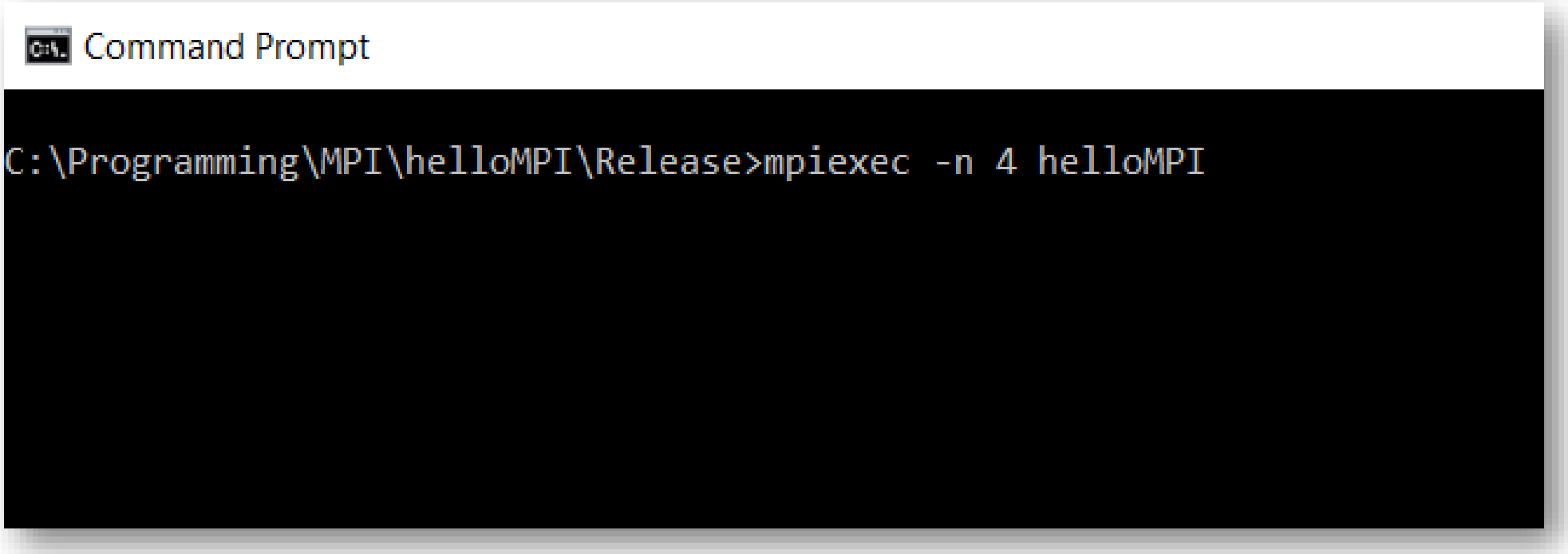
`cd`- ermöglicht das Wechseln des Arbeitsverzeichnisses

Bauhaus-
Universität
Weimar

Einrichtung unter **Windows**

Schritt **6:**

Programm ausführen



mpiexec –n *<Anzahl der Prozessen> <Programmname>**

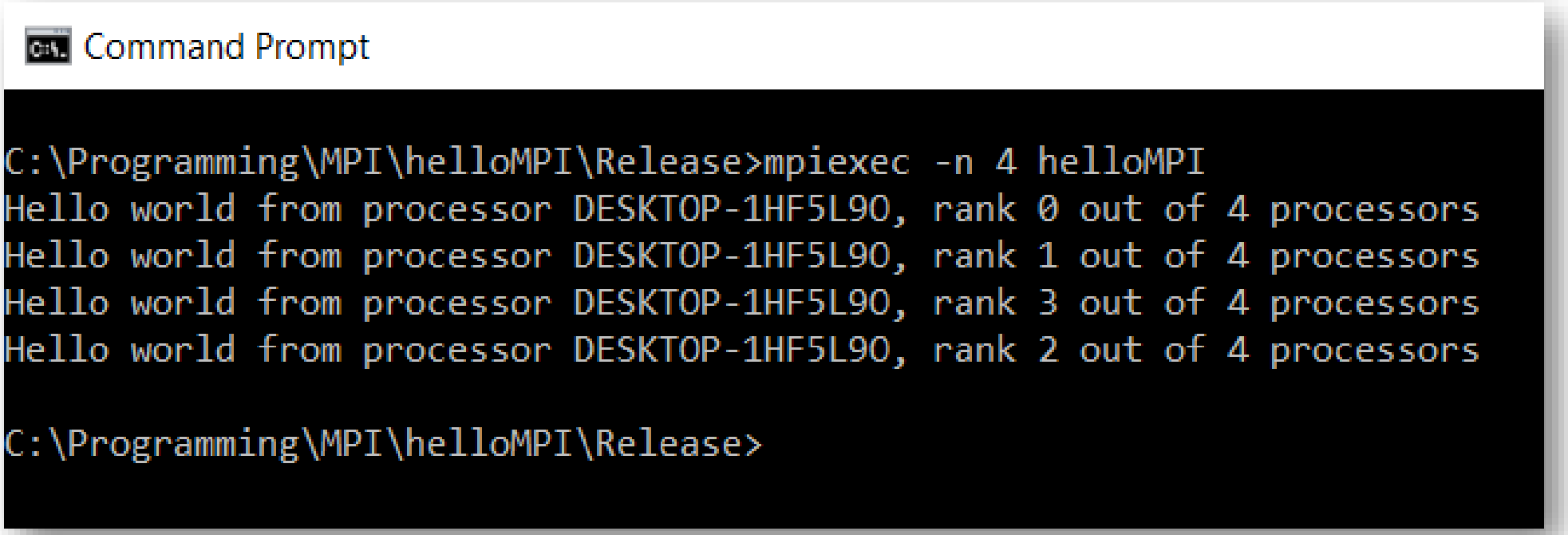
*Unterschied zu OpenMPI

Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **6:**

Programm ausführen



Project → Properties… → …

Bauhaus-Universität Weimar

Einrichtung unter **Windows**

Schritt **6:**

Programm ausführen

Alternative Parametereingabe in Visual Studio (anstatt per Konsole)