



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



Einrichtung von OpenMPI & MS-MPI



Einrichtung unter **LINUX**

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



Einrichtung unter **LINUX**

Schritt 1:

Quellcode schreiben

```
helloMPI.c
~/PVS/MPI

#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



Einrichtung unter **LINUX**

Schritt **2**:

Quellcode kompilieren

```
vajo3185@billl01:~/PVS/MPI
File Edit View Search Terminal Tabs Help
vajo3185@billl01:~/PVS/MPI
[vajo3185@billl01 MPI]$ mpic++ helloMPI.c -o helloMPI
[vajo3185@billl01 MPI]$
```

Konsole (Terminal)



Einrichtung unter **LINUX**

Schritt **3**:

Programm ausführen

The screenshot shows a terminal window titled 'vajo3185@billl01:~/PVS/MPI'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', 'Tabs', and 'Help'. Below the menu bar, there are two tabs: 'vajo3185@billl01:~/PVS/MPI' (active) and 'vajo3185@billl01:~'. The terminal content shows the prompt '[vajo3185@billl01 MPI]\$' followed by the command 'mpirun -n 4 helloMPI' with a cursor at the end. A blue bracket is drawn below the terminal window.

Konsole (Terminal)



Einrichtung unter **LINUX**

Schritt 3:

Programm ausführen

```
vajo3185@billl01:~/PVS/MPI
File Edit View Search Terminal Tabs Help
vajo3185@billl01:~/PVS/MPI x vajo3185@billl01:~
[vajo3185@billl01 MPI]$ mpirun -n 4 helloMPI
Hello world from processor billl01, rank 0 out of 4 processors
Hello world from processor billl01, rank 2 out of 4 processors
Hello world from processor billl01, rank 3 out of 4 processors
Hello world from processor billl01, rank 1 out of 4 processors
[vajo3185@billl01 MPI]$
```

Konsole (Terminal)



Einrichtung unter **LINUX**

Falls der mpic++ Compiler fehlt

```
mariya@mariya-X555LN: ~  
File Edit View Search Terminal Help  
mariya@mariya-X555LN:~$ mpic++  
Command 'mpic++' not found, but can be installed with:  
sudo apt install lam4-dev  
sudo apt install libmpich-dev  
sudo apt install libopenmpi-dev  
mariya@mariya-X555LN:~$
```

`sudo apt install libopenmpi-dev`



Einrichtung unter **LINUX**

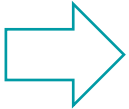
Falls mpic++ Compiler fehlt

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



Einrichtung unter **Windows**

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



Einrichtung unter Windows

Schritt 1:

MS-MPI herunterladen

The screenshot shows the Microsoft MPI documentation page. The browser address bar displays the URL: docs.microsoft.com/en-us/message-passing-interface/microsoft-mpi. The page title is "Microsoft MPI". The left sidebar shows a navigation menu with "Microsoft MPI" selected, and sub-items "MPI Release Notes" and "MPI Reference". The main content area includes a "Filter by title" search bar, the article title "Microsoft MPI", a date "03/28/2018", and a reading time "2 minutes to read". The article text describes Microsoft MPI (MS-MPI) as a Microsoft implementation of the Message Passing Interface standard for Windows. It lists several benefits: ease of porting code, security based on Active Directory, high performance, and binary compatibility. Below this, there are sections for "MS-MPI Source Code" (linking to GitHub) and "MS-MPI Downloads". The "Downloads" section lists two items: "MS-MPI v10.1.2 (new!) - see Release notes" (highlighted with a yellow box) and "Debugger for MS-MPI Applications with HPC Pack 2012 R2". A "Community Resources" section at the bottom lists links to the Windows HPC MPI Forum and the MS-MPI Team. The right sidebar contains a "Is this page helpful?" poll and a list of related topics.

Microsoft MPI

Filter by title

Microsoft MPI

MPI Release Notes

> MPI Reference

Microsoft MPI

03/28/2018 • 2 minutes to read • [User Avatars]

Microsoft MPI (MS-MPI) is a Microsoft implementation of the [Message Passing Interface standard](#) for developing and running parallel applications on the Windows platform.

MS-MPI offers several benefits:

- Ease of porting existing code that uses [MPICH](#).
- Security based on Active Directory Domain Services.
- High performance on the Windows operating system.
- Binary compatibility across different types of interconnectivity options.

MS-MPI Source Code

Microsoft MPI source code is available on [GitHub](#).

MS-MPI Downloads

The following are current downloads for MS-MPI:

- **MS-MPI v10.1.2 (new!) - see [Release notes](#)**
- [Debugger for MS-MPI Applications with HPC Pack 2012 R2](#)

Earlier versions of MS-MPI are available from the [Microsoft Download Center](#).

Community Resources

- [Windows HPC MPI Forum](#)
- [Contact the MS-MPI Team](#)

Download PDF

Is this page helpful?
Yes No

In this article

- [MS-MPI Source Code](#)
- [MS-MPI Downloads](#)
- [Community Resources](#)
- [Microsoft High Performance Computing Resources](#)
- [Related Topics](#)

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



Einrichtung unter **Windows**

Schritt **1**:

MS-MPI herunterladen

Choose the download you want

<input type="checkbox"/> File Name	Size
<input type="checkbox"/> msmpisetup.exe	7.5 MB
<input checked="" type="checkbox"/> msmpisdk.msi	2.2 MB

Download Summary:
KBMBGB

1. msmpisdk.msi

Total Size: 2.2 MB

Next

+

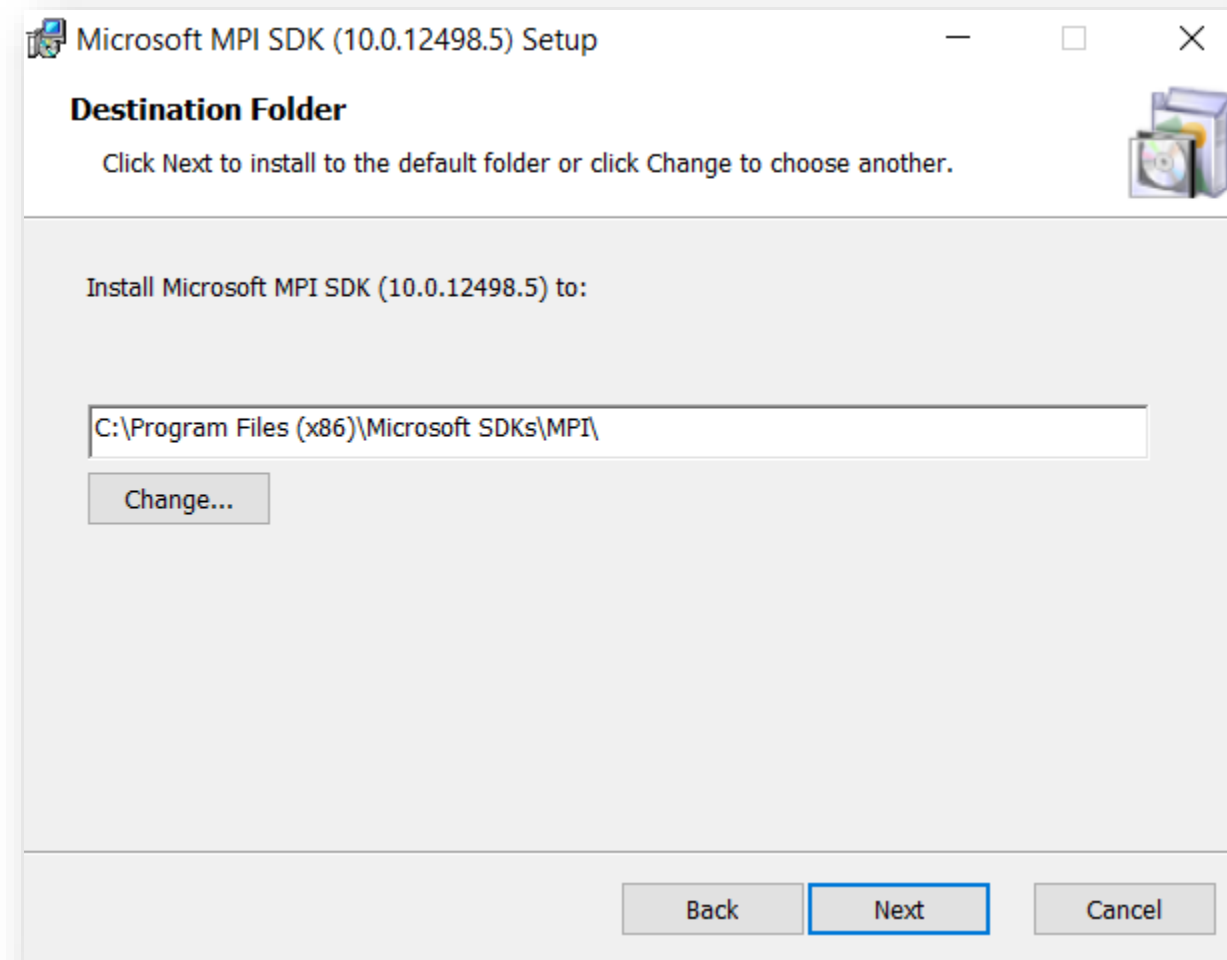
 Install instructions



Einrichtung unter Windows

Schritt 2:

MS-MPI installieren





Einrichtung unter **Windows**

Test nach Schritt 2:

```
Command Prompt
C:\>set MSMPI_
```



Einrichtung unter **Windows**

Test nach Schritt 2:

```
Command Prompt

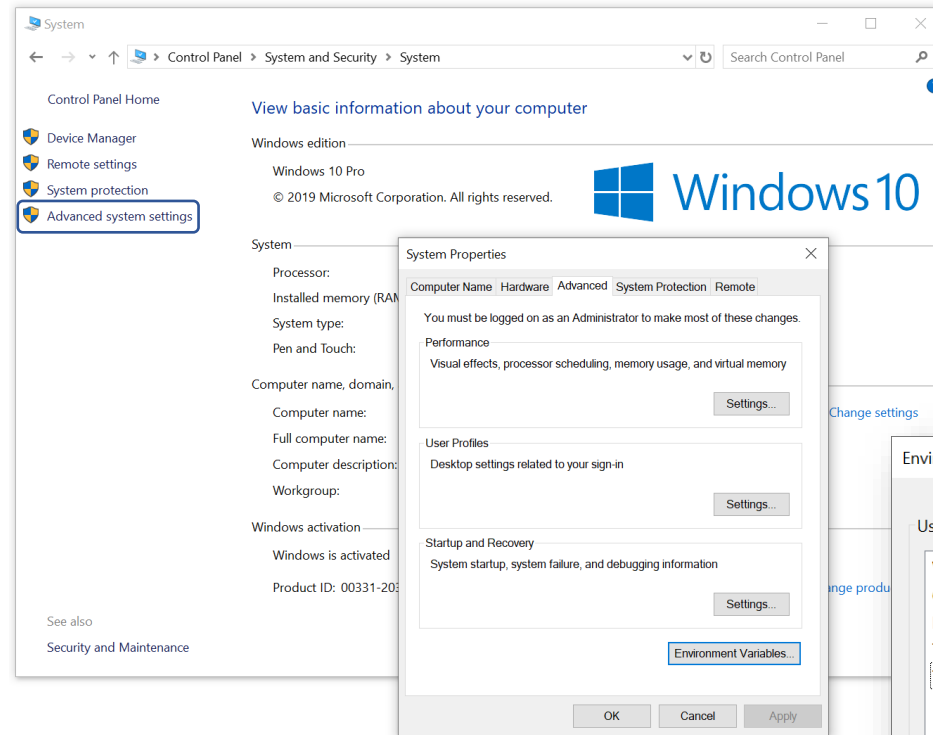
C:\>set MSMPI
MSMPI_BENCHMARKS=C:\Program Files\Microsoft MPI\Benchmarks\
MSMPI_BIN=C:\Program Files\Microsoft MPI\Bin\
MSMPI_INC=C:\Program Files (x86)\Microsoft SDKs\MPI\Include\
MSMPI_LIB32=C:\Program Files (x86)\Microsoft SDKs\MPI\Lib\x86\
MSMPI_LIB64=C:\Program Files (x86)\Microsoft SDKs\MPI\Lib\x64\

C:\>_
```



Einrichtung unter Windows

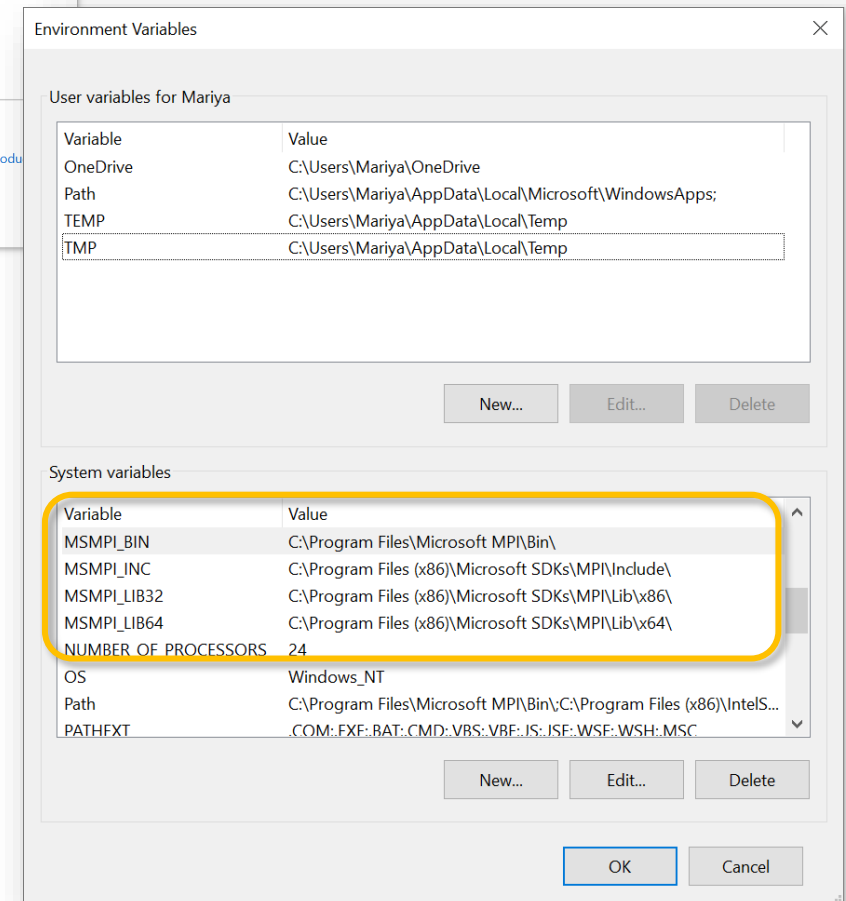
Test nach Schritt 2:



Command Prompt

```
C:\>set MSMPI
MSMPI_BENCHMARKS=C:\Program Files\Microsoft MPI\Benchmarks\
MSMPI_BIN=C:\Program Files\Microsoft MPI\Bin\
MSMPI_INC=C:\Program Files (x86)\Microsoft SDKs\MPI\Include\
MSMPI_LIB32=C:\Program Files (x86)\Microsoft SDKs\MPI\Lib\x86\
MSMPI_LIB64=C:\Program Files (x86)\Microsoft SDKs\MPI\Lib\x64\
```

```
C:\>_
```



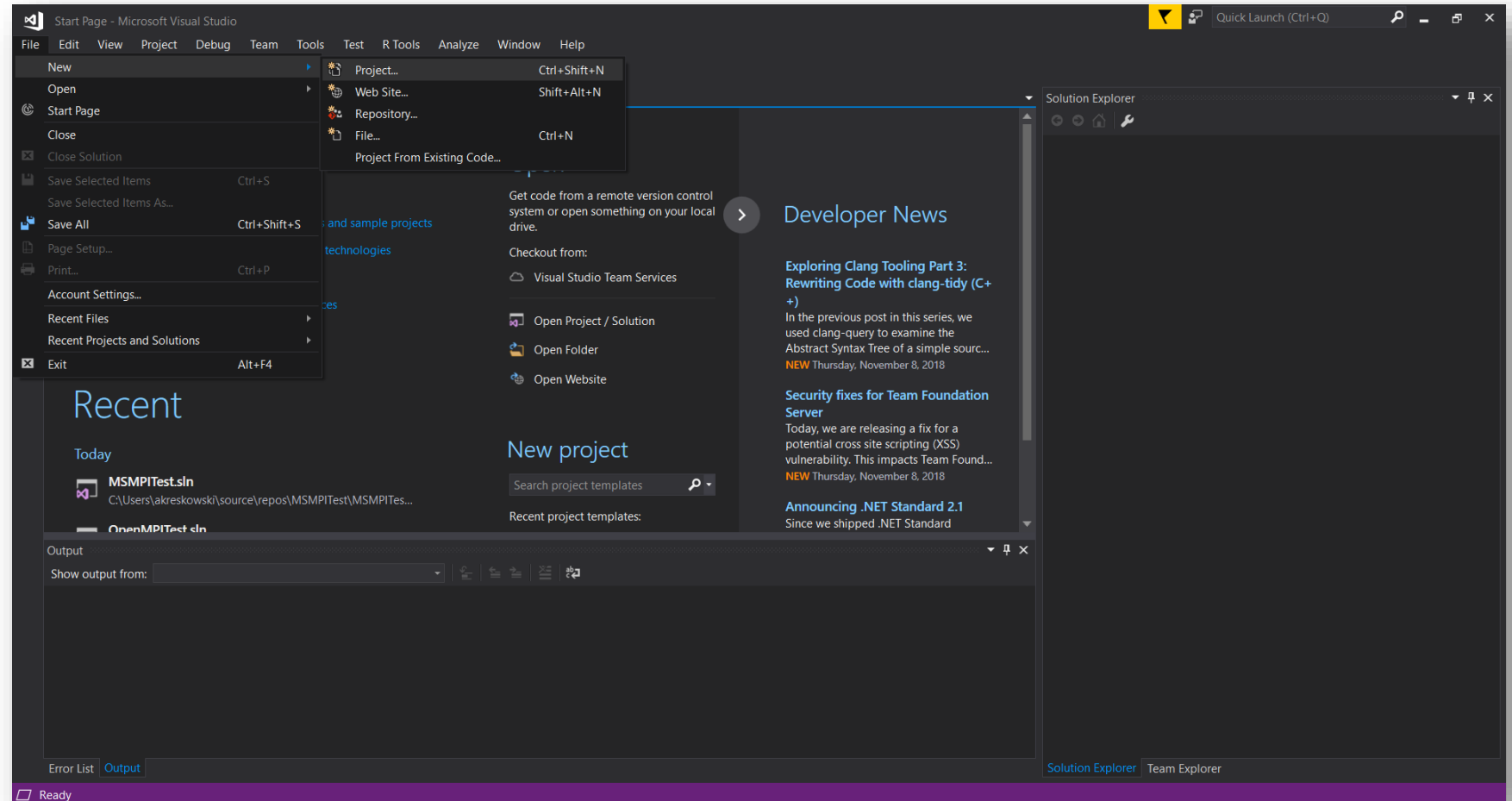


Einrichtung unter Windows

Schritt 3:

Projekt erstellen in
Visual Studio*

*Bitte nicht mit
Visual Studio Code
verwechseln



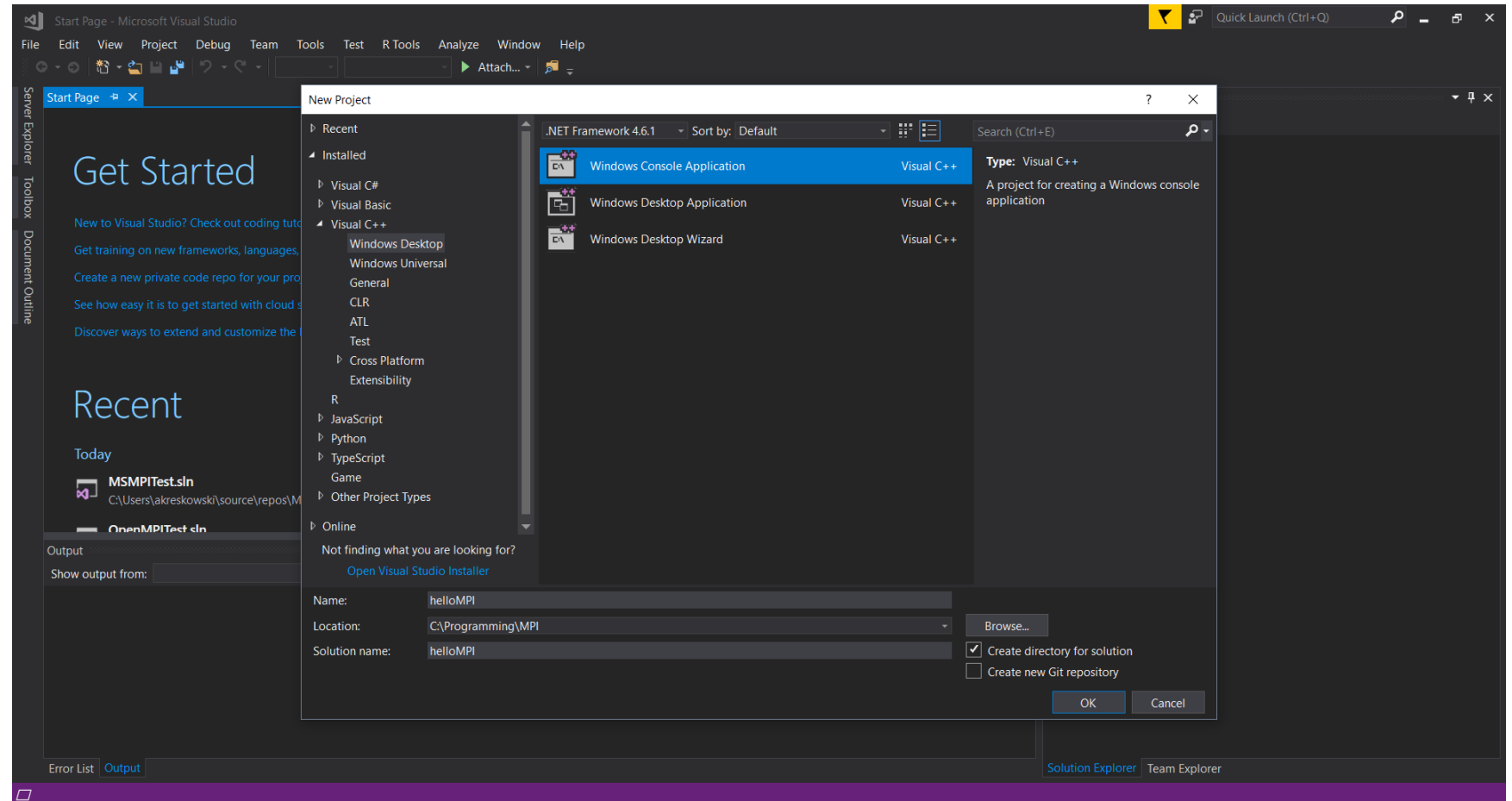


Einrichtung unter Windows

Schritt 3:

Projekt erstellen in
Visual Studio*

*Bitte nicht mit
Visual Studio Code
verwechseln



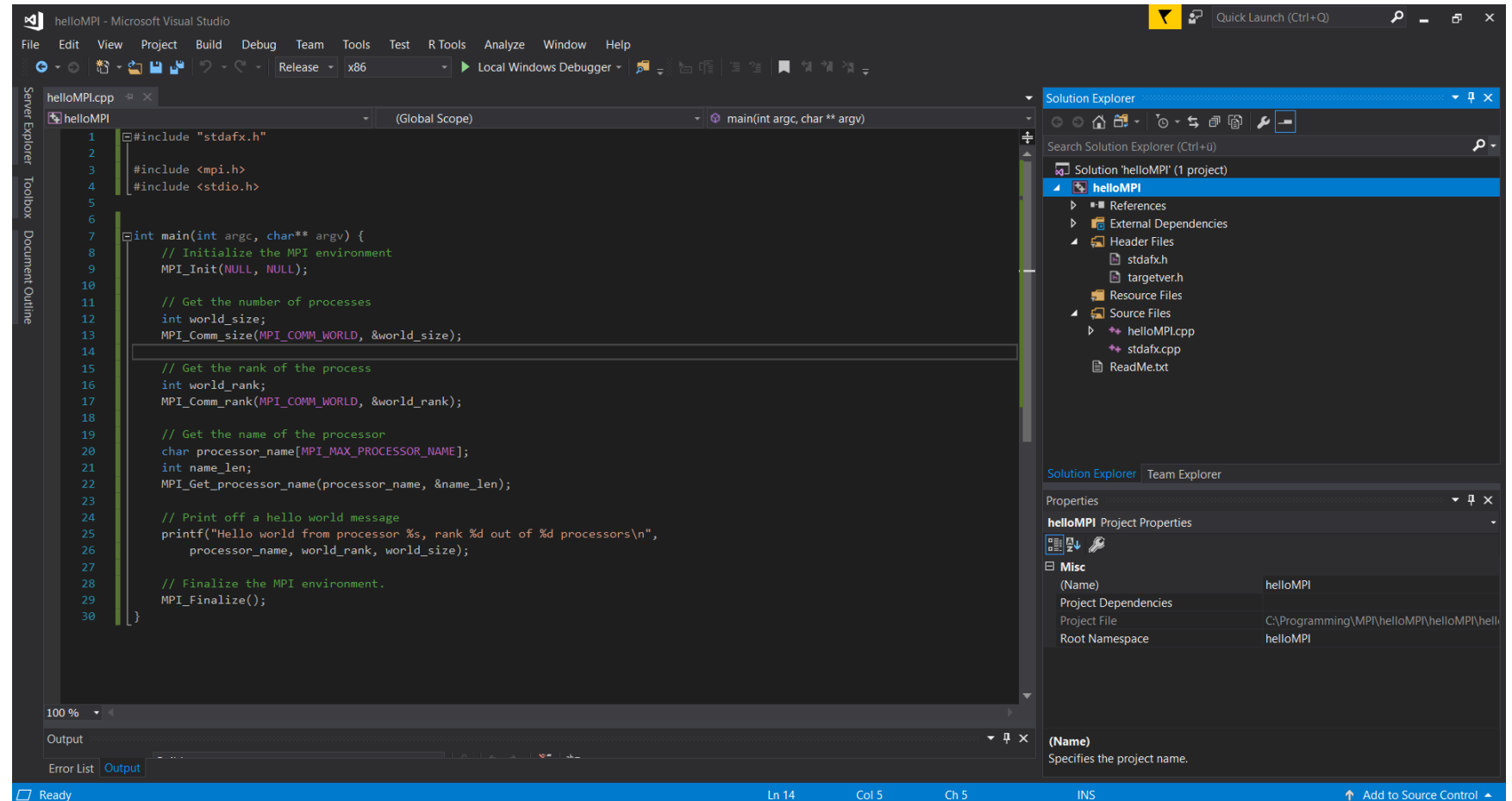


Einrichtung unter Windows

Schritt 3:

Projekt erstellen in
Visual Studio*

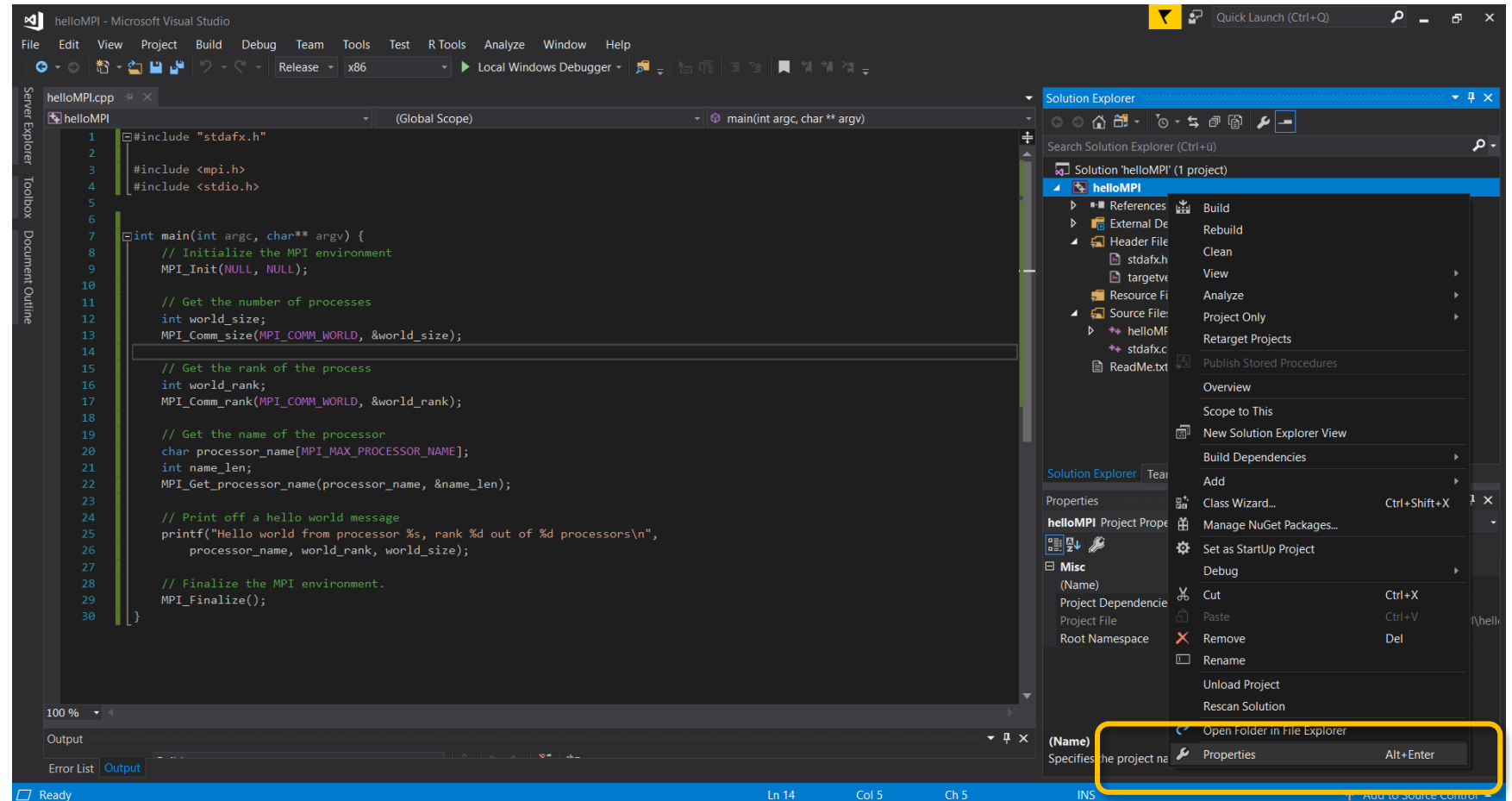
*Bitte nicht mit
Visual Studio Code
verwechseln





Einrichtung unter Windows

Schritt 4:
MS-MPI für das erstellte
Visual Studio Projekt
einrichten



Project → Properties...



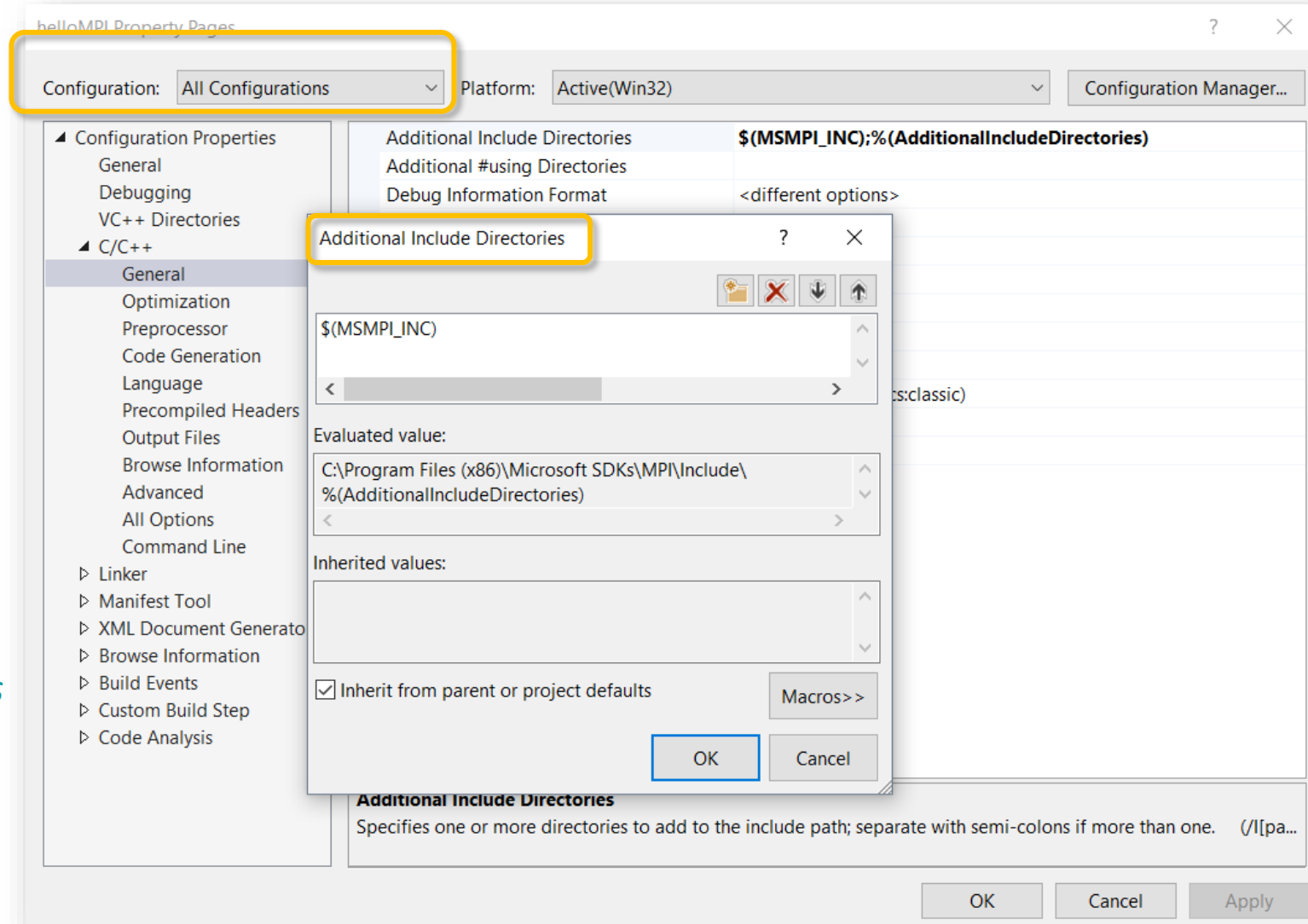
Project → Properties... → ...

Einrichtung unter Windows

Schritt 4.1.:

MS-MPI für das erstellte
Visual Studio Projekt
einrichten:

\$(MSMPI_INC) unter
Additional Include Directories





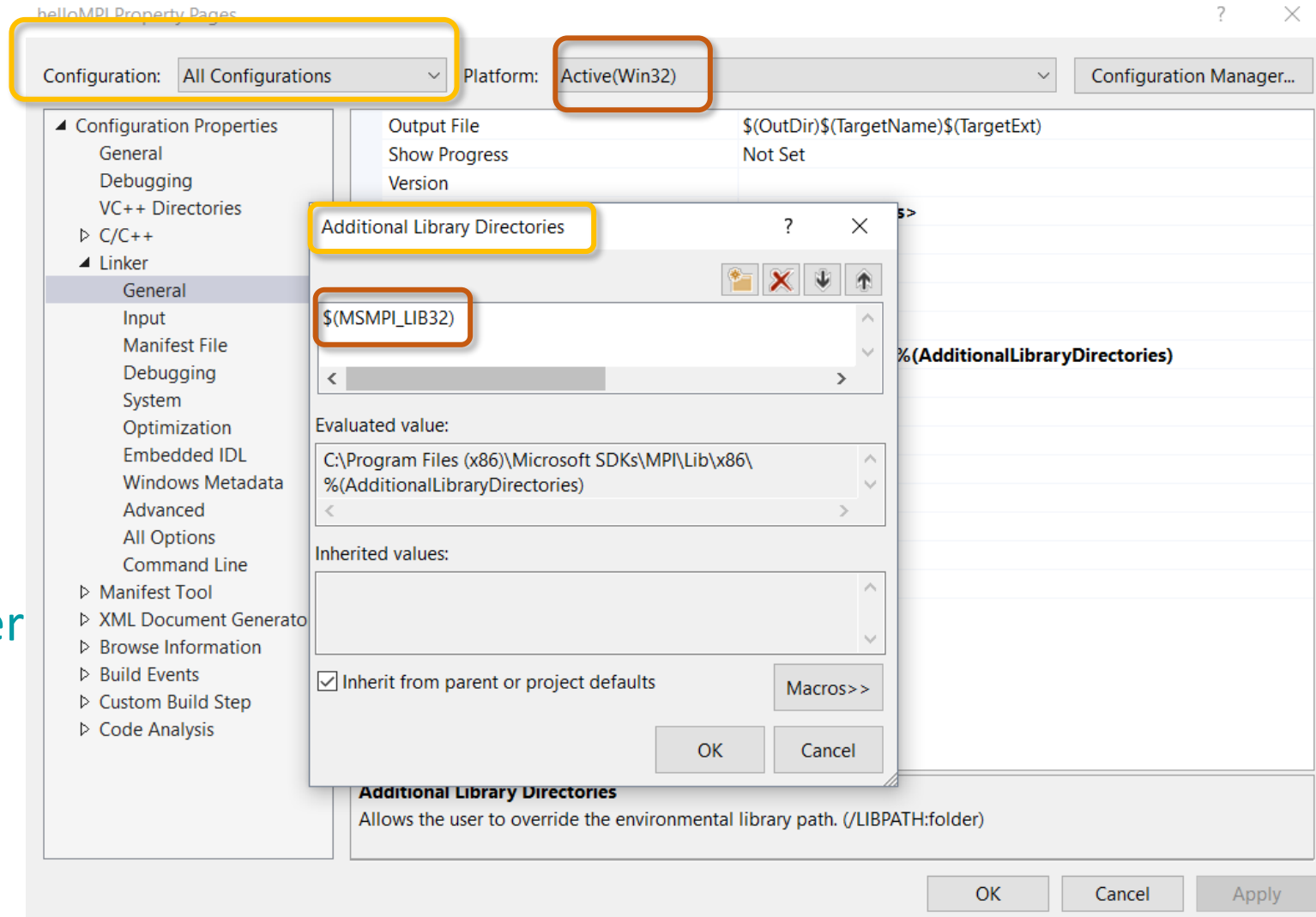
Project → Properties... → ...

Einrichtung unter Windows

Schritt 4.2.:

MS-MPI für das erstellte
Visual Studio Projekt
einrichten:

`$(MSMPI_LIB32)` unter
Additional Library Directories





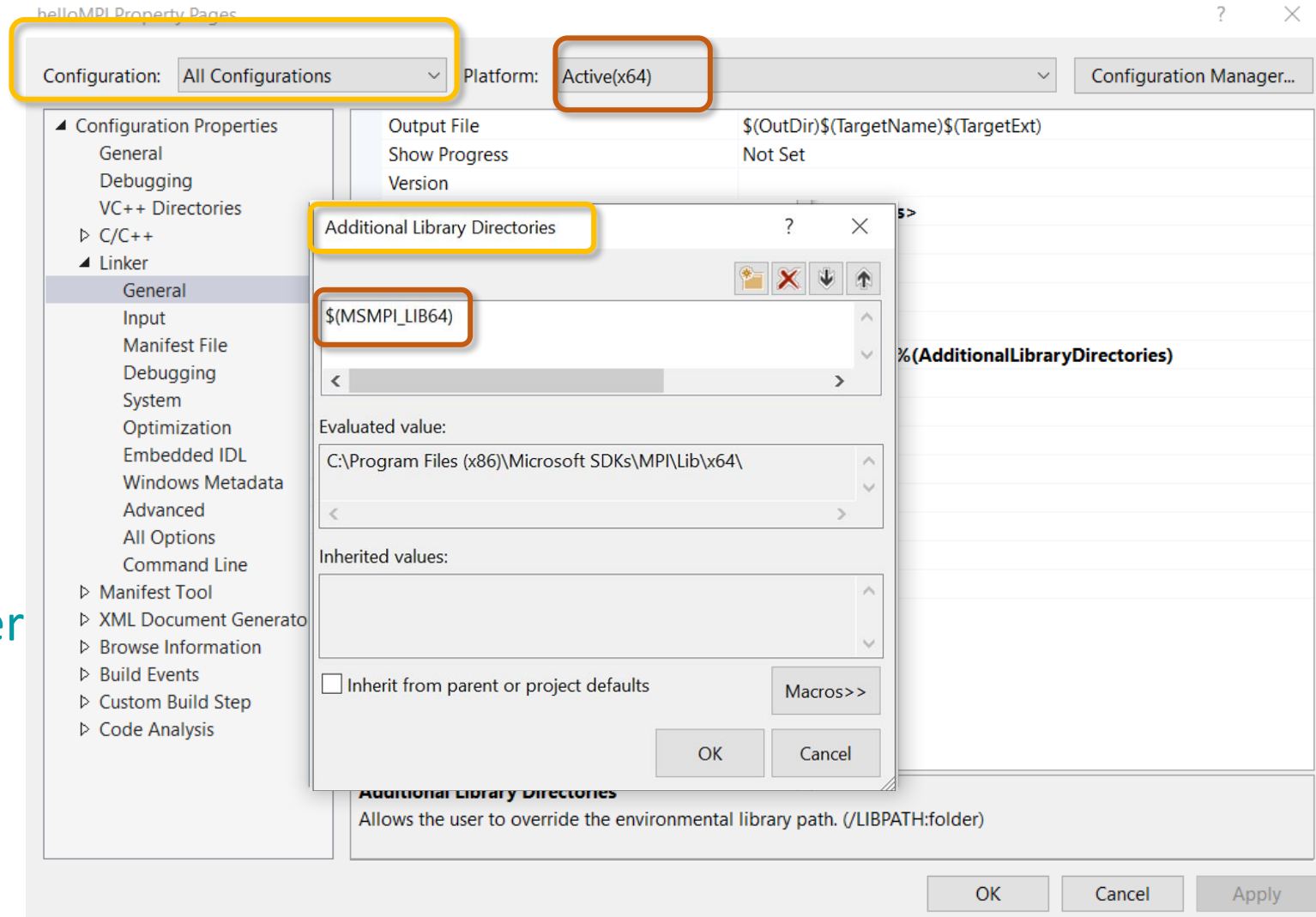
Project → Properties... → ...

Einrichtung unter Windows

Schritt 4.2.:

MS-MPI für das erstellte
Visual Studio Projekt
einrichten:

`$(MSMPI_LIB64)` unter
Additional Library Directories





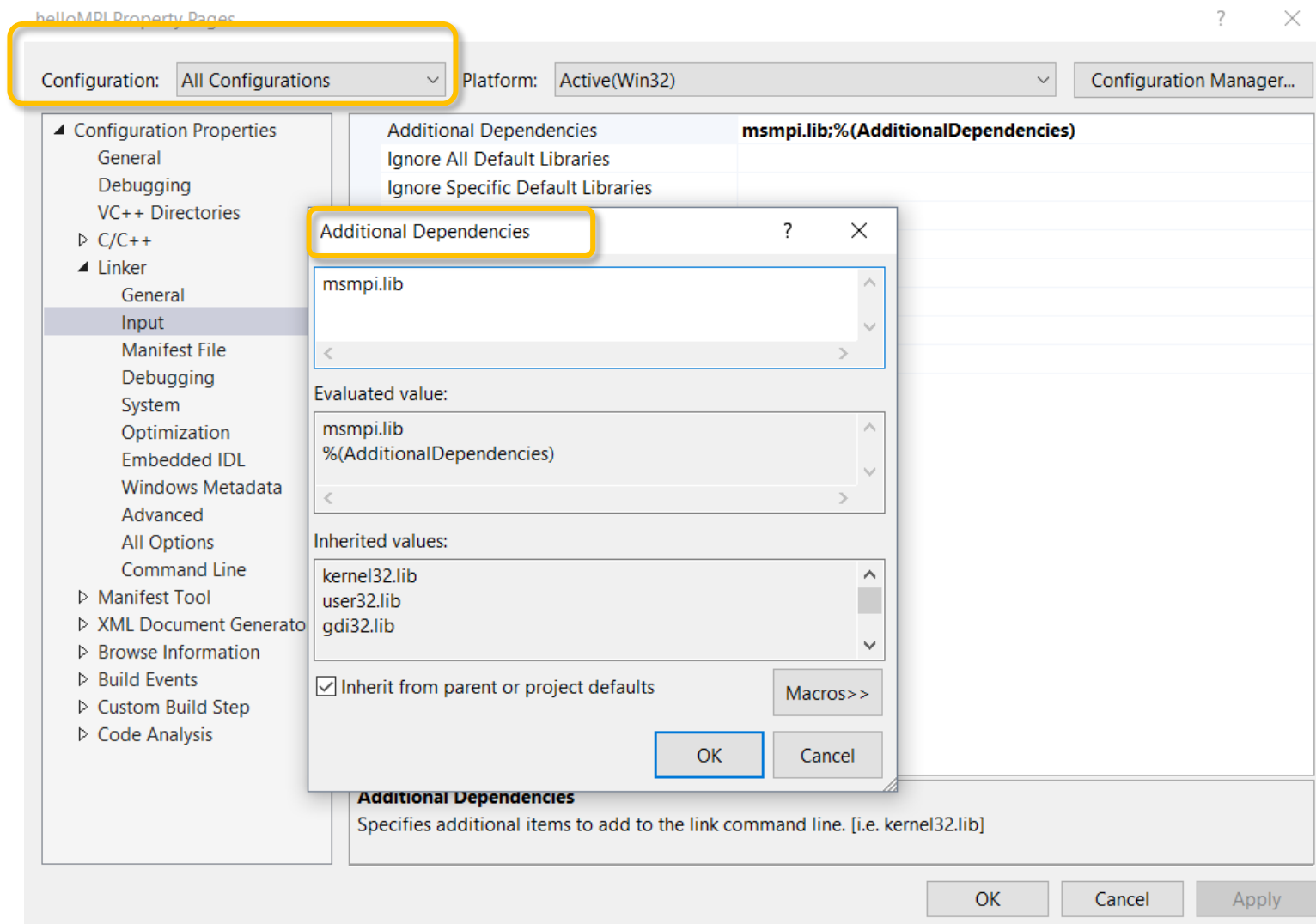
Project → Properties... → ...

Einrichtung unter Windows

Schritt 4.3.:

MS-MPI für das erstellte
Visual Studio Projekt
einrichten:

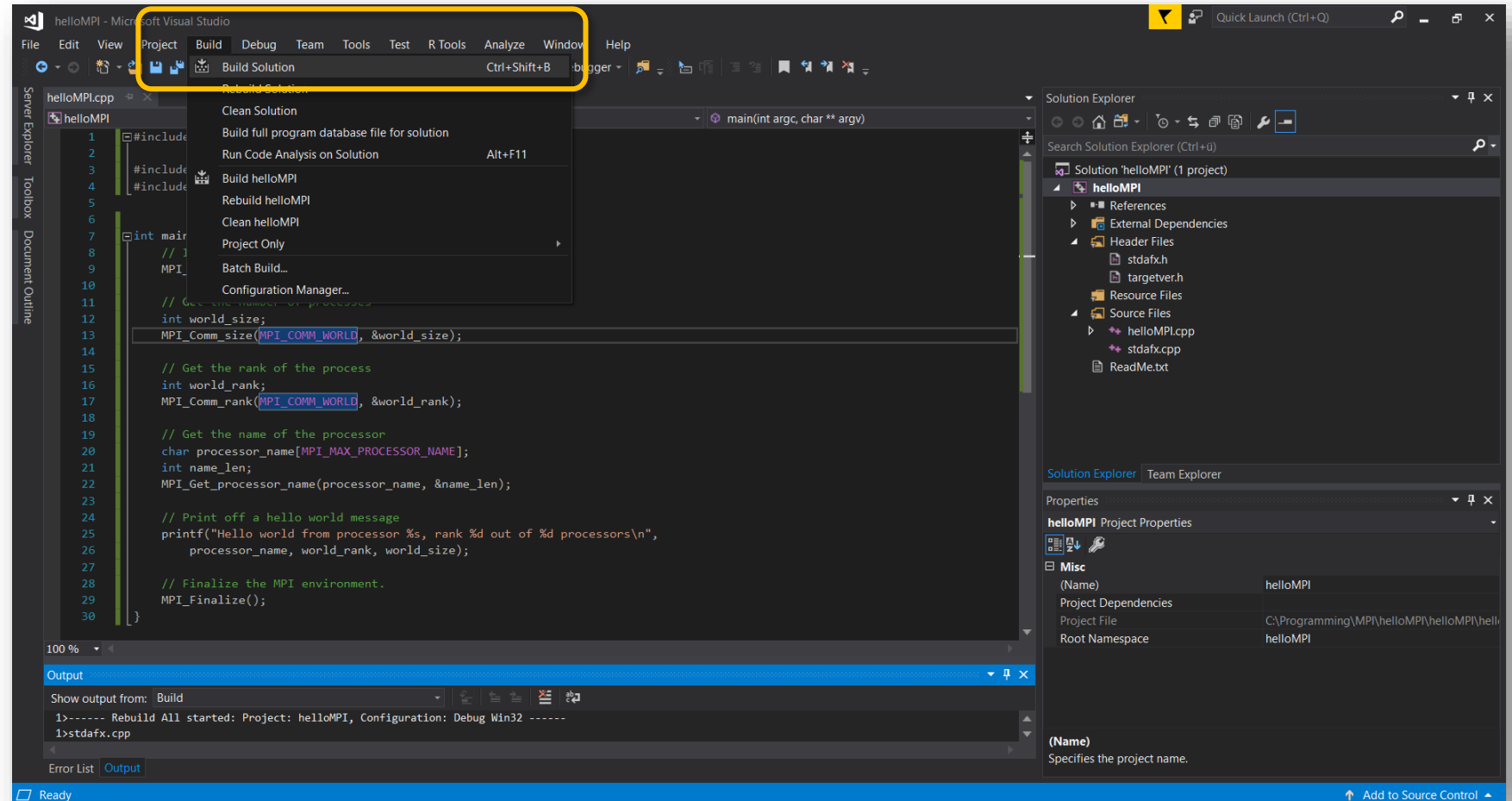
msmpi.lib unter
Additional Dependencies





Einrichtung unter Windows

Schritt 5: Programm erstellen (Kompilieren von Quellcode)





Einrichtung unter Windows

Schritt 5: Programm erstellen (Kompilieren von Quellcode)

```
1 #include "stdafx.h"
2
3 #include <mpi.h>
4 #include <stdio.h>
5
6
7 int main(int argc, char** argv) {
8     // Initialize the MPI environment
9     MPI_Init(NULL, NULL);
10
11     // Get the number of processes
12     int world_size;
13     MPI_Comm_size(MPI_COMM_WORLD, &world_size);
14
15     // Get the rank of the process
16     int world_rank;
17     MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
18
19     // Get the name of the processor
20     char processor_name[MPI_MAX_PROCESSOR_NAME];
21     int name_len;
22     MPI_Get_processor_name(processor_name, &name_len);
23
24     // Print off a hello world message
```

Output

```
1>----- Build started: Project: helloMPI, Configuration: Release Win32 -----
1>stdafx.cpp
1>helloMPI.cpp
1>Generating code
1>All 5 functions were compiled because no usable IPDB(IOB) from previous compilation was found.
1>Finished generating code
1>helloMPI.vcxproj -> C:\Programming\MPI\helloMPI\Release\helloMPI.exe
1>helloMPI.vcxproj -> C:\Programming\MPI\helloMPI\Release\helloMPI.pdb (Full PDB)
1>----- Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped -----
```

Go to Next Message

Properties

helloMPI Project Properties

Misc	
(Name)	helloMPI
Project Dependencies	
Project File	C:\Programming\MPI\helloMPI\helloMPI\hell
Root Namespace	helloMPI


(Name)
Specifies the project name.



Einrichtung unter **Windows**

Schritt 6:

Programm ausführen

 Command Prompt

```
C:\Programming\MPI\helloMPI\Release>
```



Einrichtung unter Windows

Schritt 6:
Programm ausführen

Command Prompt

```
C:\Programming\MPI\helloMPI\Release>ls  
helloMPI.exe helloMPI.iobj helloMPI.ipdb helloMPI.pdb  
C:\Programming\MPI\helloMPI\Release>
```

ls - zeigt den Inhalt eines Verzeichnisses bzw. Ordners an
cd - ermöglicht das Wechseln des Arbeitsverzeichnisses



Einrichtung unter Windows

Schritt 6:

Programm ausführen

Command Prompt

```
C:\Programming\MPI\helloMPI\Release>mpiexec -n 4 helloMPI
```

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

*Unterschied zu OpenMPI



Einrichtung unter Windows

Schritt 6:

Programm ausführen



Command Prompt

```
C:\Programming\MPI\helloMPI\Release>mpiexec -n 4 helloMPI
Hello world from processor DESKTOP-1HF5L90, rank 0 out of 4 processors
Hello world from processor DESKTOP-1HF5L90, rank 1 out of 4 processors
Hello world from processor DESKTOP-1HF5L90, rank 3 out of 4 processors
Hello world from processor DESKTOP-1HF5L90, rank 2 out of 4 processors

C:\Programming\MPI\helloMPI\Release>
```



Project → Properties... → ...

Einrichtung unter Windows

Schritt 6:

Programm ausführen

Alternative
Parametereingabe
in Visual Studio
(anstatt per Konsole)

