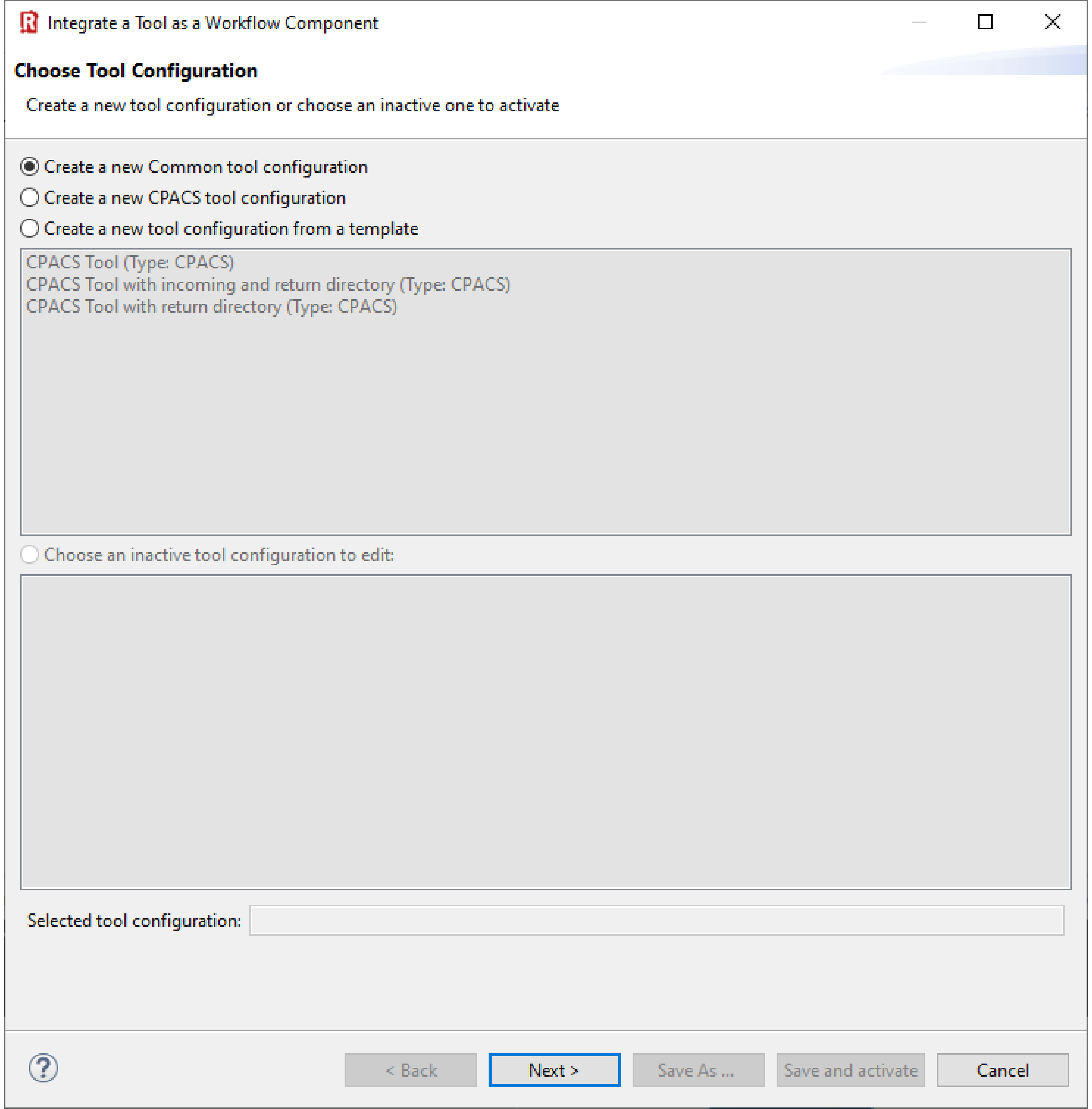


Integrate a Tool as a Workflow Component

Choose Tool Configuration

Create a new tool configuration or choose an inactive one to activate

● Create a new Common tool configuration

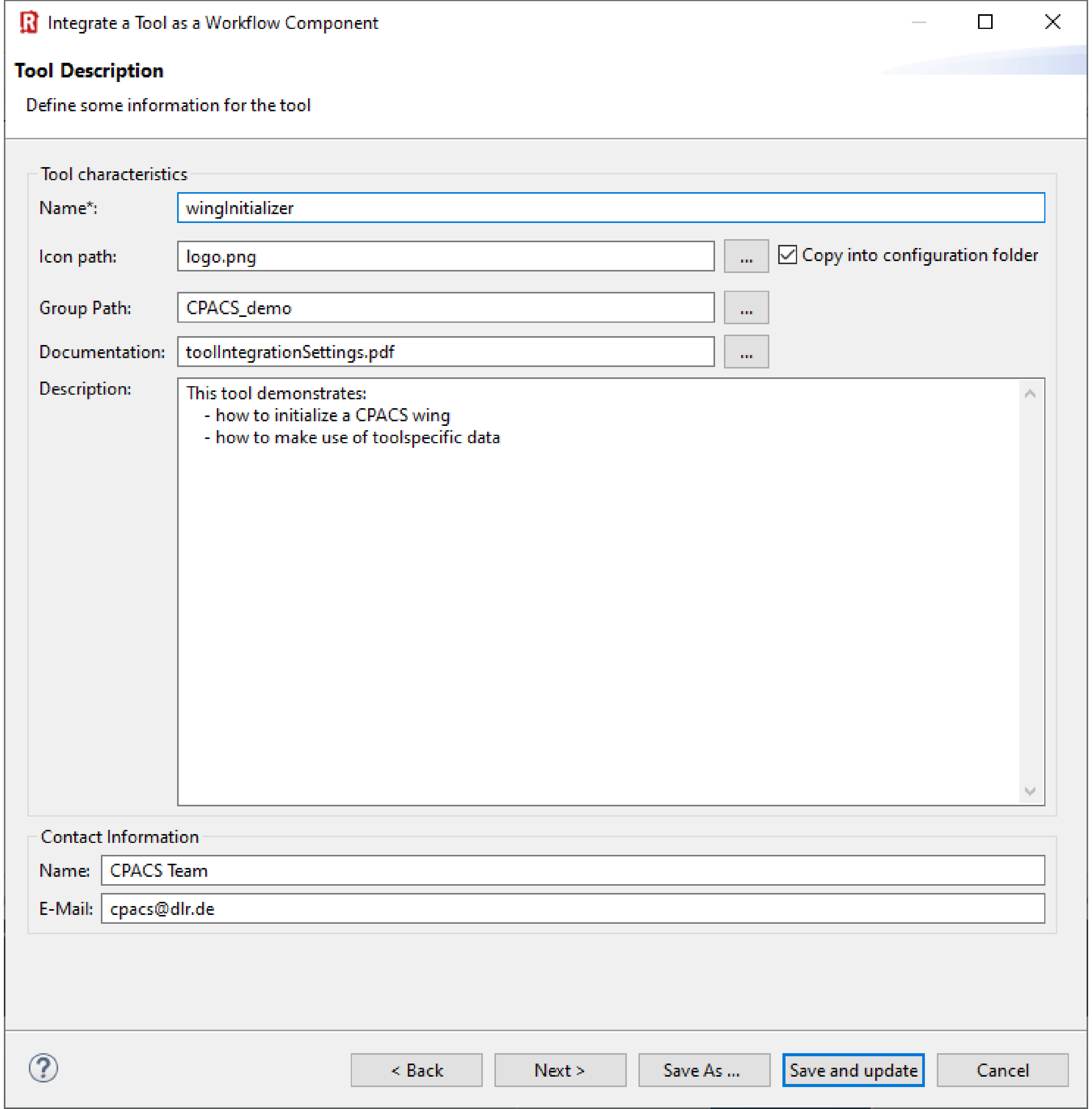
○ Create a new CPACS tool configuration

○ Create a new tool configuration from a template

CPACS Tool (Type: CPACS)
CPACS Tool with incoming and return directory (Type: CPACS)
CPACS Tool with return directory (Type: CPACS)

○ Choose an inactive tool configuration to edit:

Selected tool configuration:



Integrate a Tool as a Workflow Component

Tool Description

Define some information for the tool

Tool characteristics

Name*: wingInitializer

Icon path: logo.png ☑ Copy into configuration folder

Group Path: CPACS_demo

Documentation: toolIntegrationSettings.pdf

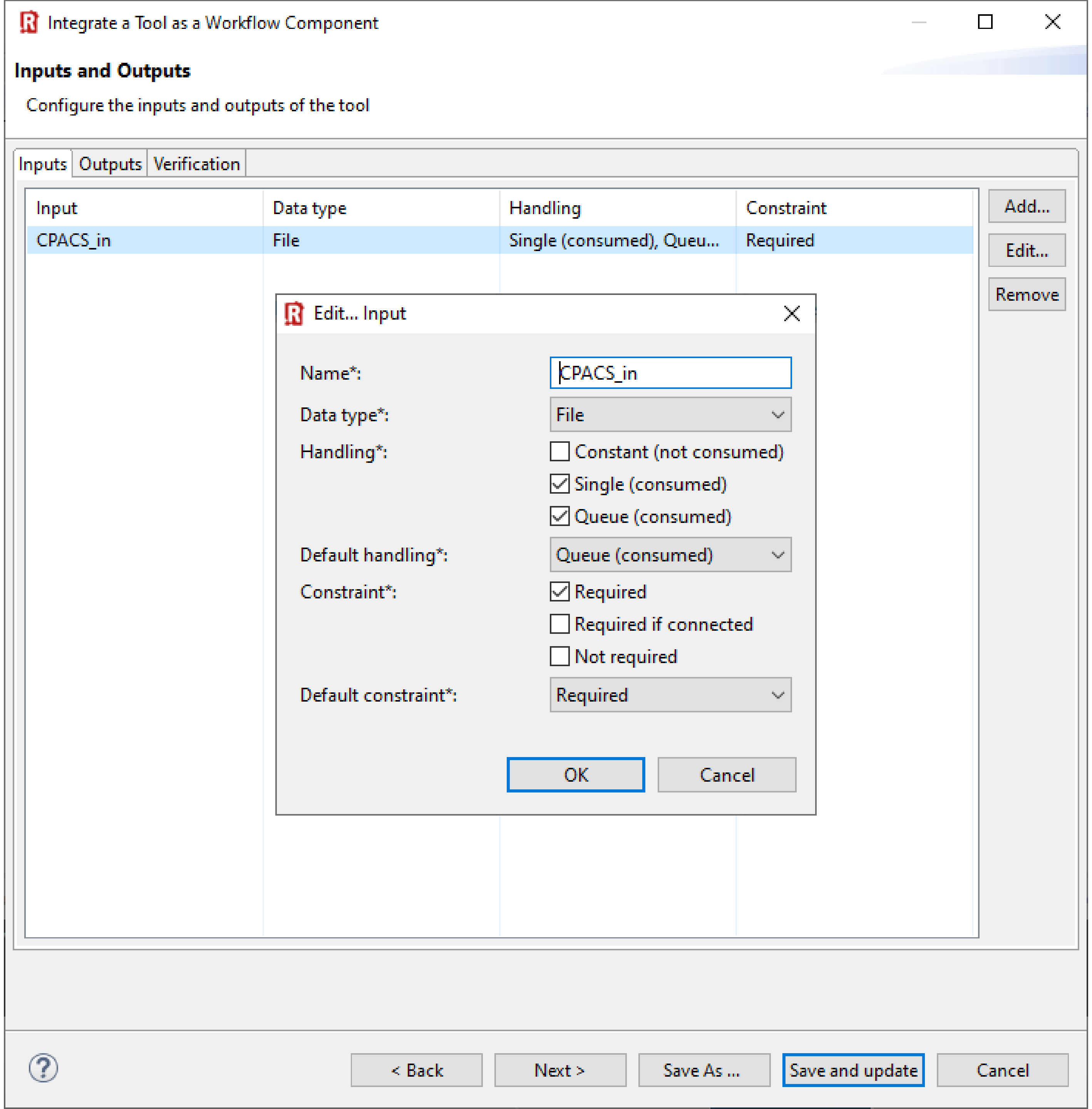
Description:

This tool demonstrates:
- how to initialize a CPACS wing
- how to make use of toolspecific data

Contact Information

Name: CPACS Team

E-Mail: cpacs@dlr.de



Inputs and Outputs

Configure the inputs and outputs of the tool

Input	Data type	Handling	Constraint
CPACS_in	File	Single (consumed), Queu...	Required

Name*: CPACS_in

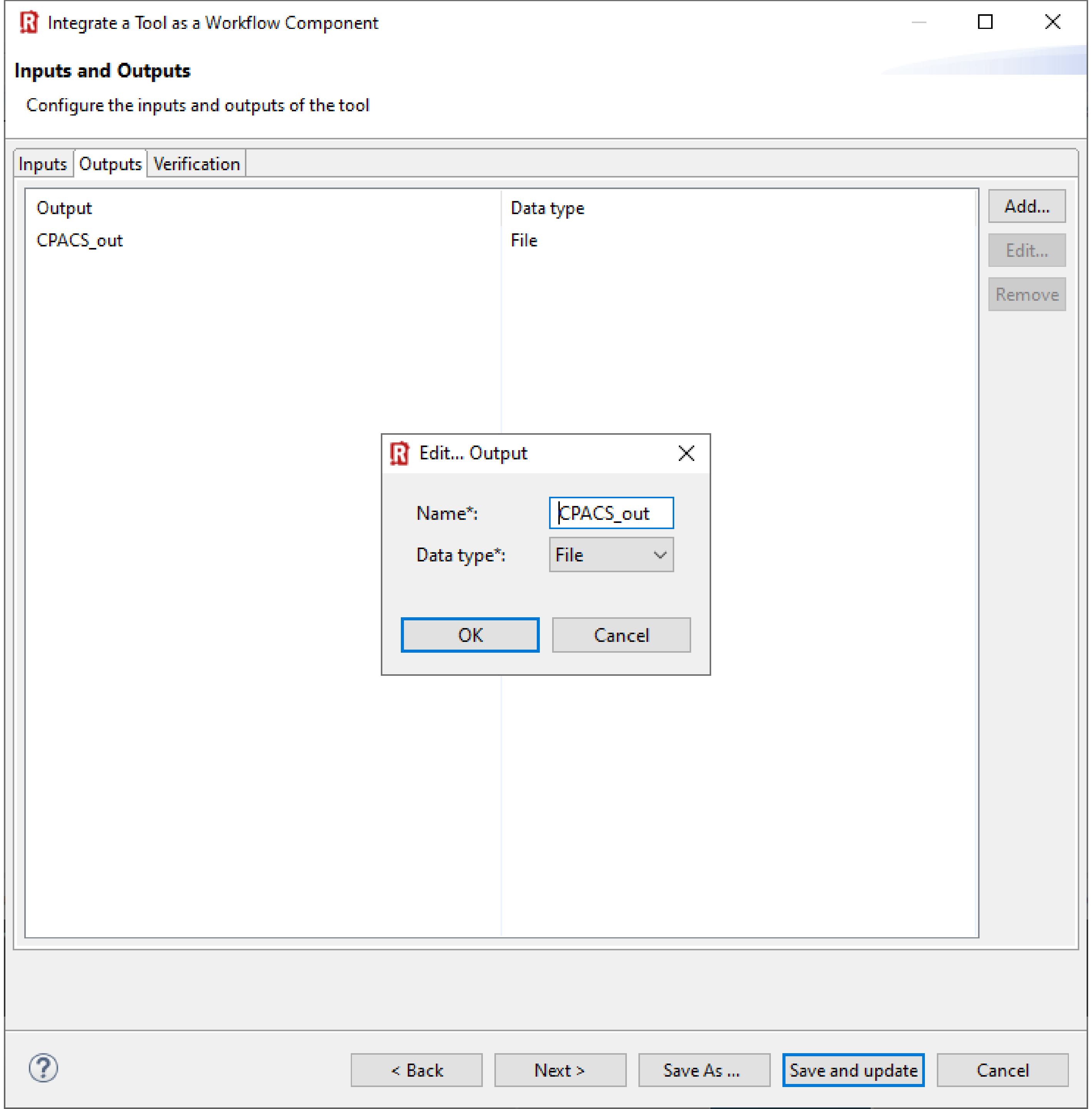
Data type*: File

Handling*:
☐ Constant (not consumed)
☑ Single (consumed)
☑ Queue (consumed)

Default handling*: Queue (consumed)

Constraint*:
☑ Required
☐ Required if connected
☐ Not required

Default constraint*: Required



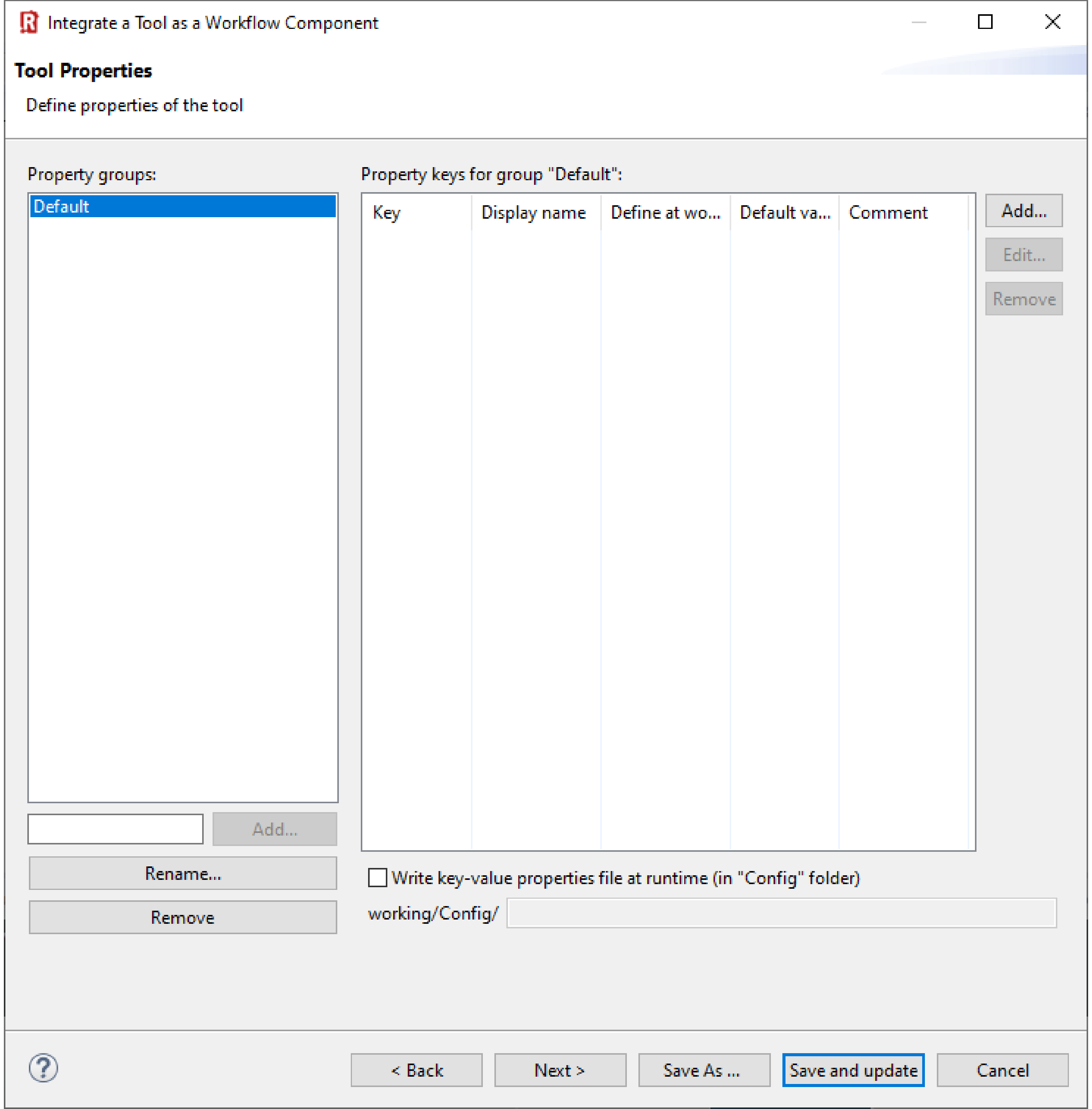
Inputs and Outputs

Configure the inputs and outputs of the tool

Output	Data type
CPACS_out	File

Name*: CPACS_out

Data type*: File



Integrate a Tool as a Workflow Component

Tool Properties

Define properties of the tool

Property groups:

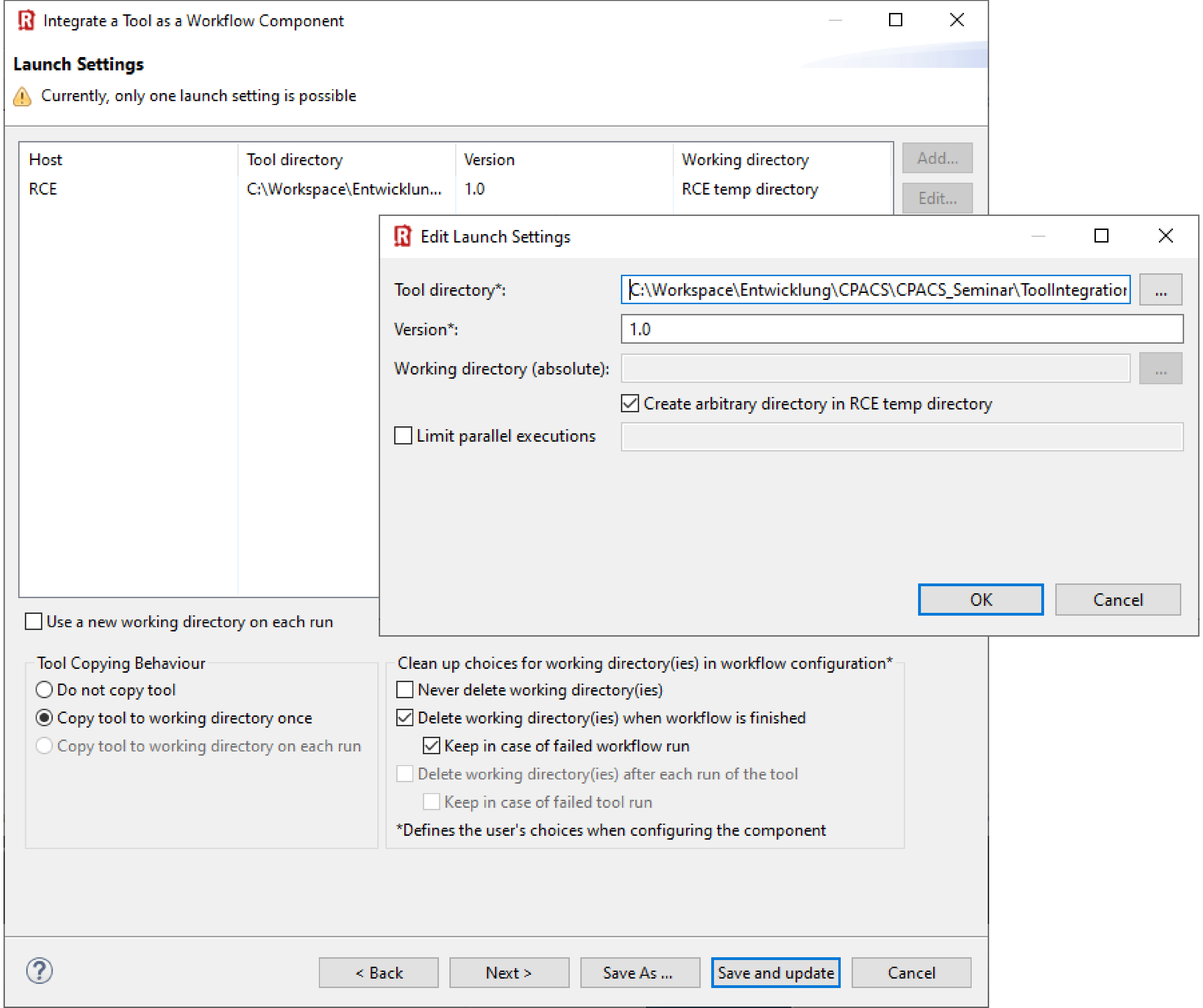
Default

Property keys for group "Default":

Key	Display name	Define at wo...	Default va...	Comment

☐ Write key-value properties file at runtime (in "Config" folder)

working/Config/



Launch Settings

Currently, only one launch setting is possible

Host	Tool directory	Version	Working directory
RCE	C:\Workspace\Entwicklun...	1.0	RCE temp directory

Tool directory*: C:\Workspace\Entwicklung\CPACS\CPACS_Seminar\ToolIntegratior

Version*: 1.0

Working directory (absolute):

☑ Create arbitrary directory in RCE temp directory

□ Limit parallel executions

□ Use a new working directory on each run

Tool Copying Behaviour

○ Do not copy tool

● Copy tool to working directory once

○ Copy tool to working directory on each run

Clean up choices for working directory(ies) in workflow configuration*

□ Never delete working directory(ies)

☑ Delete working directory(ies) when workflow is finished

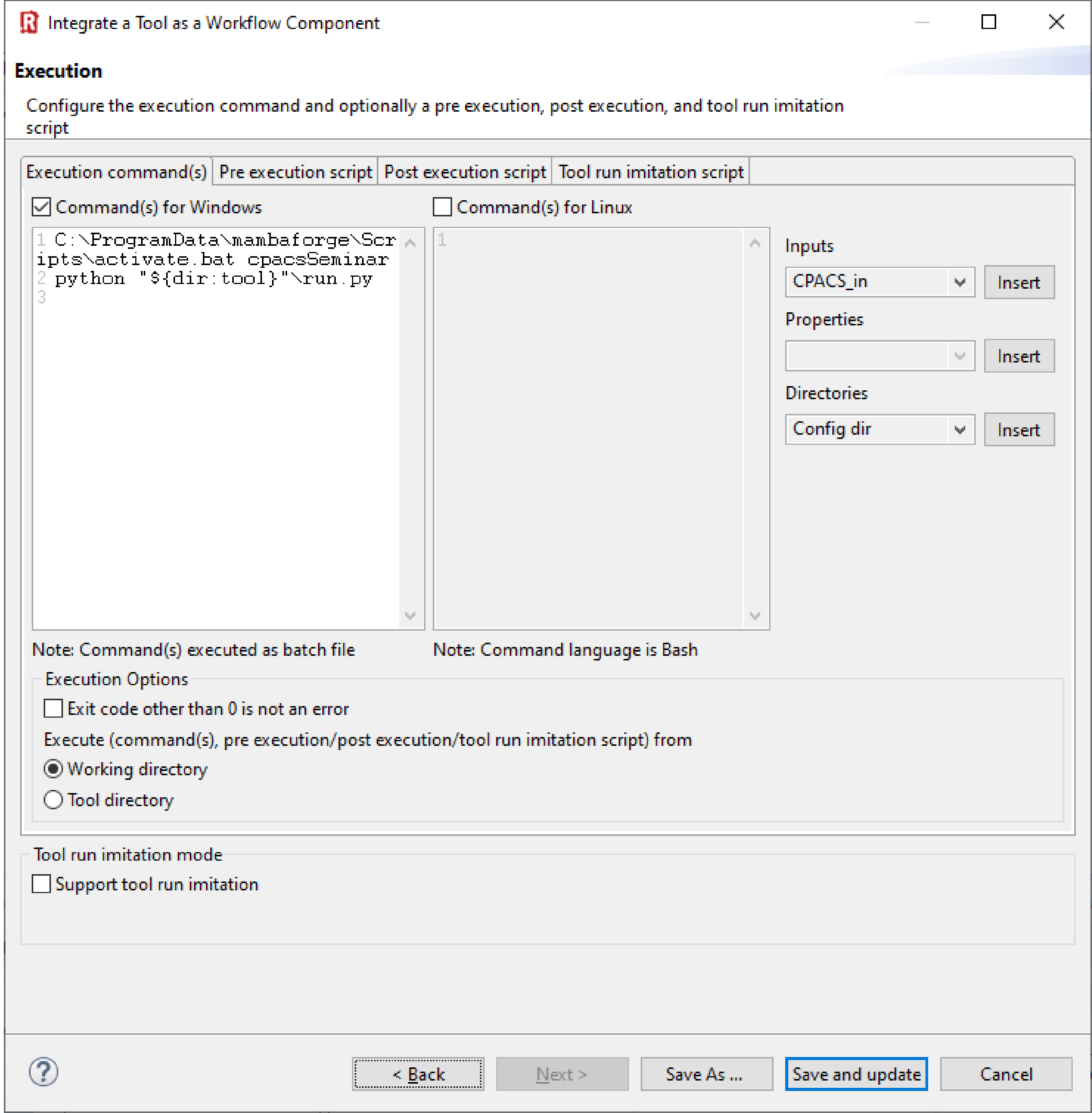
☑ Keep in case of failed workflow run

□ Delete working directory(ies) after each run of the tool

□ Keep in case of failed tool run

*Defines the user's choices when configuring the component

```
C:\ProgramData\mambaforge\Scripts\
activate.bat cpacsSeminar
python "${dir:tool}"\run.py
```



Integrate a Tool as a Workflow Component

Execution

Configure the execution command and optionally a pre execution, post execution, and tool run imitation script

☑ Command(s) for Windows

```
C:\ProgramData\mambaforge\Scripts\activate.bat cpacsSeminar
python "${dir:tool}"\run.py
```

□ Command(s) for Linux

Inputs: CPACS_in

Properties:

Directories: Config dir

Note: Command(s) executed as batch file

Note: Command language is Bash

Execution Options

□ Exit code other than 0 is not an error

Execute (command(s), pre execution/post execution/tool run imitation script) from

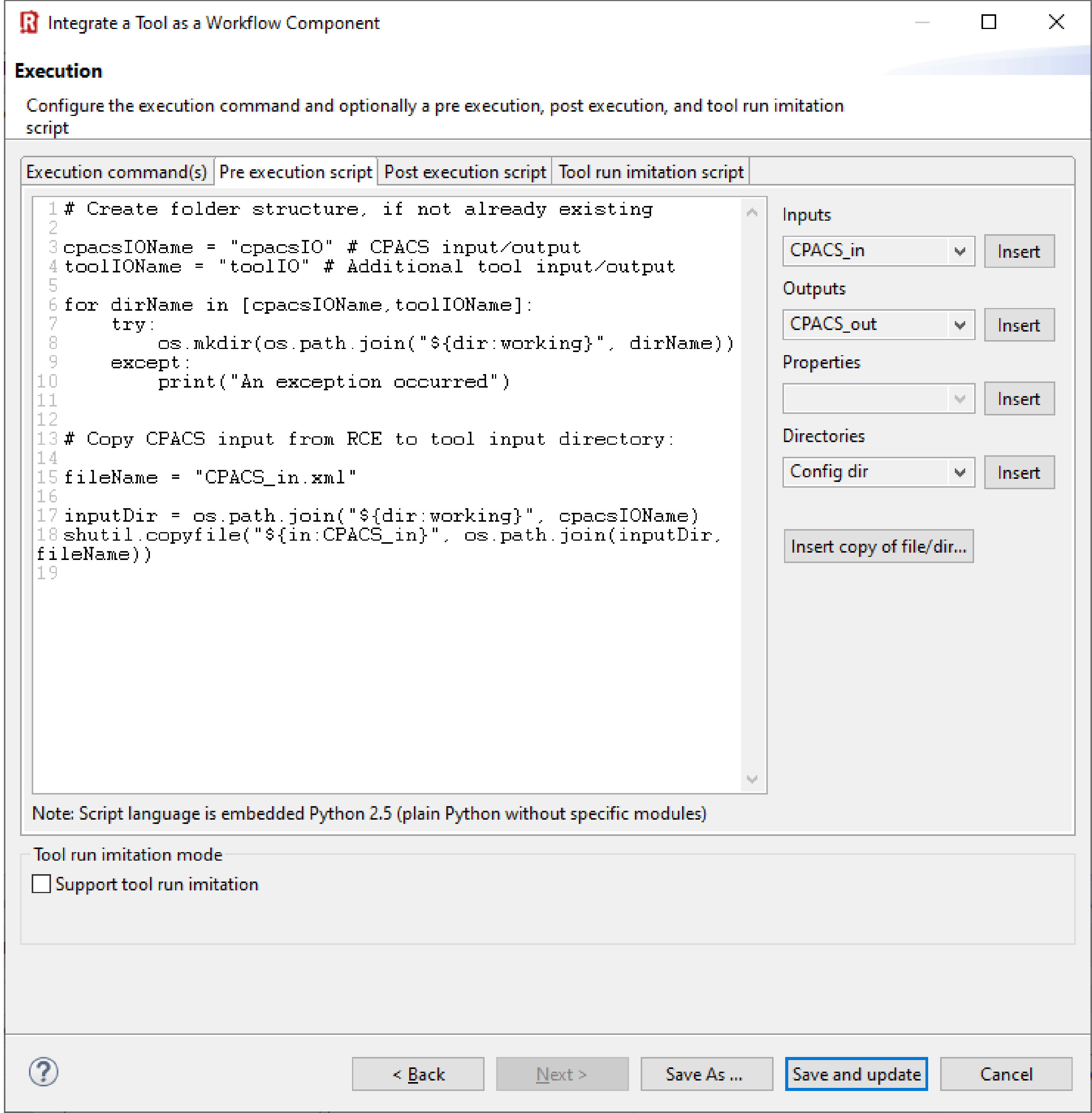
● Working directory

○ Tool directory

Tool run imitation mode

□ Support tool run imitation

```
# Create folder structure, if not already existing

cpacsIOName = "cpacsIO" # CPACS input/output
toolIOName = "toolIO" # Additional tool input/output

for dirName in [cpacsIOName,toolIOName]:
    try:
        os.mkdir(os.path.join("${dir:working}", dirName))
    except:
        print("An exception occurred")


# Copy CPACS input from RCE to tool input directory:

fileName = "CPACS_in.xml"

inputDir = os.path.join("${dir:working}", cpacsIOName)
shutil.copyfile("${in:CPACS_in}", os.path.join(inputDir,
fileName))
```



Integrate a Tool as a Workflow Component

Execution

Configure the execution command and optionally a pre execution, post execution, and tool run imitation script

```
# Create folder structure, if not already existing

cpacsIOName = "cpacsIO" # CPACS input/output
toolIOName = "toolIO" # Additional tool input/output

for dirName in [cpacsIOName,toolIOName]:
    try:
        os.mkdir(os.path.join("${dir:working}", dirName))
    except:
        print("An exception occurred")


# Copy CPACS input from RCE to tool input directory:

fileName = "CPACS_in.xml"

inputDir = os.path.join("${dir:working}", cpacsIOName)
shutil.copyfile("${in:CPACS_in}", os.path.join(inputDir,
fileName))
```

Inputs: CPACS_in

Outputs: CPACS_out

Properties:

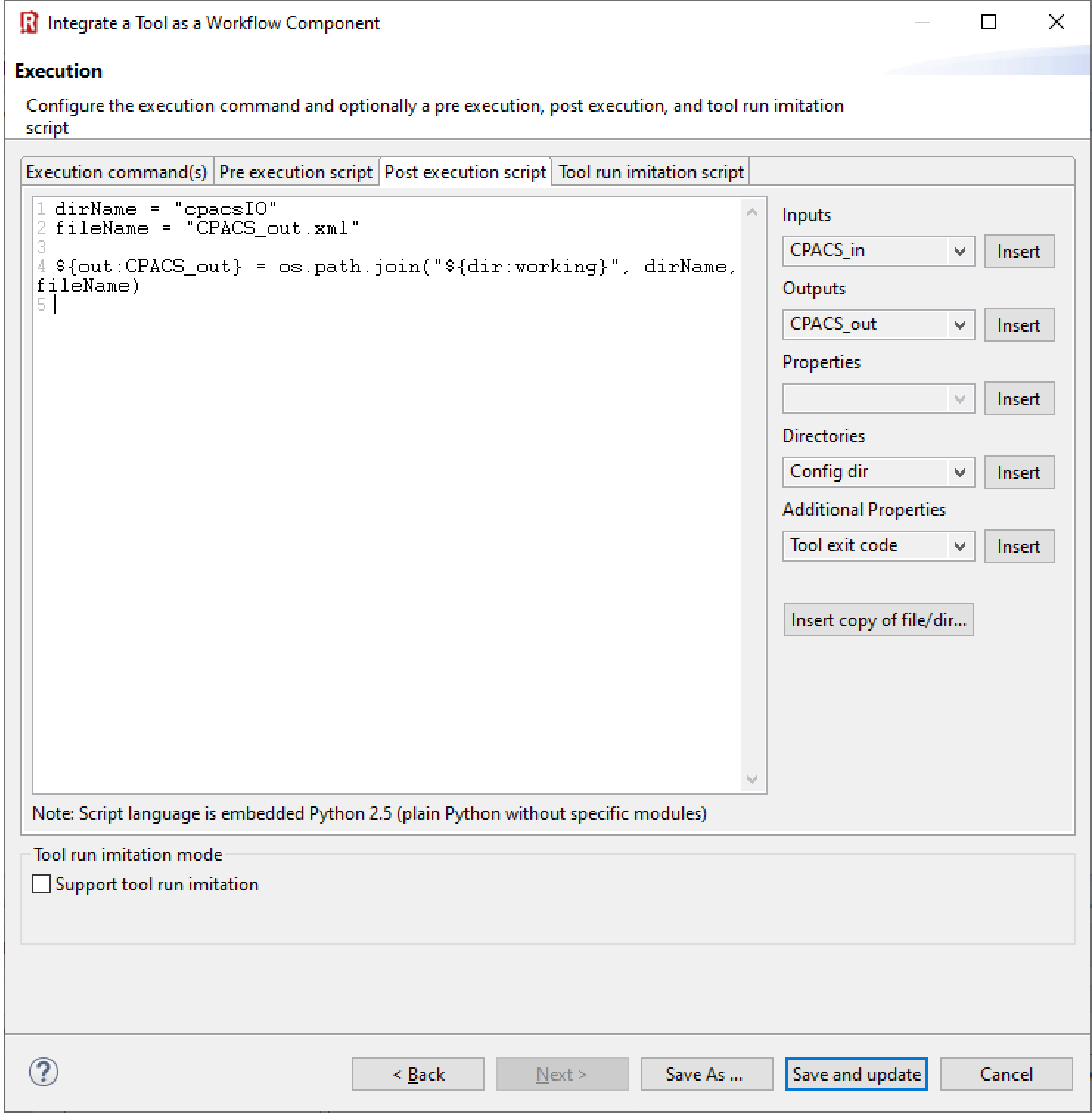
Directories: Config dir

Note: Script language is embedded Python 2.5 (plain Python without specific modules)

Tool run imitation mode

☐ Support tool run imitation

```
dirName = "cpacsIO"
fileName = "CPACS_out.xml"

${out:CPACS_out} = os.path.join("${dir:working}",
dirName, fileName)
```



Integrate a Tool as a Workflow Component

Execution

Configure the execution command and optionally a pre execution, post execution, and tool run imitation script

```
dirName = "cpacsIO"
fileName = "CPACS_out.xml"

${out:CPACS_out} = os.path.join("${dir:working}", dirName,
fileName)

```

Inputs: CPACS_in

Outputs: CPACS_out

Properties:

Directories: Config dir

Additional Properties: Tool exit code

Note: Script language is embedded Python 2.5 (plain Python without specific modules)

Tool run imitation mode

□ Support tool run imitation