Adding a new C++ program

Frank PLCnext Team

A PLCnext Technology C++ application may consist of one or more components and each component of one or more programs.

This article explains how to add an additional C++ program in an existing Eclipse® application.

This document is based on the release firmware of the AXC F 2152.

The knowledge of the following articles is assumed for the information given here.

- How to create a PLCnext Technology C++ application - Tutorial Video - Your first steps with Eclipse and C++
- General structure of a PLCnext Technology application - Structure of a PLCnext application

Eclipse® allows the simple adding of additional C++ programs (.cpp and .hpp) as empty files. Note that the namespaces and includes must be added manually when adding a program this way.

An even more simple way to add a new C++ program for PLCnext Technology, is to copy an existing (empty) program and to paste it into the existing application. In the example shown below, the empty "MyFirstProgram.cpp" and "MyFirstProgram.hpp" are copied, inserted, and renamed to "MySecondProgram.cpp" and "MySecondProgram.hpp".

Note: Observe the PLCnext Technology naming conventions.

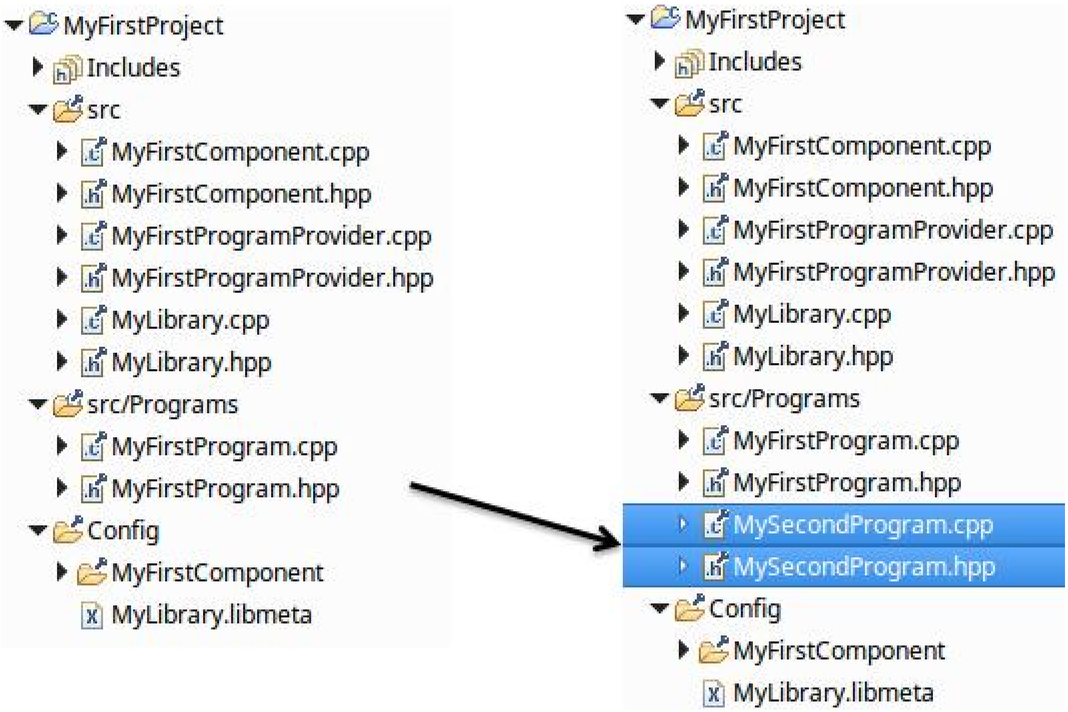
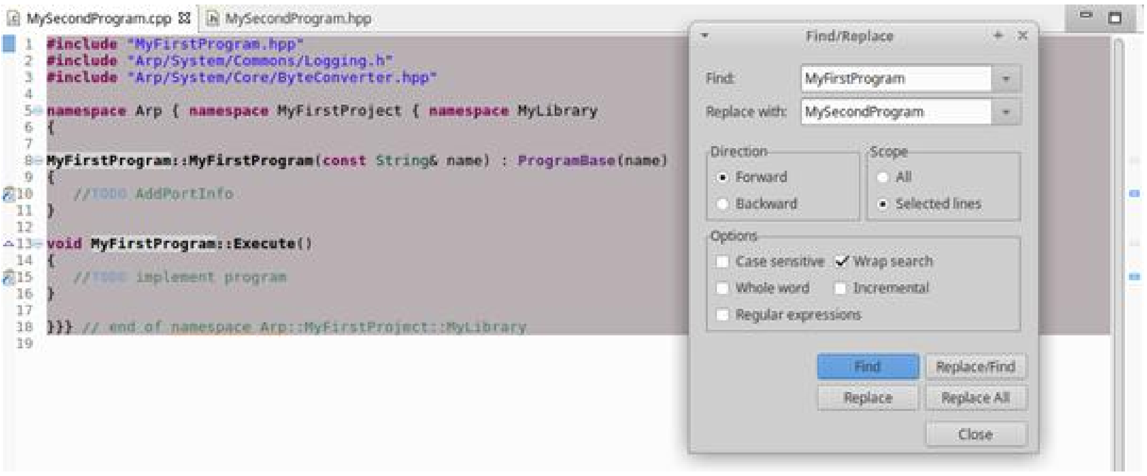


Figure 1: Adding an additional Program

In the "MySecondProgram" files, still the old references valid for "MyFirstProgram" are specified. Use the search & replace function of Eclipse® to modify these references to "MySecondProgram".



```
#include "MyFirstProgram.hpp"
#include "Arp/System/Commons/Logging.h"
#include "Arp/System/Core/ByteConverter.hpp"

namespace Arp { namespace MyFirstProject { namespace MyLibrary
{

MyFirstProgram::MyFirstProgram(const String& name) : ProgramBase(name)
{
    //TODO AddPortInfo
}

void MyFirstProgram::Execute()
{
    //TODO implement program
}

}}} // end of namespace Arp::MyFirstProject::MyLibrary
```

Find: MyFirstProgram

Replace with: MySecondProgram

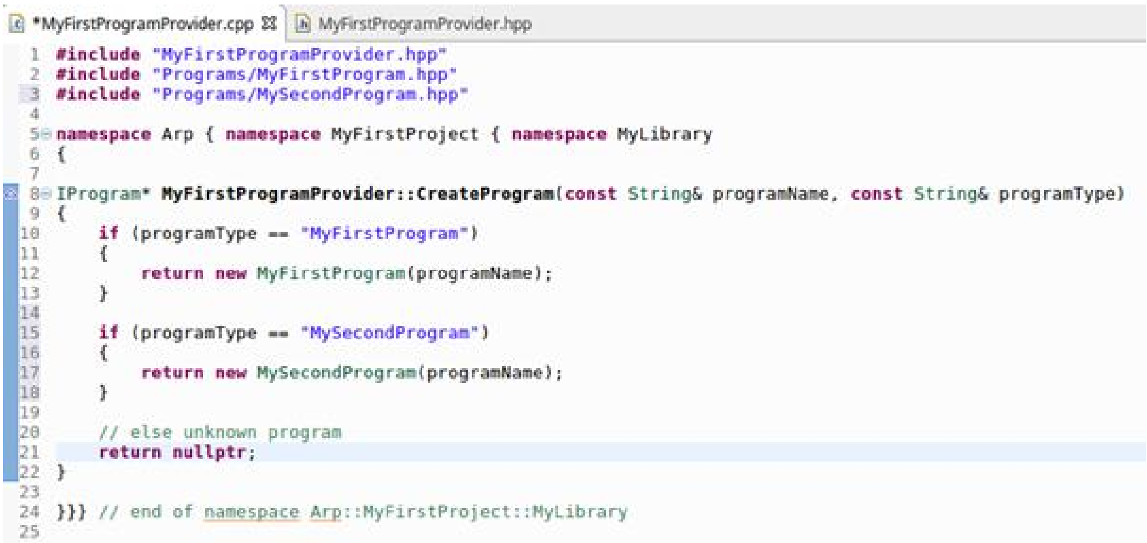
Direction: ● Forward ○ Backward

Scope: ○ All ● Selected lines

Options: □ Case sensitive ☑ Wrap search □ Whole word □ Incremental □ Regular expressions

Figure 2: Adapting name references

The body of "MySecondProgram" is now completed, but the new program has to be declared in the ProgramProvider (in this case "MyFirstProgramProvider.cpp").



```
#include "MyFirstProgramProvider.hpp"
#include "Programs/MyFirstProgram.hpp"
#include "Programs/MySecondProgram.hpp"

namespace Arp { namespace MyFirstProject { namespace MyLibrary
{

IProgram* MyFirstProgramProvider::CreateProgram(const String& programName, const String& programType)
{
    if (programType == "MyFirstProgram")
    {
        return new MyFirstProgram(programName);
    }

    if (programType == "MySecondProgram")
    {
        return new MySecondProgram(programName);
    }

    // else unknown program
    return nullptr;
}

}}} // end of namespace Arp::MyFirstProject::MyLibrary

```

Figure 3: Adding the new program to the ProgramProvider

Prior to creating an executable application for PLCnext Technology, the meta files have to be updated.
This can be done using the PLCnext Technology CDT AddIn.

Figure 4: Adding a new GDS Port and update the meta files

Copy the created "MySecondProgramm.progmeta" into the "MySecondProgram" folder.

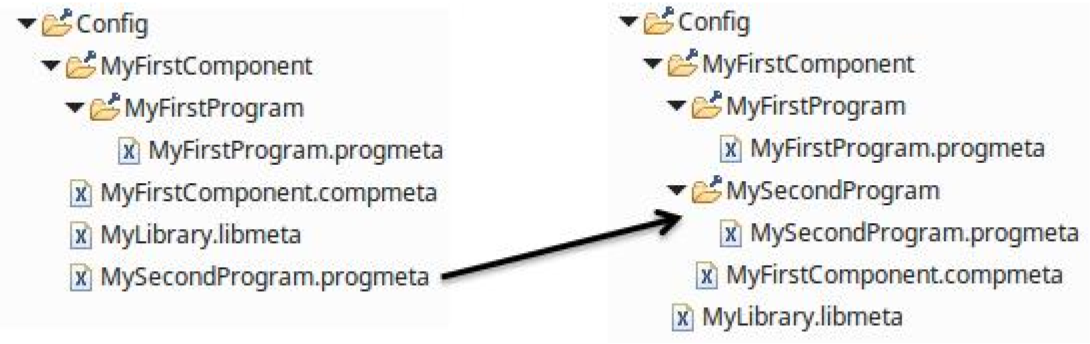
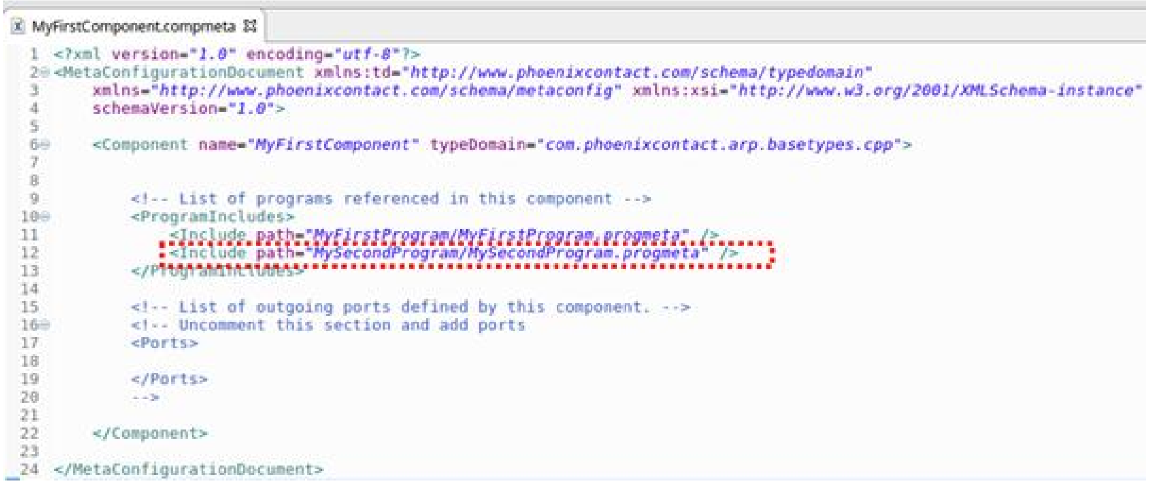


Figure 5: Copying the updated meta file

Finally, the component meta file "MyFirstComponent.compmeta" must be updated with the reference to "MySecondProgramm.progmeta".



```
<?xml version="1.0" encoding="utf-8"?>
<MetaConfigurationDocument xmlns:td="http://www.phoenixcontact.com/schema/typedomain"
    xmlns="http://www.phoenixcontact.com/schema/metaconfig" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    schemaVersion="1.0">

    <Component name="MyFirstComponent" typeDomain="com.phoenixcontact.arp.basetypes.cpp">


        <!-- List of programs referenced in this component -->
        <ProgramIncludes>
            <Include path="MyFirstProgram/MyFirstProgram.progmeta" />
            <Include path="MySecondProgram/MySecondProgram.progmeta" />
        </ProgramIncludes>

        <!-- List of outgoing ports defined by this component. -->
        <!-- Uncomment this section and add ports
        <Ports>

        </Ports>
        -->

    </Component>

</MetaConfigurationDocument>
```

*Figure 6: Including *.progmeta into *.compmeta*

The PLCnext Technology application can now be compiled and the PC Worx Engineer library created as shown in [Tutorial Video - Your first steps with Eclipse and C++](#).

After having inserted the generated library in the PC Worx Engineer solution, "MySecondProgram" is listed in the 'References' category (COMPONENTS area) and can be used.

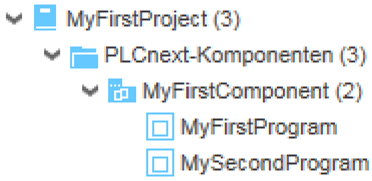


Figure 7: PC Worx Engineer Library with MySecondProgram

The PCWorx Engineer online help provides detailed information how to include such external programs in the 'Task and Events' editor of the PLCnext PLANT tree node thus creating a program instance.