TMcraft

Node Tutorial

Basic Development

Original Instructions

Document version: 1.00

Release date: 2024-02-07

This Manual contains information of the Techman Robot product series (hereinafter referred to as the TM AI Cobot).The information contained herein is the property of Techman Robot Inc. (hereinafter referred to as the Corporation). No part of this publication may be reproduced or copied in any way, shape or form without prior authorization from the Corporation. No information contained herein shall be considered an offer or commitment. It may be subject to change without notice. This Manual will be reviewed periodically. The Corporation will not be liable for any error or omission.

and logo are registered trademark of TECHMAN ROBOT INC. in Taiwan and other countries and the company reserves the ownership of this manual and its copy and its copyrights.

TECHMAN ROBOT INC.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

Table of Contents

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

Revision History

Revision	Date	Description
1.00	2024-02-07	Original release

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

1. Introduction

The TMcraft Node, designed with C#/WPF, is tailored for creating applications in flow projects. Similar to the TM Component, it is suitable for use in these scenarios:

(1) As a Device Node to set up and manipulate the 3rd party device during the project run.

(2) As an Application Node to set up and execute processes such as palletizing, machine tending, etc. during the project operation. Here is an example of a TMcraft Node customized for welding:

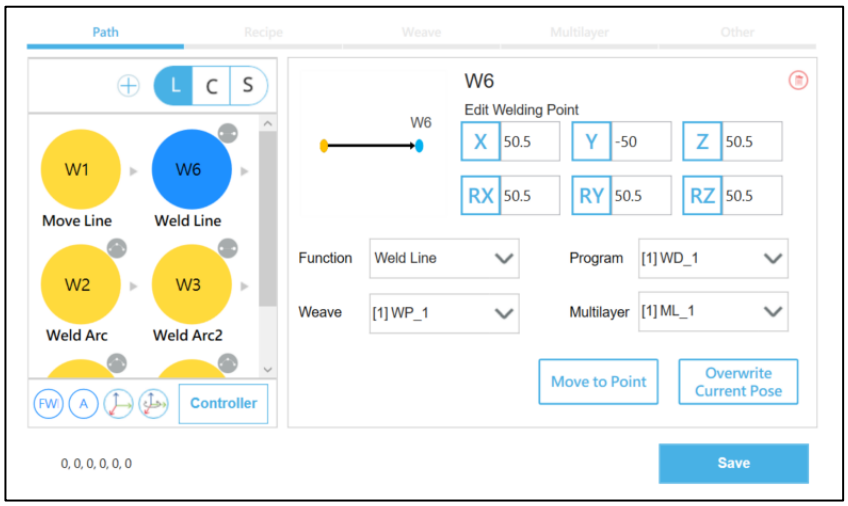


Figure 1: The TM Weld Node based on a TMcraft Node

Users can add the TMcraft Node onto the flow and do the settings through its customized Node (edit) UI; when the Node UI is about to close, the TMcraft Node Program can collect the settings, generate the corresponding TMscript and write it onto the project. During the project operation, the flow reaches that TMcraft Node and the robot will execute that TMscript.

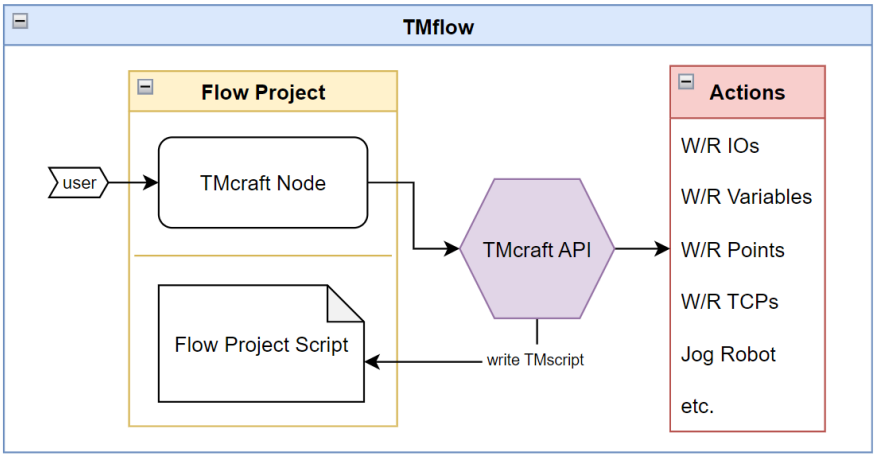


Figure 2: The Architecture of a TMcraft Node

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

To determine whether the TMcraft Node is the right fit to realize concepts of users, consider these questions:

Are users familiar with script programming?

As built with C#, developers need C# skills to develop the TMcraft Node.

Is a fancy GUI necessary?

One of the distinctions between TMcraft Node and TM Component lies in the capability for users to create a customized Node UI using C#/WPF, offering a unique user experience to customers.

Is the goal to provide flexibility in application definition for customers?

With the TMcraft Node, one can create a wizard for some processes, but this may not cover the entire application since the completion of the flow project rests with the users. This approach offers benefits in terms of flexibility. However, for those aiming to provide a complete, customized user experience, developers are advised to create the TMcraft Shell.

To understand how to develop a TMcraft Node, please check the following diagram:

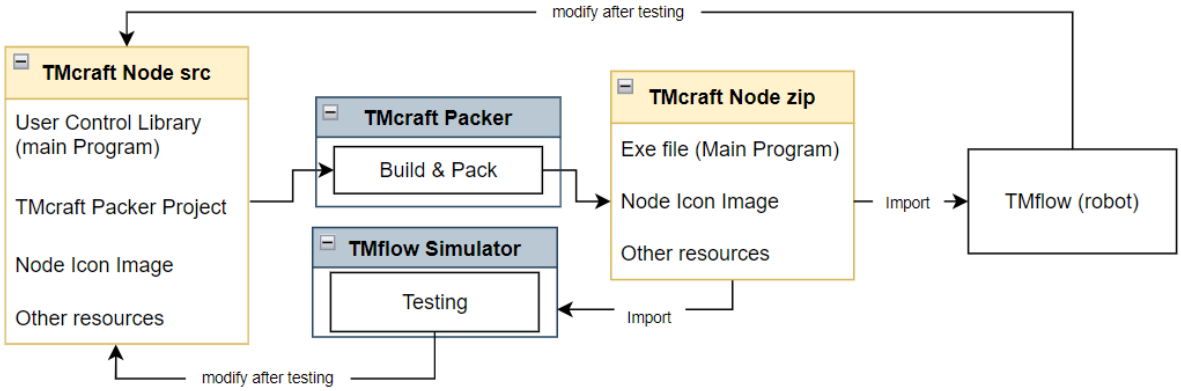


Figure 3: Basic Concept of TMcraft Node development

Developers create the TMcraft Node UI as a User Control Library and use TMcraft Packer to build and package it into a zip file. They can then test it on tools like TMflow Simulator, TMstudio Pro, or directly on a robot. Following multiple iterations of modifications and testing, they can release the TMcraft Node.

This tutorial organizes the content based on TMcraft.dll version 1.16.1400 from TMflow 2.16.

- Section 2: prerequisites for developing a TMcraft Node
- Section 3: the concept and features of the TMcraft API and TMcraft Node Program
- Section 4: the process of how to develop a simple TMcraft Node
- Section 5: how to complete building a TMcraft Node Package
- Section 6: the concept of Node Unique Identifier
- Section 7: how to import and use TMcraft Node on TMflow
- Section 8: how to import and use TMcraft Node on TMstudio Pro
- Section 9: the dos and don'ts

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

2. Prerequisites

To develop TMcraft Nodes, developers should be capable of:

- Having Basic knowledge of programming with C#
- Using WPF to build UI. Understanding of the User Control concept is preferred.
- Using TM script the Programming Language

Software requirements,

- An integrated development environment for C# and WPF (such as Microsoft Visual Studio 2022)
- TMflow 2.16 or above
- .NET 6.0 and .NET SDK
- TMcraft Packer for building and packaging the TMcraft item

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

3. TMcraft Node Program Structure

3.1 Brief Explanation of TMcraft API

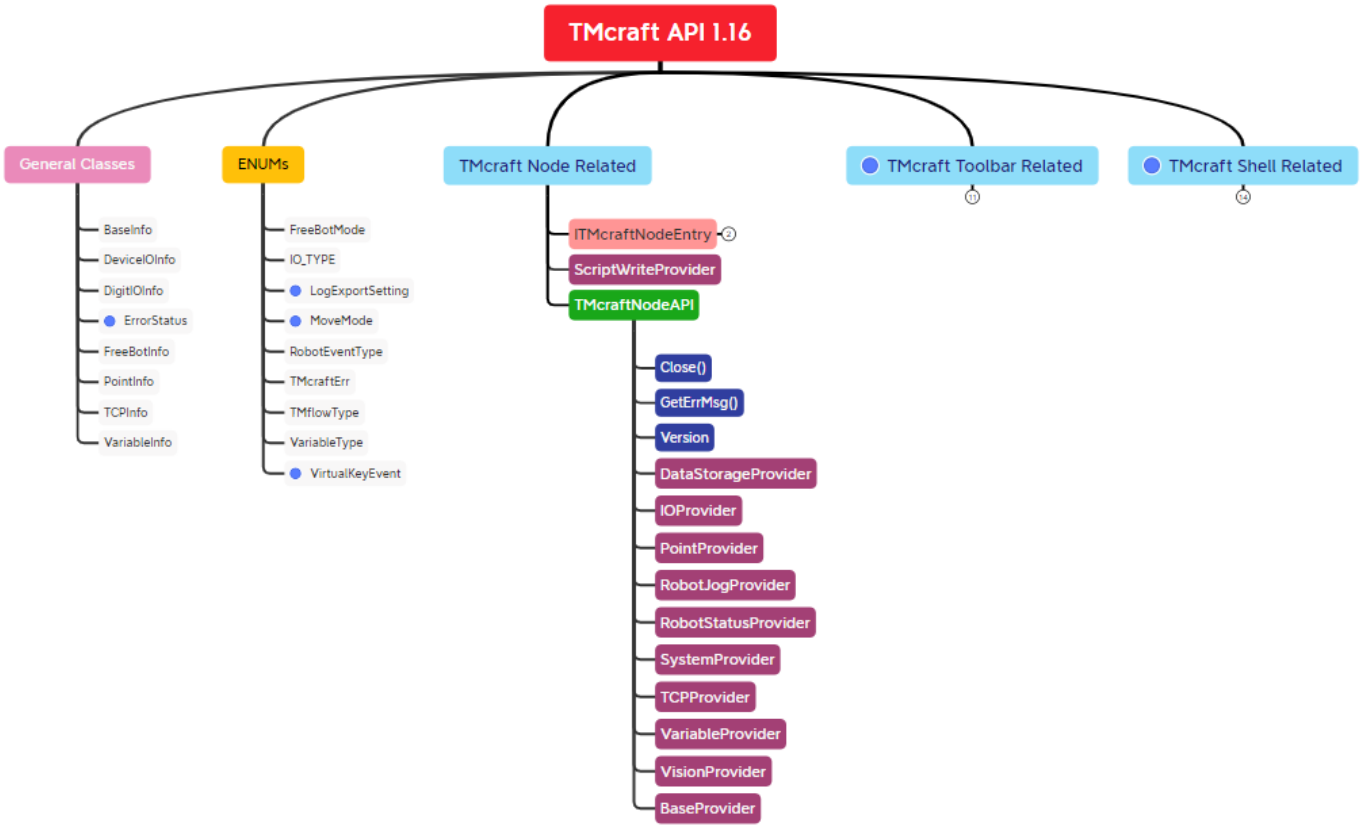


Figure 4: TMcraft API Architecture

TMcraft API is an interface that connects the TMcraft Node with TMflow to get all sorts of robot information and request TMflow to take different actions. In addition, this API is a Dynamic Link Library file (DLL); developers should add it to program dependencies so that the program can use the functions within the API.

TMcraft.dll consists of classes, enums, and functions for every TMcraft items. Here are those most related to TMcraft Node:

- ITMcraftNodeEntry is an interface that defines the program as a TMcraft Node. Users must define the member functions, InitializeNode() and InscribeScript() to make it work.
- TMcraftNodeAPI has all the functions and provider classes to build TMcraft Node. Note that each TMcraft item comes with its own set of provider classes. Even if some share the same name, their member functions vary.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

- Providers are a set of classes with functions associated with different interactions with TMflow frequently used among the Programs.

- General classes are a collection of classes, such as DeviceIOInfo, TCPInfo, and ErrorStatus, used by different types of Provider functions.
- Enum, or enumerators, are value types defined by a set of named constants of the underlying integral numeric type used by provider functions.

Refer the table below to overview the API capabilities across various TMcraft plugins:

Plugins / capabilities	Node	Shell	Toolbar
Base (Add\Edit\Delete)	O		
Point (Add\Edit\Delete)	O		
Tool (Add\Edit\Delete)	O	O	O
Digital IO (Read/Write)	O	O	O
Analog IO (Read/Write)	O	O	O
Variables (New/Edit)	O	O	O
Vision Job (Add\Open\Delete)	O		
Jog the robot	O	O	
Freebot (Set/Get)	O	O	O
End Button Event	O	O	O
Get Current Language	O	O	O
Get TMflow Type	O	O	O
TMscript on flow project (Read/Write)	O		
Login/Logout/Get Control		O	
script Project (Add\Edit\Delete)		O	
Robot status (Error, Run, etc.)		O	O
Error Event		O	
Virtual Robot Stick		O	
Export/Import		O	
Variables Runtime Value (Read/Write)		O	Read only

3.2 Program structure of a TMcraft Node

Refer to the sample code below for the program structure.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

```
using TMcraft;

namespace TMcraftSample
{
    public partial class MainPage : UserControl, ITMcraftNodeEntry
    {
        TMcraftNodeAPI NodeUI;
        string _TMscript = string.Empty;
        bool fgSave = false;

        public MainPage()
        {
            InitializeComponent();
        }

        public void InitializeNode(TMcraftNodeAPI _nodeUI)
        {
            NodeUI = _nodeUI; //connect TMflow
        }

        public void InscribeScript(ScriptWriteProvider scriptWriter)
        {
            if(fgSave)
            {
                //_TMscript could be modified elsewhere
                scriptWriter.AppendScript(_TMscript);
            }
        }
}
```

To create the foundation of a TMcraft Node program, please remember the key points below:

1. Include TMcraft.dll as a reference. Remember to add the namespace, using TMcraft, onto the program.
2. Declare a global object based on the class TMcraftNodeAPI.
3. Implement the interface ITMcraftNodeEntry and define its member functions InitializeNode and InscribeScript.

The rest of the program should be all sorts of event functions that can interact with TMflow through TMcraft functions.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

4. Start Programming a TMcraft Node

Developers are favored to follow the steps to develop a TMcraft Node.

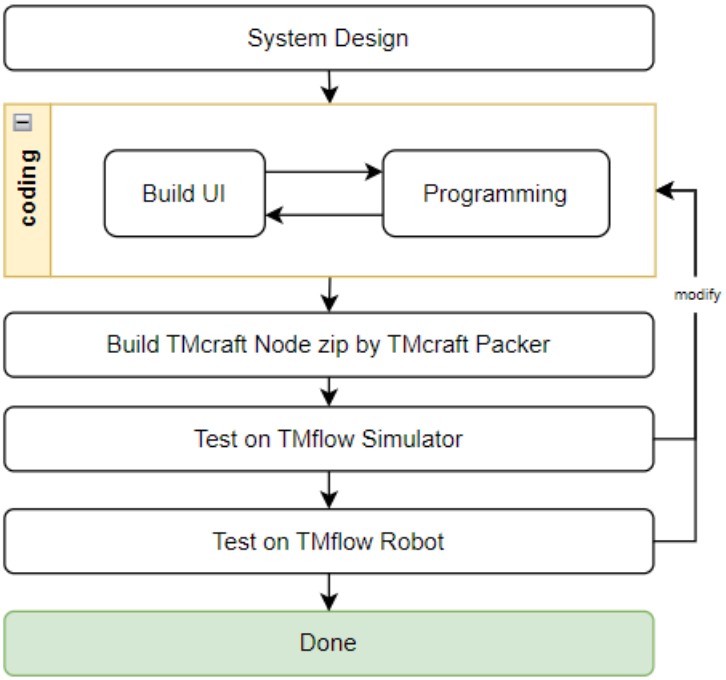


Figure 5: The Recommended Procedure for Developing a TMcraft Node

From section 4 to section 6, users will learn how to approach these steps with a simple TMcraft Node, *HelloWorldNode*.

4.1 System Design

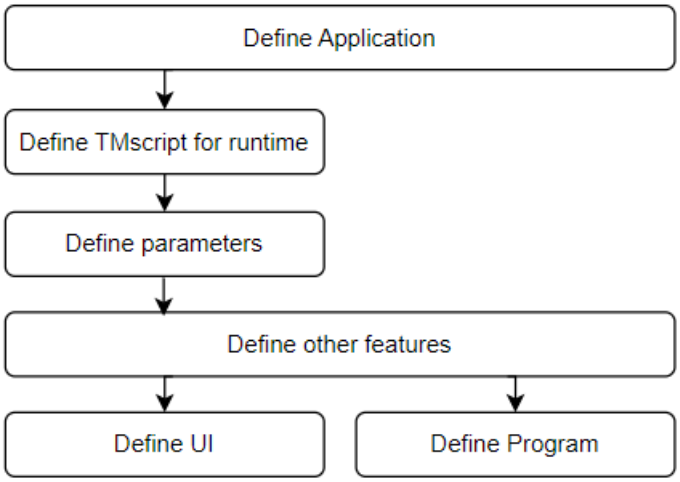


Figure 6: The Recommended Procedure of System Design

The outline of the requirements:

- A simplified version of Display Node that users can type a message on the textbox to display when the project run reaches this node.
- Save the message above as a local variable.
- “Hello, World” will display as a banner when the project operation reaches this node.

With such a definition, the relevant TMscript should go like this:

Display(“Green” ,” White” ,” Hello World” , [Content])

Therefore, [Content] is the user-decide parameter. Please refer to the figures below for the UI and the program architecture.

Figure 7: HelloWorldNode UI

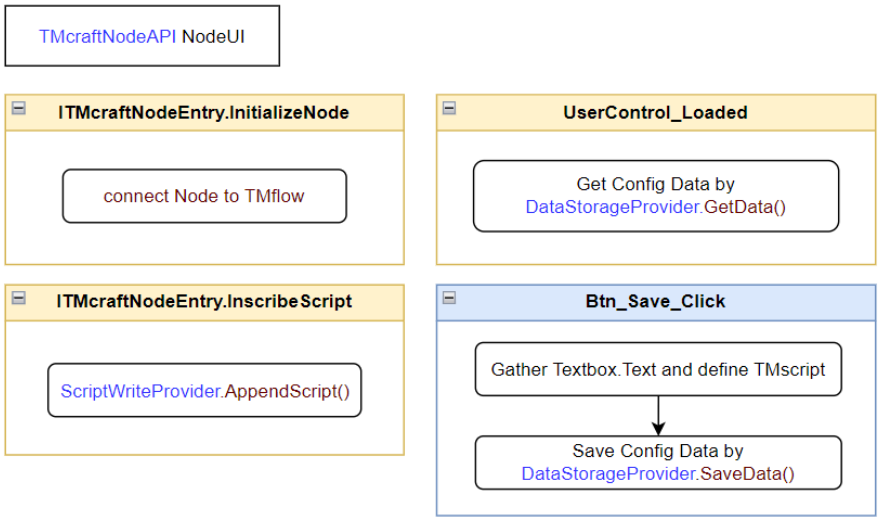


Figure 8: Program architecture of HelloWorldNode

4.2 Create a WPF Application Project

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

This section is written based on the usage of Microsoft Visual Studio 2022. First, users can create a new project with the *WPF User Control Library* template.

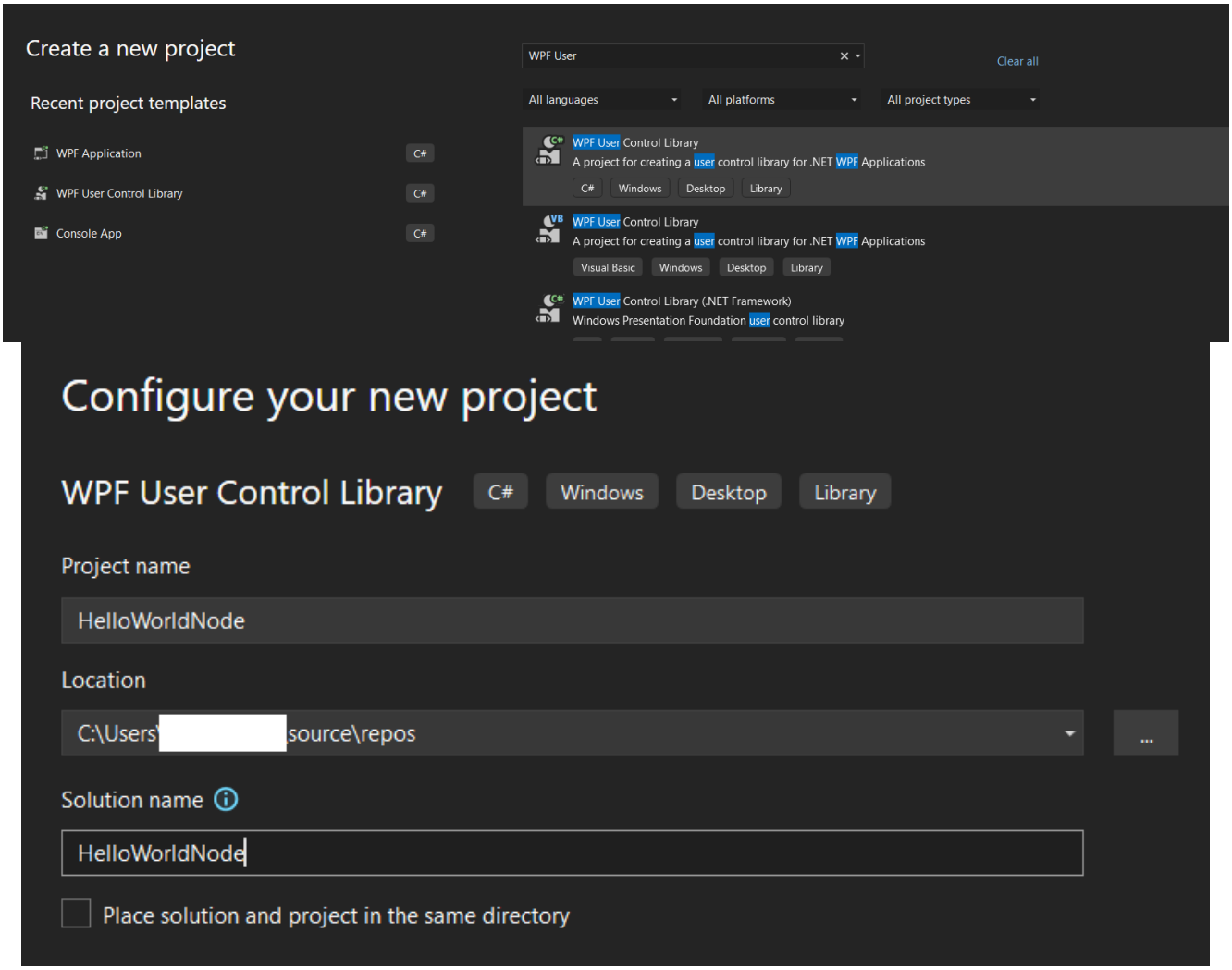
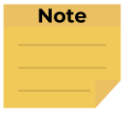


Figure 9: Create a New Project with the WPF User Control Library Template



NOTE:
Be reminded that it is important to use different names for the UserControl DLL and the node exe name to be defined later on TMcraft Packer (see Section 5).

Remember to set up with .NET 6.0 as the framework of the project.

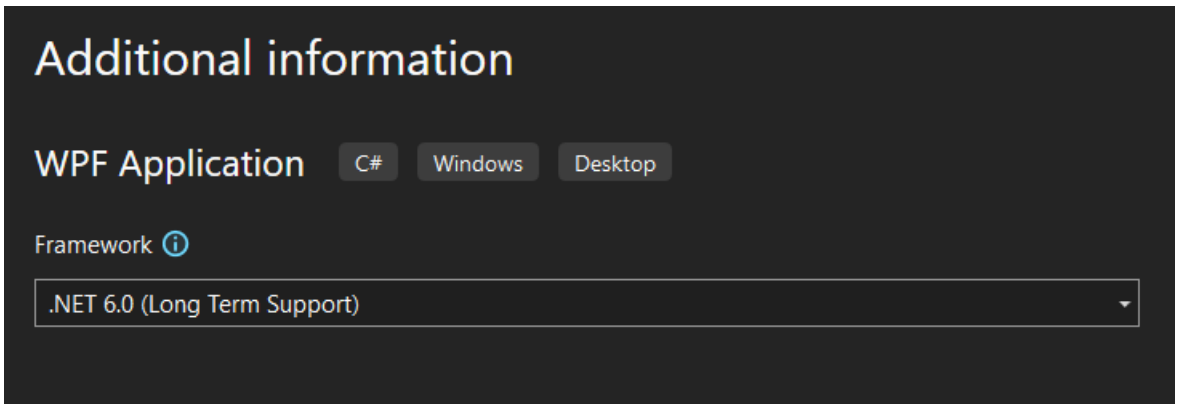


Figure 10: .NET 6.0 Setup as the Framework

Once opened the project in the editor, add the TMcraft.dll as the reference.

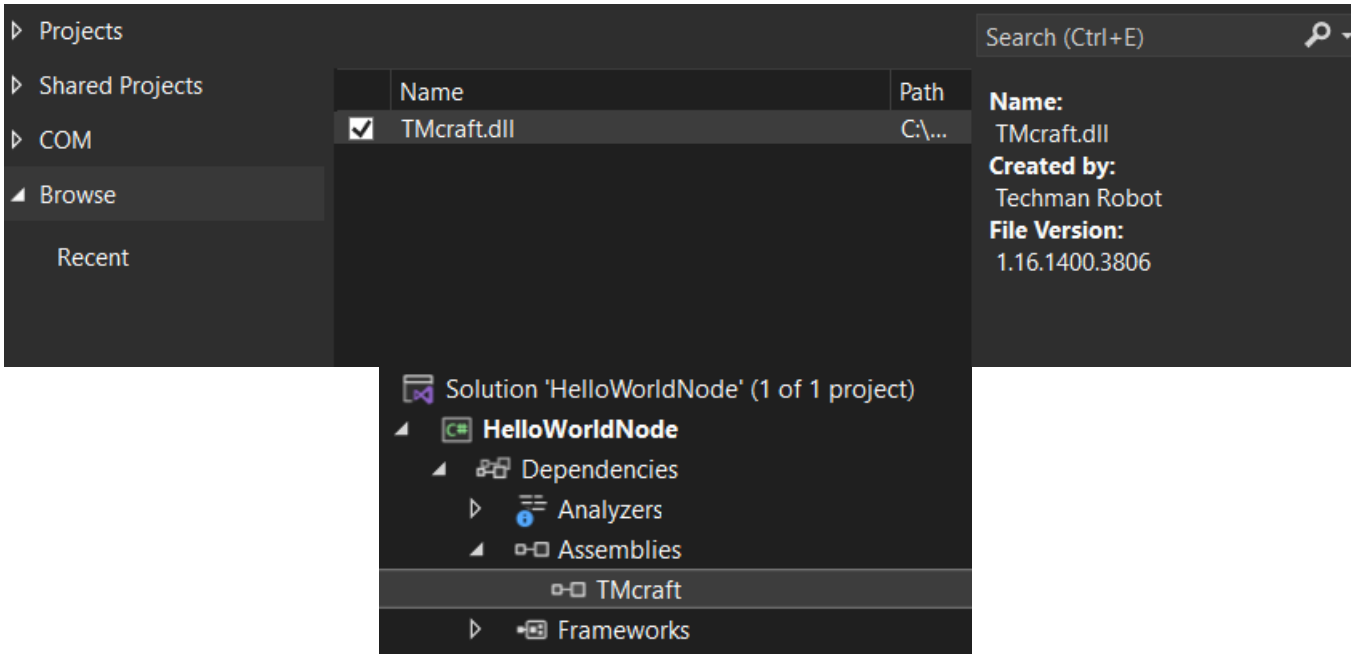


Figure 11: Add TMcraft.dll as a Reference

4.3 Source Code

This section focuses on the key aspects of the HelloWorldNode source code. For the complete source code, please refer to TMcraft Development Kit\Node\Samples\ HelloWorldNode\

4.3.1 MainPage.xaml

MainPage.xaml defines the UI. Here are a few points necessary:

- x:Class="HelloWorldNode.MainPage", used as one of the required parameters to build the executable file of the TMcraft Node. Refer to section 5 for more details.
- Loaded="UserControl_Loaded": defines the actions taken during UI initialization.
- Button click event Btn_Save_Click.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

4.3.2 MainPage.xaml.cs

MainPage.xaml.cs defines the program of the TMcraft Node; here are a few points necessary:

- At first, add TMcarft.dll onto the program reference and the interface ITMcraft-NodeEntry.

```
using TMcraft;

namespace HelloWorldNode
{
    /// <summary>
    /// Interaction logic for MainPage.xaml
    /// </summary>
    4 references
    public partial class MainPage : UserControl,ITMcraftNodeEntry
    {
```

- Declare a global TMcraftNodeAPI object NodeUI and global variables with string _TMscript that represents the status of the target digital output.

```
public partial class MainPage : UserControl,ITMcraftNodeEntry
{
    TMcraftNodeAPI NodeUI;
    string _TMscript = string.Empty;
    bool fgSave = false;
```

- Implement the first ITMcraftNodeEntry member function, InitializeNode(); this function executes once the Node Program starts and connects to TMflow.

```
0 references
public void InitializeNode(TMcraftNodeAPI _nodeUI)
{
    NodeUI = _nodeUI; //connect TMflow
}
```

- Implement the second member ITMcraftNodeEntry function, InscribeScript (); this function executes once the Node Program is about to close. If fgSave is true, write the _TMscript onto the flow project by calling ScriptWriteProvider.AppendScript().

```
0 references
public void InscribeScript(ScriptWriteProvider scriptWriter)
{
    if (fgSave)
    {
        scriptWriter.AppendScript(_TMscript);
    }
}
```


TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

- Define the UserControl_Loaded function. Since users might use the same TMcraft Node on the flow project for multiple times, even these nodes share the same exe file, they must have their own configuration data. When loading the TMcraft Node UI, check if the data storage of the current node is empty or not (call DataStorageProvider.GetAllData(out data) and check if it returns a null data storage).

```
Dictionary<string, object> NodeConfigs = new Dictionary<string, object>();
result = NodeUI.DataStorageProvider.GetAllData(out NodeConfigs);

if(result == 0)
{
    if (NodeConfigs.Count == 0)//Node is implement at the first time.
    {
        TextB_Main.Text = String.Empty;
    }
```

- If yes, it denotes that the node is newly added; otherwise, get the previously saved configuration data by calling DataStorageProvider.GetAllData(key, out value). In this example, the text of the textbox is the only configuration data. Users can input the data into the textbox once they obtain it.

```
else if (NodeConfigs.Count == 1)
{
    string str_Content = string.Empty;
    result = NodeUI.DataStorageProvider.GetData("Content", out str_Content);
    if (result != 0)...
    else
    {
        TextB_Main.Text = str_Content;
    }
```

- Define the click event, Btn_Save_Click. Gather the text from the textbox, define it as _TMscript, then set fgSave to true, and finally close the Node Program."

```
1 reference
private void Btn_Save_Click(object sender, RoutedEventArgs e)
{
    string str = TextB_Main.Text;
    _TMscript = "Display(\"Green\", \"White\", \"Hello World\", \"" + str + "\")";

    if (NodeUI != null)
    {
        uint result;
        result = NodeUI.DataStorageProvider.SaveData("Content", str);
        string TMErrMsg = string.Empty;

        if (result == 0)
        {
            fgSave = true;
            NodeUI.Close();
        }
```


TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

Note

NOTE:
Remember to use the specific configuration: \"“+ string +“\" for double quoting to a string variable.

Note

NOTE:
There are some instructions recommended when using the TMcraft API functions:

1. Check if the TMcraftNodeAPI object is null or not
2. Use a uint object to get the result of the API function (replied by TMflow)
3. Check if the result is 0 or not; if not, call GetErrMsg() to get the corresponding error message
4. Be aware that errors might not only originate from TMflow but also from the API itself. These errors are represented as TMcraftErr objects and can be retrieved by calling GetErrMsg().

4.4 Compile and Test UI

Once the source code is ready, compile the code and generate the DLL file. Be reminded to double-check items below beforehand.

1. Right-click on the DLL Project on the Solution Explorer to open the Properties page.
2. On the Properties**Application** page, check the parameters. The most important is to make sure the **Output type** is **Class Library** and **Target framework** is **.NET 6.0**.

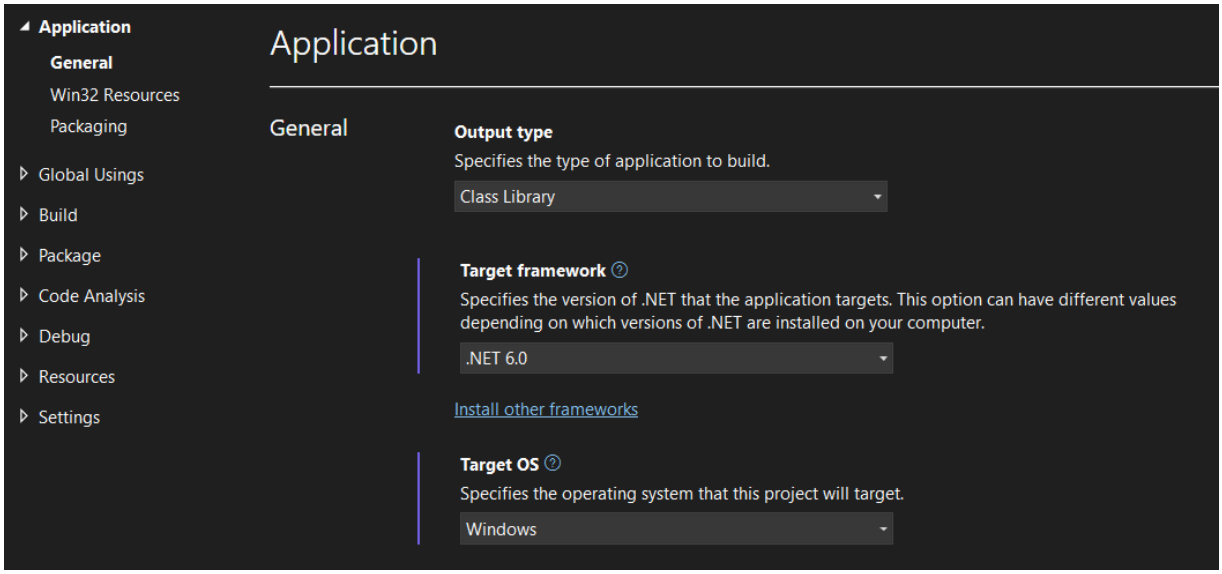


Figure 12: Set the Application Property before Building the UserControl Library

Before constructing the TMcraft Node Package with the UserControl DLL, users should test the UI functioning at the very first.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

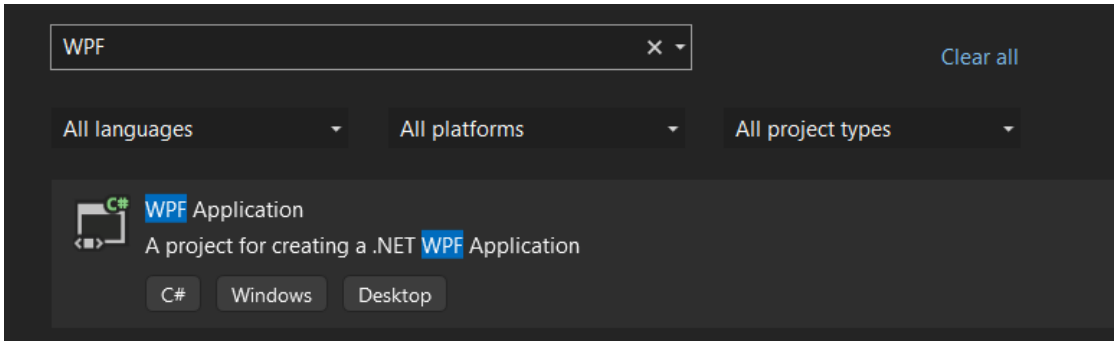


Figure 13: Create a WPF Application for UI Testing

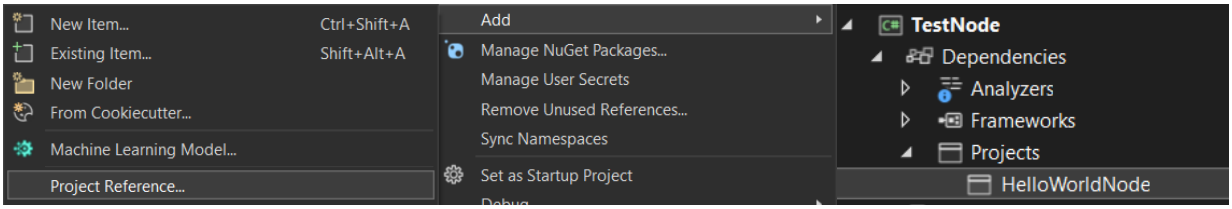


Figure 14: Add the HelloWorldNode User Control as Reference

The application source code is simple, i.e., to define a UserControl object with HelloWorld-Node.MainPage (namespace of the Node UI) and add it to the Grid. (Remember to name it first on the xaml file or the Properties page of the Grid)

```
using HelloWorldNode;

namespace TestNode
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    2 references
    public partial class MainWindow : Window
    {
        0 references
        public MainWindow()
        {
            InitializeComponent();
            UserControl nodeUI = new MainPage();
            gd_main.Children.Clear();
            gd_main.Children.Add(nodeUI);
        }
    }
}
```

Figure 15: Source code of the node testing application

Set the testing application as the Startup Project before compiling the code. Then, run the program to test the Node UI.

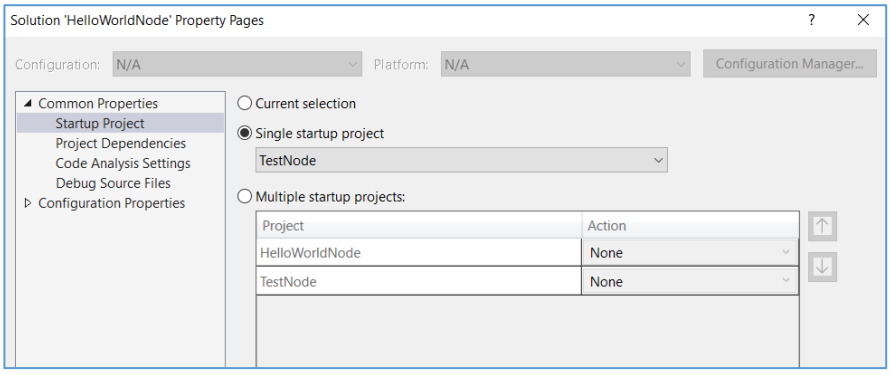


Figure 16: Set up the Startup Project

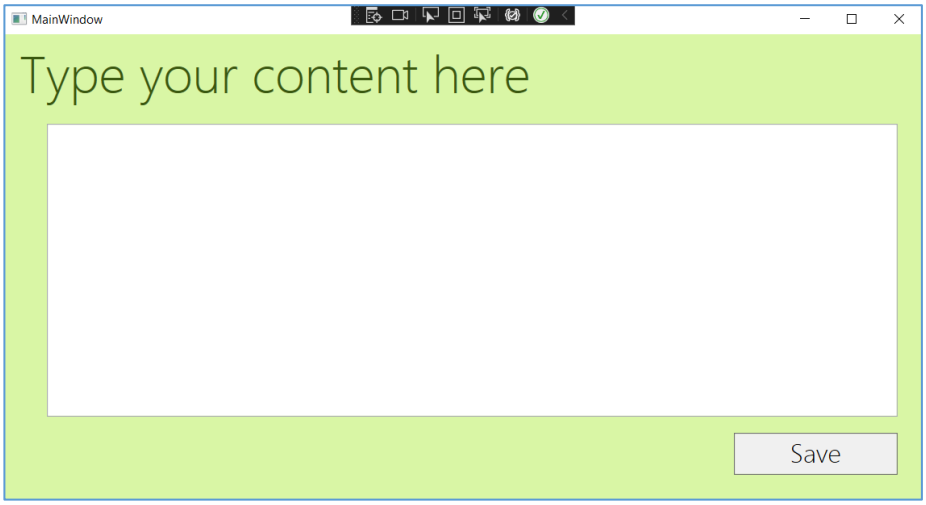


Figure 17: Test the Node UI

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

5. Build TMcraft Node Package

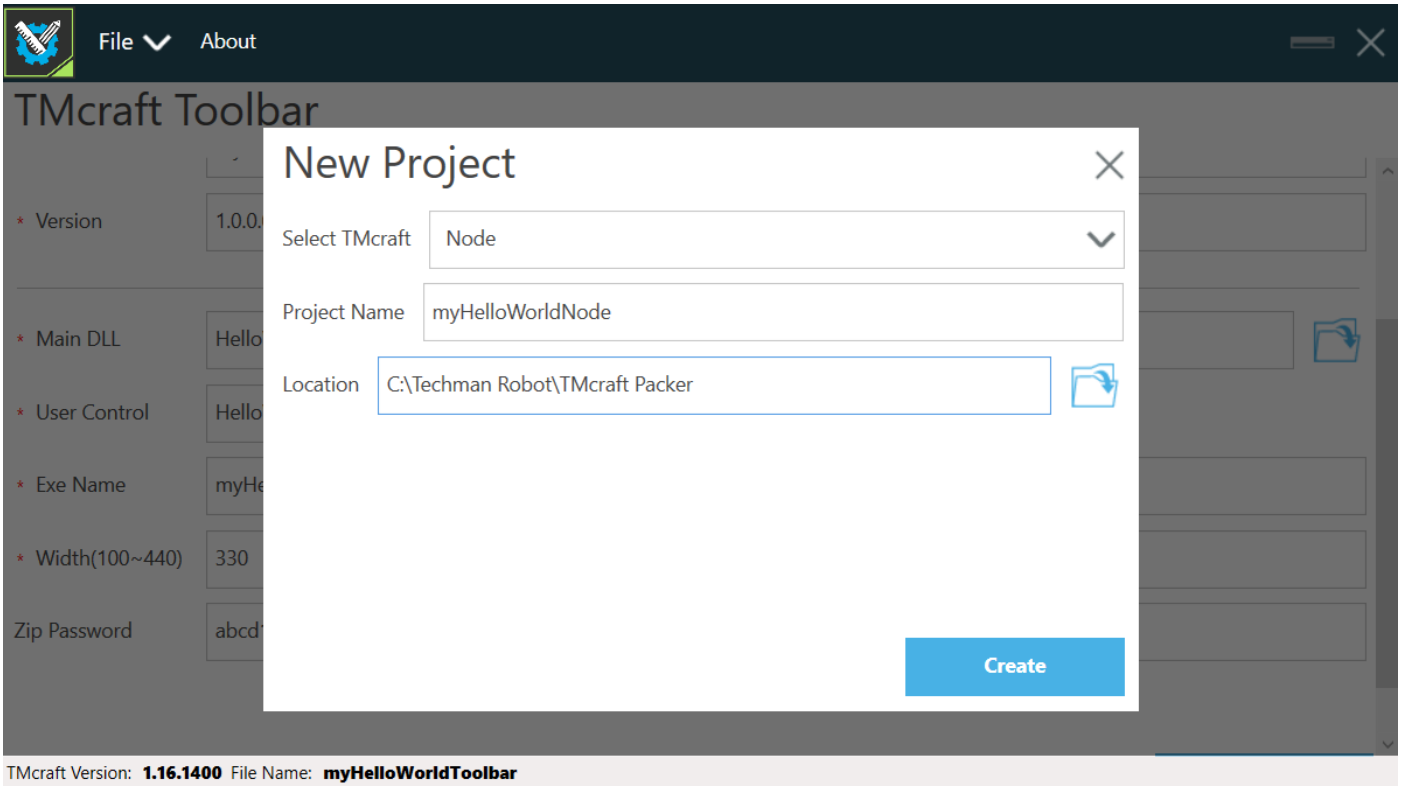
The previous section described how to build the TMcraft Node Program. Before using it on TMflow, it must be packaged using the TMcraft Packer.

First, prepare a source folder with the following items.

- The UserControl Library file (dll) of the Node UI
- All files from the TMcraft API (1.16 or above) folder from the development kit
- The Icon Image
- Other resource files, such as:
  - Other reference files used by the Program
  - Files used by the Node, such as media, documents, etc.

NOTE:
Suggest putting all files from the ..\bin\Debug folder into the source folder.

Next, open TMcraft Packer (1.12 or above) and create a project. Select **Node** as the target TMcraft item and complete the Project Settings. The system saves the project as a .tmcraft file at the specified file path in the **Location** field.



TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

Figure 18: Create a TMcraft Packaging Project

IMPORTANT:
Ensure that TMcraft Packer is running in the environment with .NET 6.0 installed.

After opening the project, users can see a form of parameters required to build and package the TMcraft Node. Refer to the following for explanations.

Parameters	Description	Requisite
Source folder	Path of the source folder	○
Target folder	Path of the target folder. For any new projects, the default target folder is ..\TMcraft Packer\USB\TM_Export*[PC Name]*\TMcraft\Node	○
Node Name	Name of the Node, which is shown on the Node List of the flow project	○
Provider	Name of the developer or the company providing this Node, which is shown on the TMcraft Management Page	
Version	The version of the TMcraft Node, which is registered onto the exe file and also shown on the TMcraft Management Page	○
Copyright	Copyright of the TMcraft Node, which is registered onto the exe file	
Main DLL	File name of the User Control Library of the TMcraft Node (as in Section 4)	○
User Control	The startup user control from the Main DLL. Since there might be several user controls within the User Control Library, it is necessary to define which one is used as the Main User Control. The format of this parameter should be *[Namespace].[UserControlName]*	○
Exe Name	Defines the name of the executable file generated. Be reminded that it should NOT be identical to the User Control Library file name.	○
Scale Transform	Denotes if the TMcraft Node would change the scale automatically by the resolution of the Control Box (1366 x 768 pixels). Disable the transformation if confident in the defined width and height.	○
Zip Password	Developers can define the password of the zip file. The password length should be between 6 and 256 characters of the non-case-sensitive Latin alphabet and numbers.	

※ Complete all parameters labeled with a red star in the TMcraft Packer GUI before proceeding.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

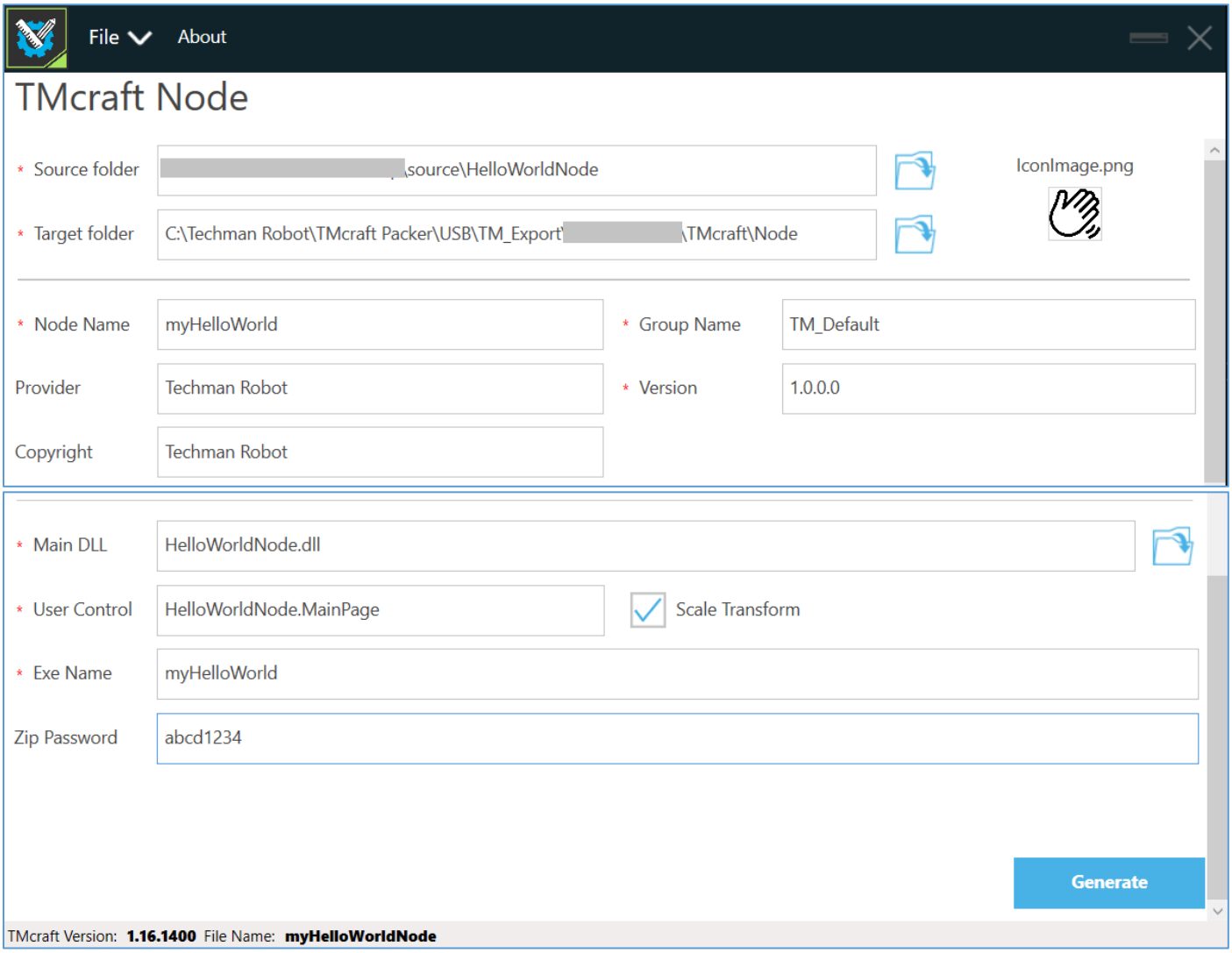


Figure 19: Set up the Packaging of TMcraft Node

Once all requirements are ready, click **Generate**. It might take several minutes to package, and users can check the current progress on the Page (Users will see an error message if anything goes wrong). After packaging the TMcraft Node successfully, users should see the successful status label.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

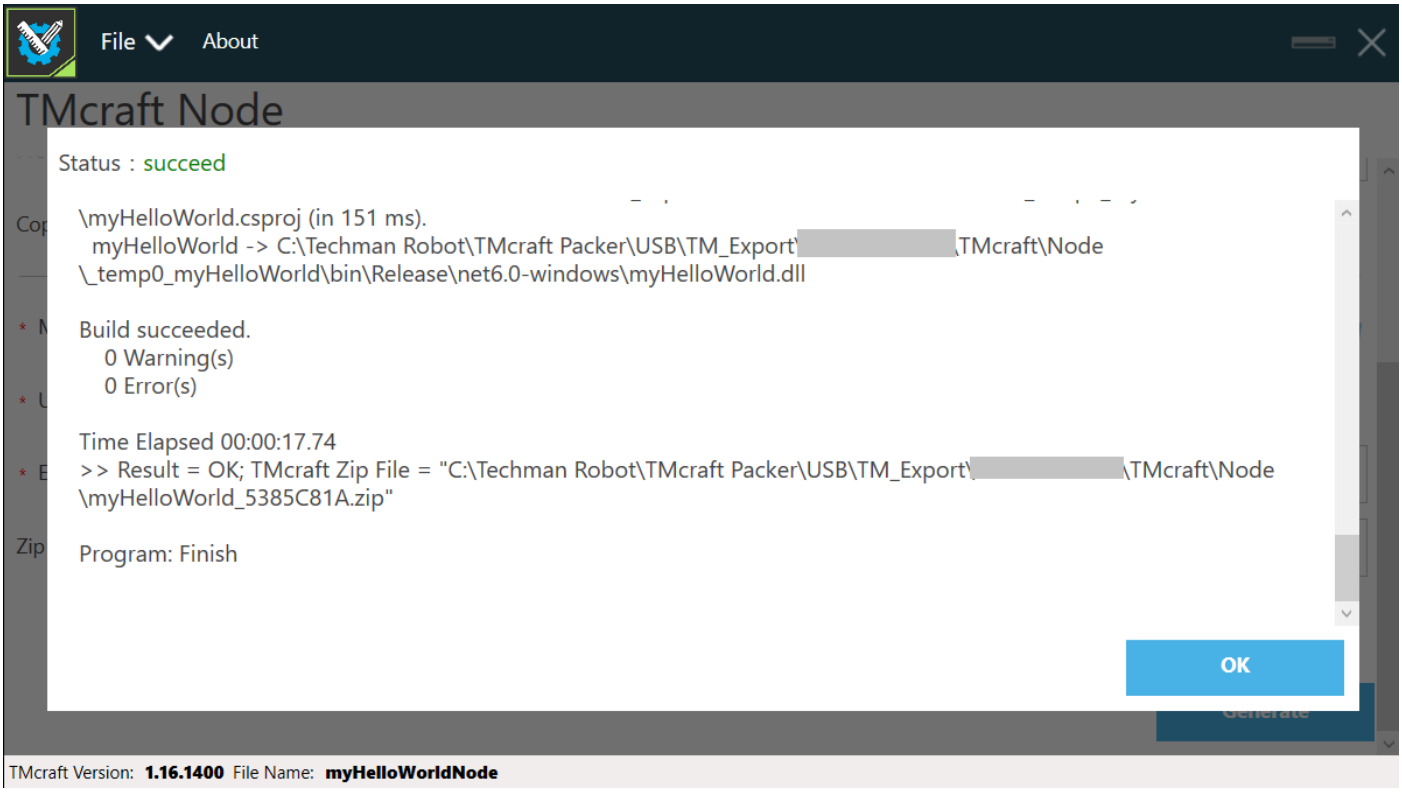


Figure 20: The Message Shown After Packaged Successfully

Finally, users can find the TMcraft Node zip file named after *[Exe Name]_[Checksum]* in the target folder.

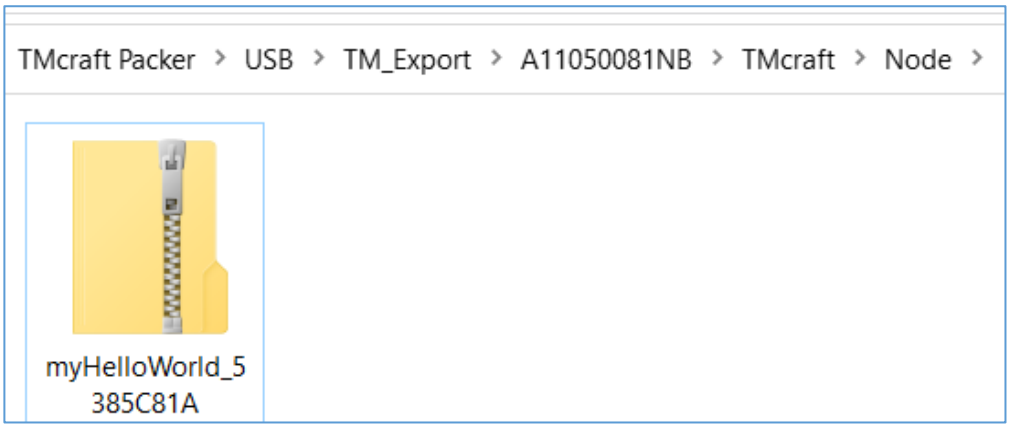


Figure 21: TMcraft Node zip at Target folder

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

6. Installation Code and Checksum

To protect the rights and interests of developers, each TMcraft Packer Project has its own encrypted Installation Code (GUID-based). Any TMcraft items generated by this project will share the same Installation Code and be able to replace one another on the same robot. Additionally, items generated from two different TMcraft Packer Projects (without identical Installation Codes) cannot replace each other, even if they share the same name and configuration. Therefore, developers should keep their packer project safe.

On the other hand, TMcraft Packer will also generate a checksum for each TMcraft item based on the binary footprint of the files (exe and dll) within the source folder and the Installation Code. The TMcraft item saves the checksum onto the configuration file. When importing the TMcraft item, TMflow will calculate a checksum by the same method; if these two checksums are not identical, TMflow will block the importing.

Developers can announce the checksum, allowing end users to verify it on the TMcraft Management Page to ensure the product's authenticity.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

7. Use TMcraft Node on TMflow

To use a TMcraft Node on both TMflow (robot) and TMflow Simulator, import the TMcraft Node Package to a robot by placing the zip file in the following path:

[Storage Drive]\TM_Export\[Folder]\TMcraft\Node

Plug the storage device into the robot and navigate to **System** > **Import/Export** > **Import**. Note that there is a new category: **TMcraft**. Select **TMcraft** > **Node**, choose the required TMcraft Node zip file, and click **Import**.

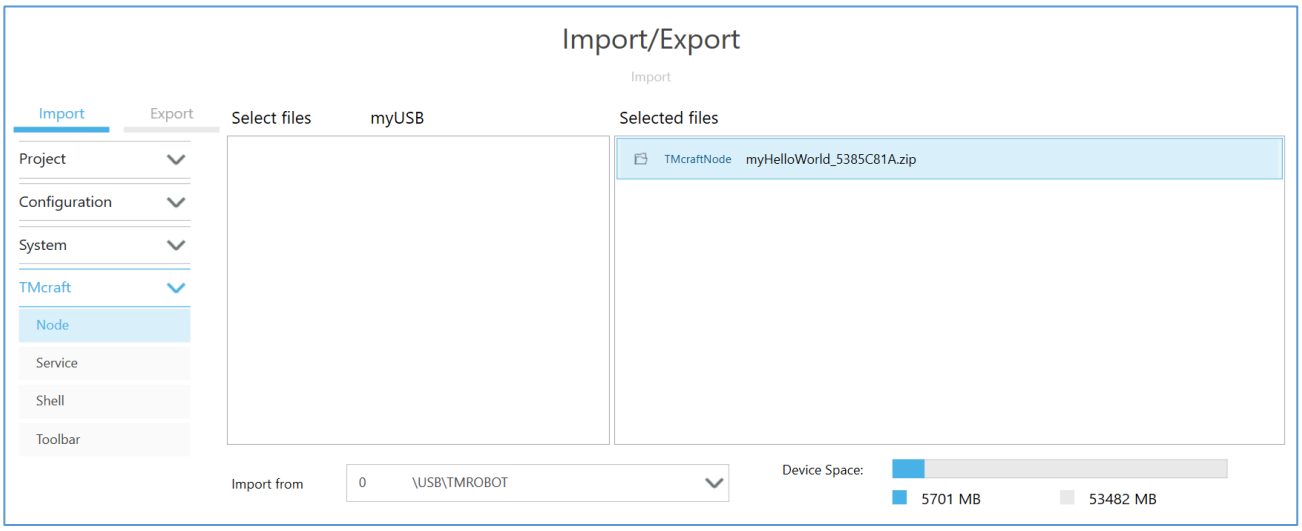


Figure 22: Import TMcraft Node zip

Next, navigate to **Configuration** > **TMcraft Management** > **Node**. Users can see the imported TMcraft Node shown in the table with the following information:

- **Name**: the executable file name of the TMcraft Node
- **Group Name**: the group name that the TMcraft Node belongs
- **Provider**: as defined in Chapter 5
- **Version**: as defined in Chapter 5
- **Checksum**: a number calculated based the binary footprint of the files (exe and dll) in the TMcraft Node folder. Developers may publish this for identification purpose to their end-users.

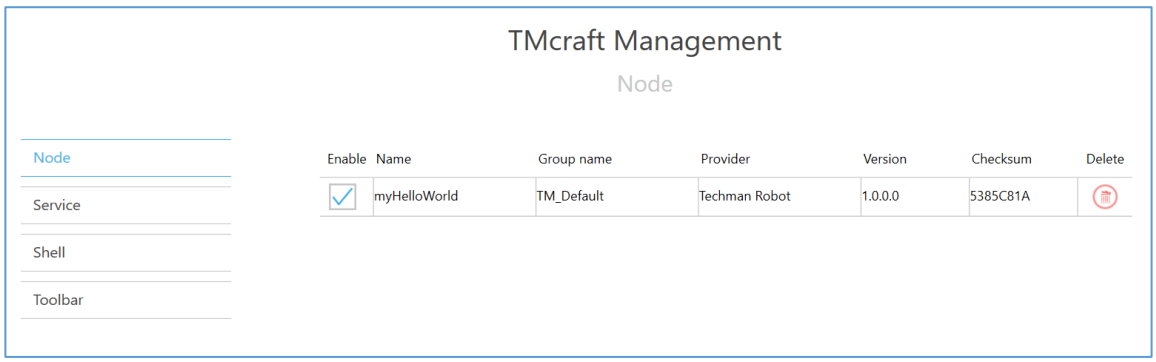


Figure 23: Enable the TMcraft Node on TMcraft Management

Create or open a flow project, and users can locate the TMcraft Node in its associated group in the node list. Drag the node into the flow and click the pencil icon to open the node UI. After setting the TMcraft Node, run the project. The relevant TMscript will execute when the flow reaches that node."

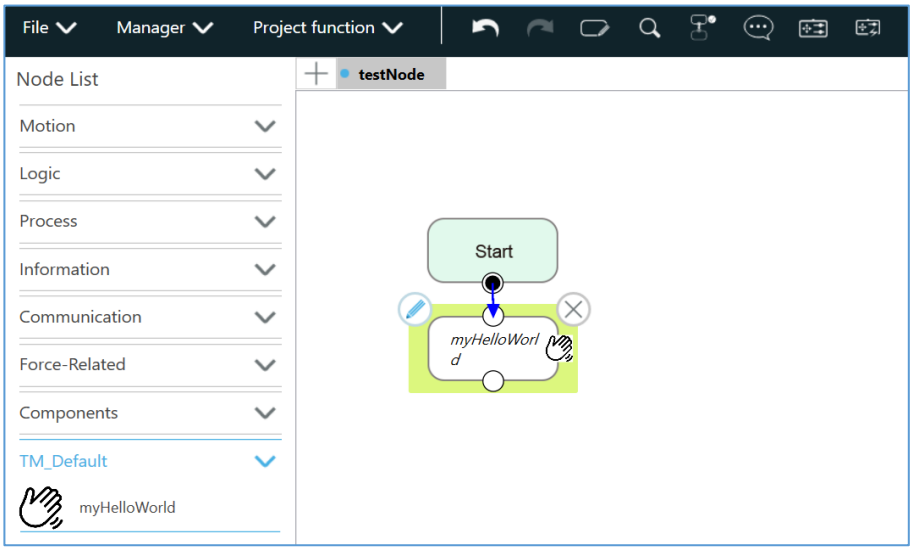


Figure 24: The TMcraft Node in Its Group

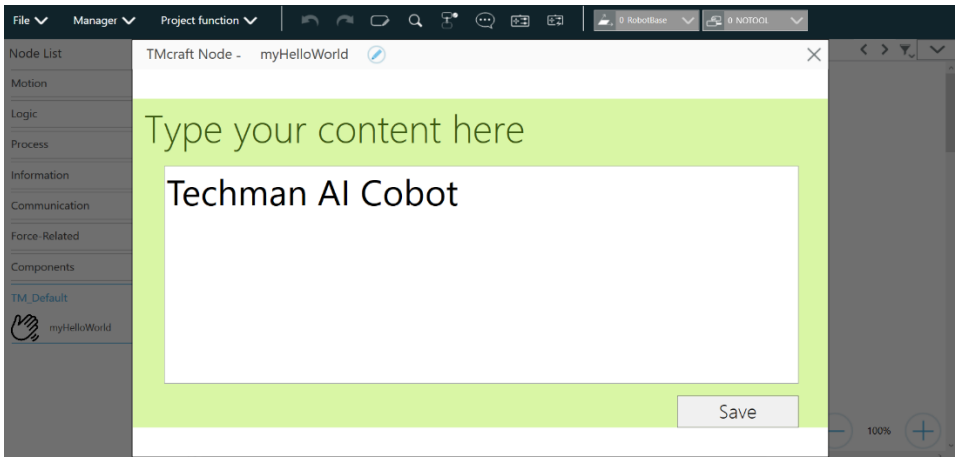


Figure 25: Edit the HelloWorld Node

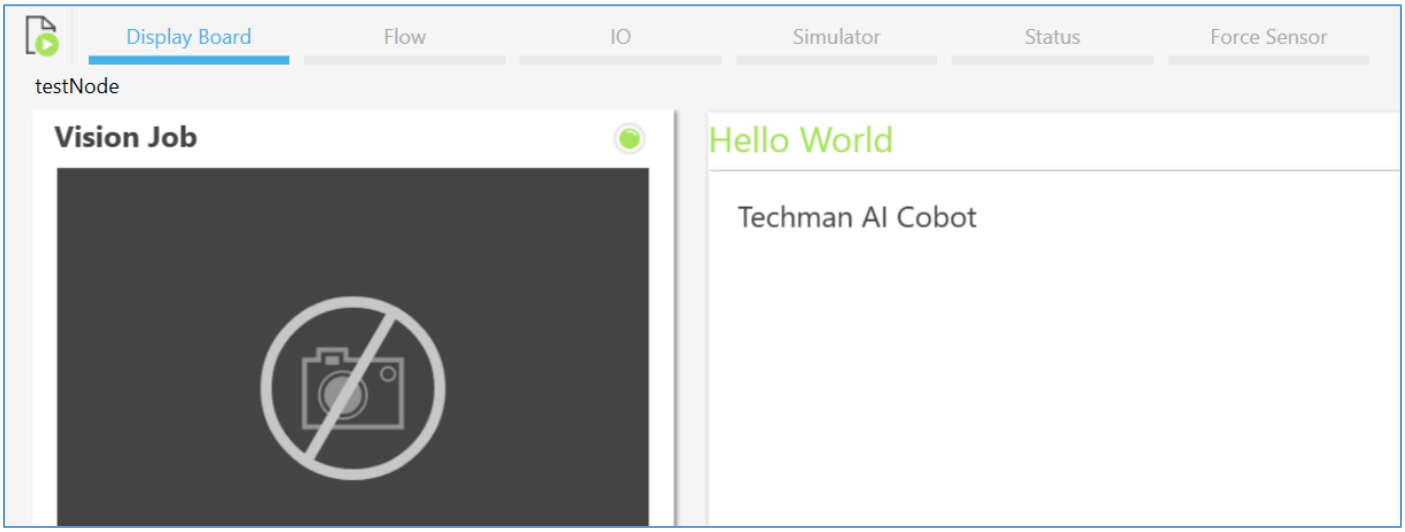


Figure 26: The Result of the HelloWorld Node at the Runtime

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

8. Use TMcraft Node in TMstudio Pro

Users can also import TMcraft Node Zip files to TMstudio. Launch **Import / Export**, select the TMcraft Node and import it onto a virtual robot.

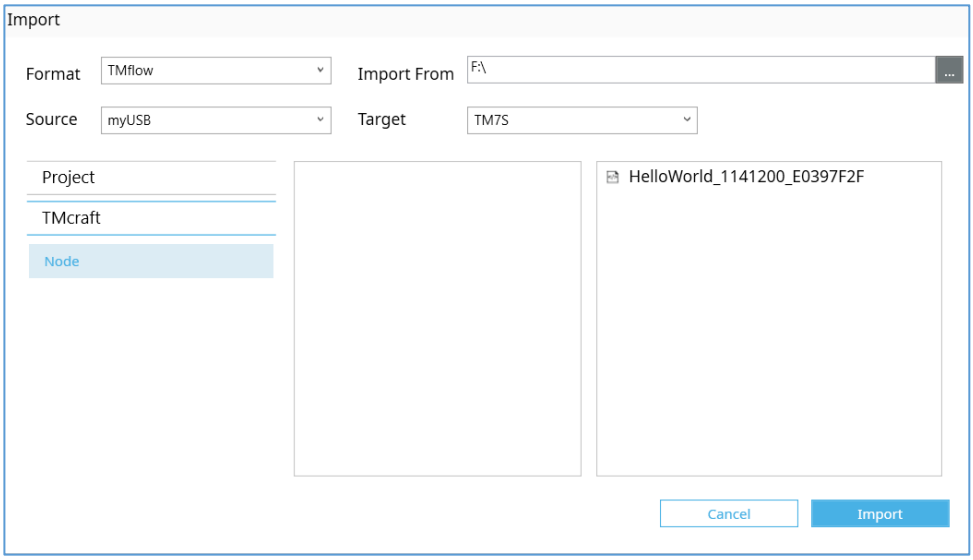


Figure 27: Export the TMcraft Node from a Robot Node

Then, users can find the TMcraft Node in the Node list of the flow editor and drag it into the flow project.

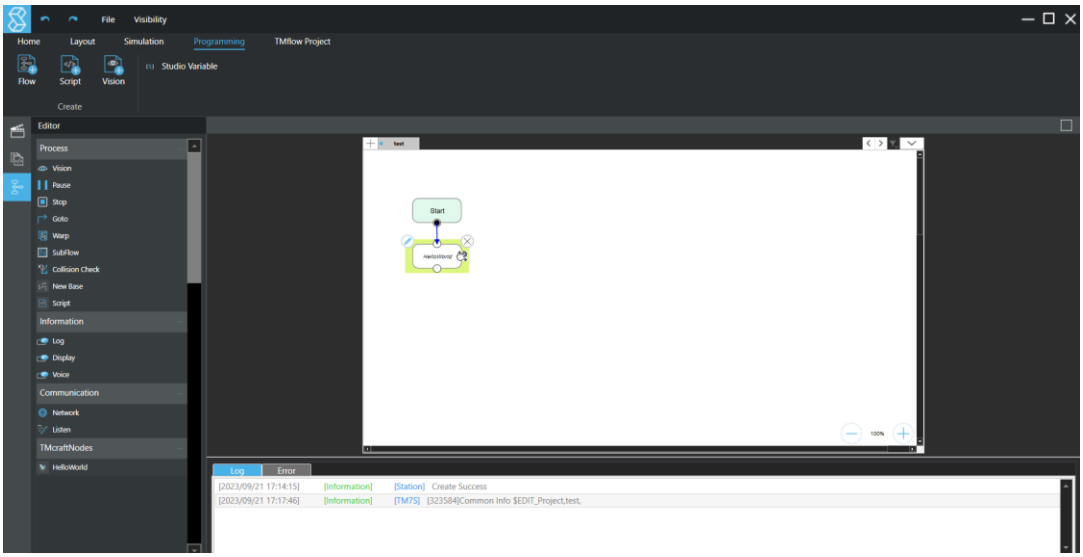


Figure 28: Use the TMcraft Node in TMstudio Pro

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan

9. Dos & Don’ts

- About program reference/libraries:

  If using NuGet to install and manage libraries within the TMcraft program, ensure the library files are accessible for placement in the TMcraft packaging source folder. Accomplish this by adding or modifying the property <CopyLocalLockFileAssemblies> to true within the csproj file. After compiling the project, users should find all reference files within the bin folder.

- About resource path definition:

  When using resources in the TMcraft Program, it is necessary to use an absolute path (pack://application:.../{Name of the User Control Library Project}; component/Resources/{String Resource file}) instead of the relative path to access resources; otherwise, it will not work.

- Suggestions on how to use TMcraft API functions:

  1. Check if the TMcraftNodeAPI object is null or not
  2. Use a uint object to get the result of the API function (replied by TMflow)
  3. Check if the result is 0 or not; if not, call GetErrMsg() to get the corresponding error message
  4. Note that, instead of TMflow error, there might also have error from the API itself; this error is represented as a TMcraftErr object and returned by calling GetErrMsg().

- Safety Assessment is necessary when using RobotJogProvider

  If a TMcraft Node utilizes RobotJogProvider functions for motion control, developers are responsible for ensuring a single point of control in compliance with ISO 10218-1.

- About writing files into Control Box through TMcraft items

  Robot System will deny the following access from TMcraft executable, including: (1) system shutdown and (2) accessing to drives C and D, except for the following folders:

  - D:\...TMflow\TMcraft\
  - D:\...TMflow\XmlFiles\
  - D:\...TMflow\TextFiles\

- Log

The TMcraft Program should save log files within its folder, which allows end-users to export the TMcraft zip, including the log files, for developer analysis.

- Before packaging with TMcraft Packer:
  - ✓ Place all API relevant files from the TMcraft Development Kit into the source folder.
  - ✓ Place all files from the bin folder of the TMcraft Project into the source folder
  - ✓ Ensure TMcraft Packer is running in the environment with .NET 6.0 installed.

TECHMAN ROBOT INC.
5F., No. 58-2, Huaya 2nd Rd., Guishan Dist., Taoyuan City, 333411 , Taiwan


TECHMAN ROBOT INC.