TMcraft Tutorial

Data Exchange between TMcrafts through Text Files

Original Instructions

Document version: 1.00

Release date: 2024-06-22

TECHMAN ROBOT INC.

Table of Contents

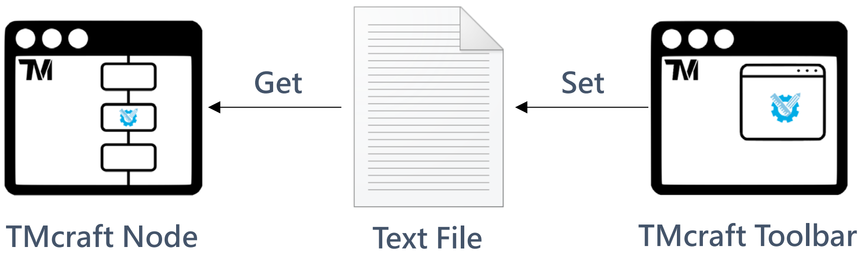
Revision History

Revision	Date	Description
1.00	2024-06-22	Original release

1. Introduction

This document explains exchanging data between TMcraft plugins using a text file. For instance, a TMcraft Toolbar writes specific data to a text file, and a TMcraft Node reads and utilizes this file in a project.

This tutorial is based on TMcraft API 1.18.1400.



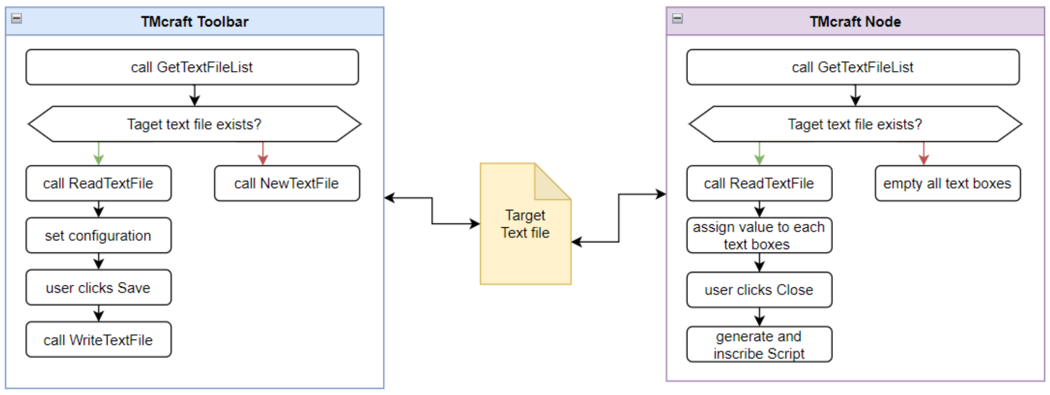
Readers must meet these prerequisites:

- Understand basic C# programming and WPF
- Have read *TMcraft Tutorial: Basic Development of TMcraft Node*
- Have read *TMcraft Tutorial: Basic Development of TMcraft Toolbar*
- Have read *TMcraft Node API Function Manual*
- Have read *TMcraft Toolbar API Function Manual*

2. Concept

TMcraft API comes with the TextFileProvider, a class that features functions for manipulating text files within TMflow. This tutorial utilizes the following functions:

GetTextFileList(out string[] list)	Retrieves the list of text file names in the current system.
NewTextFile(string filename, string fileContent)	Create a new text file.
WriteTextFile(string filename, string fileContent)	Write contents to the specified text file.
ReadTextFile(string filename, out string fileContent)	Read content of the specified text file.



When the TMcraft Toolbar opens, it activates the GetTextFileList to retrieve the text file list from the current TMflow and checks for the existence of the target text file named DataExchangeDemo. If yes, get the data by calling ReadTextFile and setting it as configuration. If the file does not exist, it calls the NewTextfile to create a new text file named DataExchangeDemo. After users modify the data and click Save, the program will gather the data, call the function WriteTextFile, and write the data onto the text file.

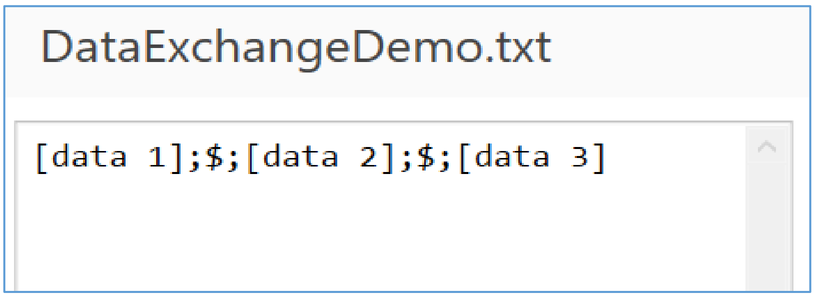
On the other hand, when the TMcraft Node opens, it checks for the target text file. If yes, get the data by calling ReadTextFile and set it as the configuration. If the file does not exist, create the text file. When the TMcraft node is about to close, it gathers the data, generates the corresponding script, and inscribes it to the current project.

3. Sample Code

This section discusses the fundamental components of the example's source code.

3.1 **Target text file: DataExchangeDemo.txt**

The text file includes a string that combines three pieces of data using **;$;** as the separator.



DataExchangeDemo.txt

```
[data 1];$;[data 2];$;[data 3]
```

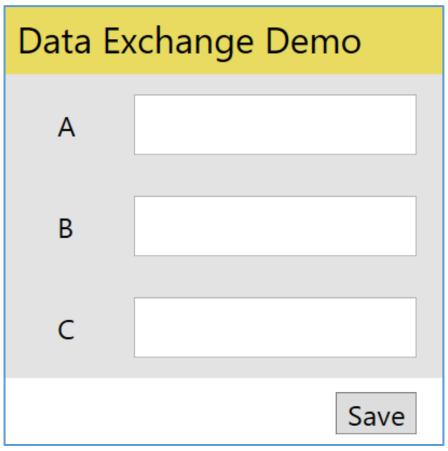
3.2 **TMcraft Toolbar: DataExchangeDemo_toolbar**

For the complete source code, refer to *TMcraft Development Kit_2.18|General Examples|[Node][Toolbar] Data Exchange Through Textfile.*

3.2.1 MainPage.xaml

MainPage.xaml defines the UI with the following elements:

- Textbox_*[A~C]*: These three textboxes allow users to input data.
- Btn_Save: Click the button to save the data onto the target text file.



3.2.2 MainPage.xaml.cs

When the UI opens, the program executes the UserControl_Loaded function. During the execution, the program initially invokes TextFileProvider.GetTextFileList to fetch a list of text file names from the current system.

```
private void UserControl_Loaded(object sender, EventArgs e)
{
    if (ToolbarUI == null || ToolbarUI.TextFileProvider == null)...

    try
    {
        string[] TextFileList;
        uint result = 0;

        result = ToolbarUI.TextFileProvider.GetTextFileList(out
          TextFileList);
```

Checks for the existence of the target text file named DataExchangeDemo.

```
bool HasTargetFile = false;
foreach(string TextFileName in TextFileList)
{
    if (TextFileName == TargetTextFile)
    {
        HasTargetFile = true;
        break;
    }
}
```

If it does not exists, it calls TextFileProvider.NewTextFile to create a new text file.

```
if(!HasTargetFile)
{
    result = ToolbarUI.TextFileProvider.NewTextFile
    (TargetTextFile, " ;$; ;$; ");
```

If it does, it calls getDataFromFile (will explain later in this chapter) to get the data array from the file and assigns values to the text boxes on the UI.

```
if (getDataFromFile(out data))
{
    TextBox_A.Text = data[0];
    TextBox_B.Text = data[1];
    TextBox_C.Text = data[2];
}
```

After the user interface loads, users can edit the text boxes and click "Save" to trigger the Btn_Save_Click function. This function checks for the existence of the target text file DataExchangeDemo, similar to the UserControl_Loaded function. If the file is absent, it creates a new text file containing the content from the UI text boxes.

```
if (!HasTargetFile)
{
    result = ToolbarUI.TextFileProvider.NewTextFile
    (TargetTextFile, fileContent);
```

If the file is present, it calls TextFileProvider.WriteTextFile to modify the text file.

```
result = ToolbarUI.TextFileProvider.WriteTextFile
 (TargetTextFile, fileContent);
```

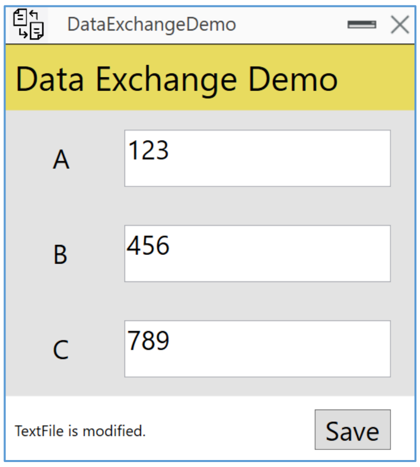
The getDataFromFrom function invokes TextFileProvider.ReadTextfile to retrieve a data string from the target text file and then splits it into a string array using the separator ;$;.

```
result = ToolbarUI.TextFileProvider.ReadTextFile(TargetTextFile,
  out str);
```

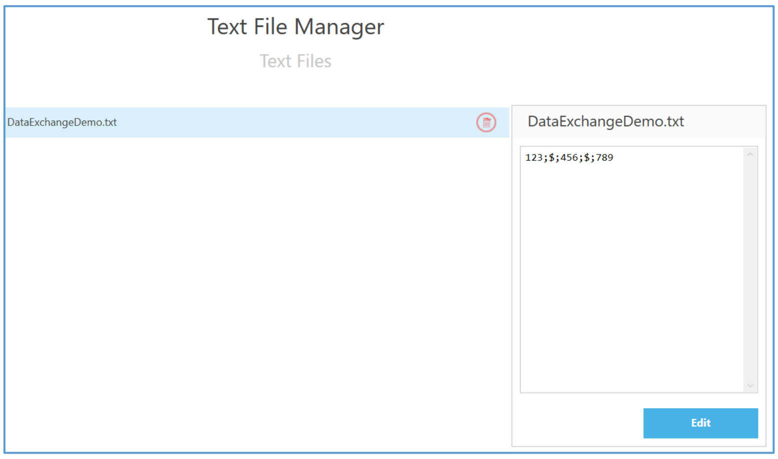
```
string[] Separator = { ";$;" };
Ary = str.Split(Separator, StringSplitOptions.None);
```

3.2.3 Quick demo

Open the toolbar, modify the data, and save it. Users can check the text file in the Text File Manager.



123
456
789
TextFile is modified.



123;$;456;$;789

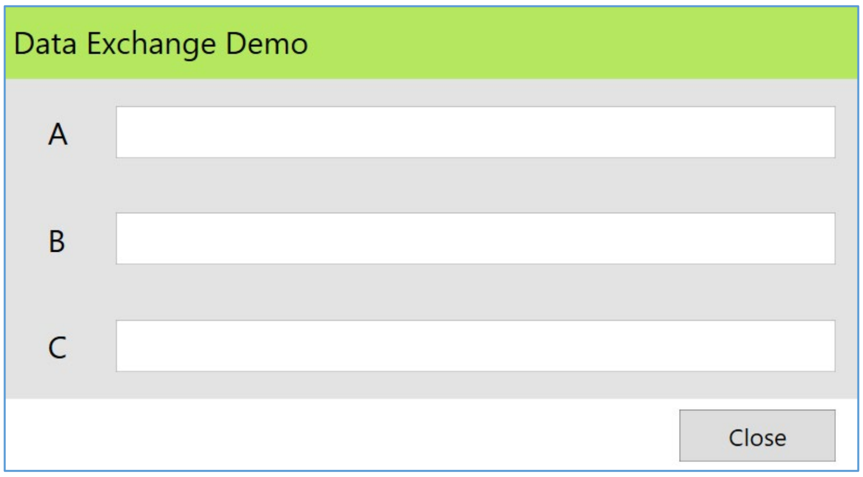
3.3 **TMcraft Node – DataExchangeDemo_node**

For the complete source code, refer to *TMcraft Development Kit_2.18|General Examples|[Node][Toolbar] Data Exchange Through Textfile.*

3.3.1 MainPage.xml

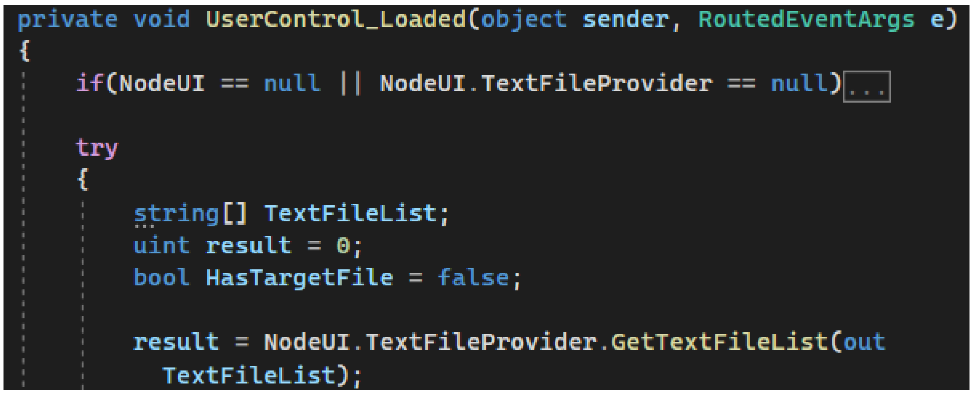
MainPage.xaml defines the UI with the following elements:

- Textbox_*[A~C]*: These three textboxes allow users to input data.
- Btn_Save: Click the button to close the node.



3.3.2 MainPage.xml.cs

When the UI opens, the program executes the UserControl_Loaded function. During the execution, the program initially invokes TextFileProvider.GetTextFileList to fetch a list of text file names from the current system.

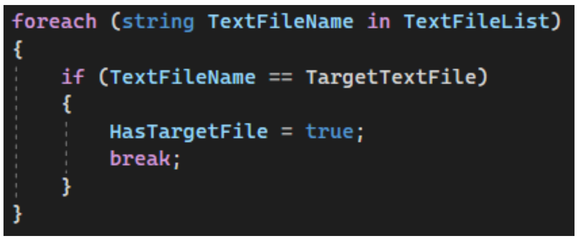


```
private void UserControl_Loaded(object sender, RoutedEventArgs e)
{
    if(NodeUI == null || NodeUI.TextFileProvider == null)...

    try
    {
        string[] TextFileList;
        uint result = 0;
        bool HasTargetFile = false;

        result = NodeUI.TextFileProvider.GetTextFileList(out
          TextFileList);
```

Checks for the existence of the target text file named DataExchangeDemo.



```
foreach (string TextFileName in TextFileList)
{
    if (TextFileName == TargetTextFile)
    {
        HasTargetFile = true;
        break;
    }
}
```

If the target text file exists, the system calls getDataFromFile (same as the function described in the previous section) to retrieve the data array from the file and assigns values to the text boxes on the UI. If the target text file does not exist, the system displays a message: Text File not Found.

```
if (getDataFromFile(out data))
{
    TextBox_A.Text = data[0];
    TextBox_B.Text = data[1];
    TextBox_C.Text = data[2];
}
```

After the user interface loads, users can check the data in the text boxes but not modify it. When the Node UI closes, the system executes the interface function InscribeScript. This function gathers the data in the text boxes, generates the corresponding script, and inscribes it to the current project.

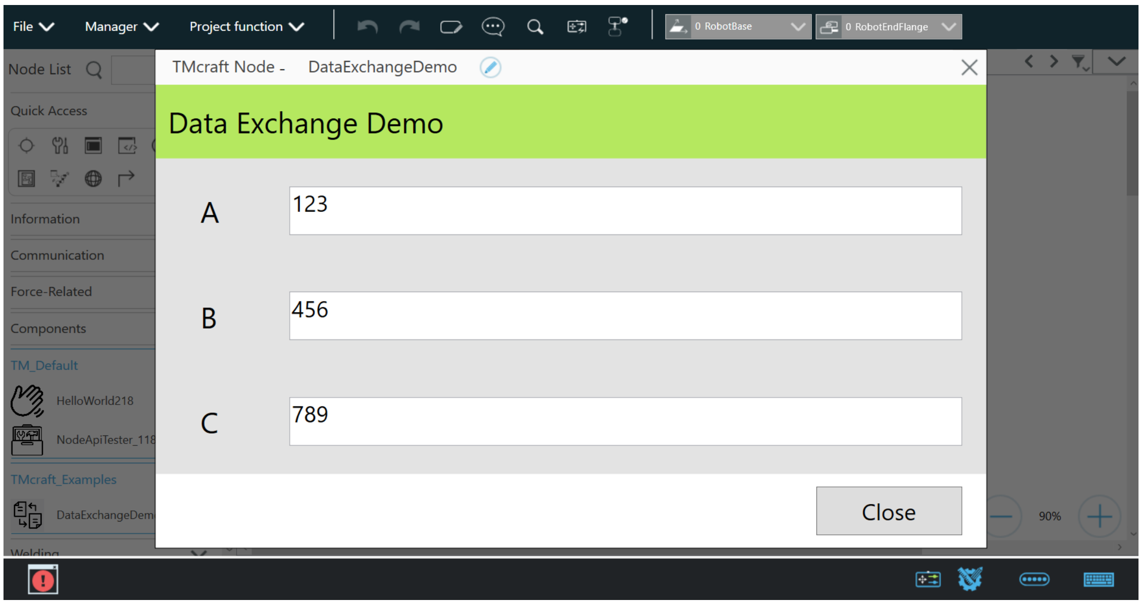
```
public void InscribeScript(ScriptWriteProvider scriptWriter)
{
    string str = "Data: " + System.Environment.NewLine + TextBox_A.Text +
      System.Environment.NewLine + TextBox_B.Text +
      System.Environment.NewLine + TextBox_C.Text;

    string script = "Display(\"" + str + "\")";

    try
    {
        scriptWriter.AppendLine(script);
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}
```

3.3.3 Quick Demo

After editing the data on the TMcraft Toolbar (refer to section 3.2.3), the same value appears on the TMcraft Node UI.



TMcraft Node - DataExchangeDemo

Data Exchange Demo

A 123

B 456

C 789

Close

TECHMAN ROBOT INC.