

# How to use "yFlowGen"(Automatic flowchart generation tool)

1	First of all .....	2
2	The function of this tool .....	2
3	How to use yFlowGen.exe .....	3
3.1	How to run yFlowGen.exe .....	3
3.2	Restrictions .....	4
3.3	Settings for generating GML files .....	4
3.3.1	Open using yEd Graph Editor .....	4
3.3.2	Use yEd Graph Editor to auto-arrange the flow chart .....	4
3.4	Settings for generating DOT+SVG files .....	5
3.4.1	Graphviz is required when using DOT files .....	5
4	About yFlowGenGUI .....	6
4.1	Unblock .....	6
5	Setting up to make it easier to see .....	7
6	About nkf32.exe .....	7
7	Operating environment .....	8
8	Terms of service .....	8
9	Disclaimer .....	8
10	Contact information .....	8
11	History .....	9

## 1 First of all

This document is a usage manual of the flowchart automatic generation tool "yFlowGen".

## 2 The function of this tool

Flowcharts are automatically generated by inputting C language (C, C++) source files to yFlowGen.exe.

The file format can be selected from "GML file" or "DOT file + SVG file".

- GML files require the installation of [yEd Graph Editor](#) software.
- DOT files require the installation of [Graphviz](#) software. In the case of DOT+SVG files, automatic alignment is also performed by the function of Graphviz.

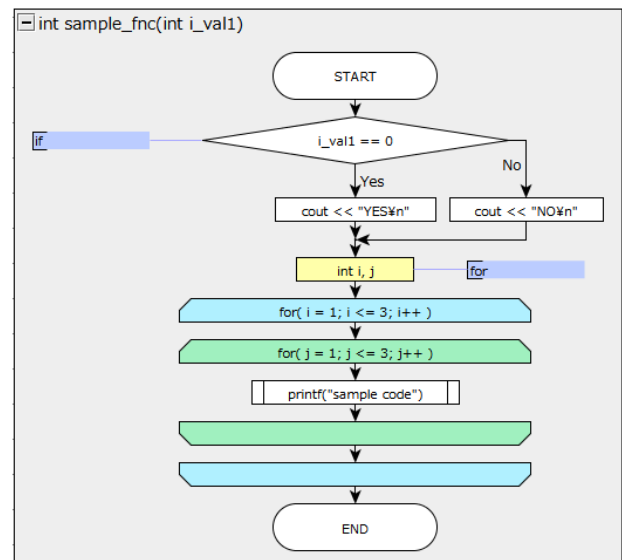
[Execution example] In case of GML file generation

Execute the following command from a windows command prompt.

In the example, sample.c is specified as the source file.

```
yFlowGen.exe -f sample.c
```

```
int sample_fnc(int i_val1) {  
    /* if */  
    if (i_val1 == 0) cout << "YES\n";  
    else  
        cout << "NO\n";  
    // for  
    int i, j;  
    for (i = 1; i <= 3; i++){  
        for (j = 1; j <= 3; j++){  
            printf("sample code");  
        }  
    }  
}
```

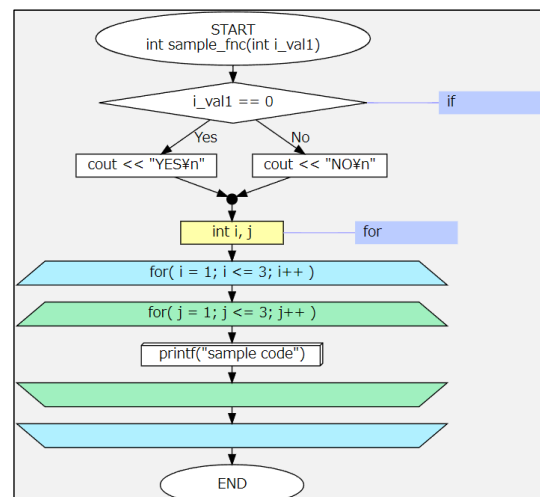


[Execution example] In case of DOT+ SVG file generation

DOT files can be generated by adding the option "-format dot".

```
yFlowGen.exe -f sample.c -format dot
```

```
int sample_fnc(int i_val1) {  
    /* if */  
    if (i_val1 == 0) cout << "YES\n";  
    else  
        cout << "NO\n";  
    // for  
    int i, j;  
    for (i = 1; i <= 3; i++){  
        for (j = 1; j <= 3; j++){  
            printf("sample code");  
        }  
    }  
}
```



### 3 How to use yFlowGen.exe

#### 3.1 How to run yFlowGen.exe

- 1) Please open a Windows command prompt.
- 2) Please execute yFlowGen.exe by specifying the source code after "-f" option.
- 3) Results are outputted in the flow\_file below the result\_yFlowGen folder.
- 4) The log is output to log\_yFlowGen.txt.

The following arguments can be specified when executing.

```
yFlowGen.exe -f <filePath> -no_compact -no_comment -out1file -out_group_comment
or
yFlowGen.exe -d <dirPath> -no_compact -no_comment -out1file -out_group_comment

-f <filePath> : Path of file to analyze. (It can be omitted if -d is described)
-d <dirPath> : Path of folder containing source files. (It can be omitted if -f is described)
-format <format> : Entering "-format dot" generates DOT and SVG files. (If omitted, GML files are generated.)
-out1file : Output 1 GML file per 1 source file. (Optional)
-no_compact : Turn off the setting to make the processing block size as small as possible. (Optional)
-no_comment : Do not display comments on flowchart. (Optional)
-no_color : Do not color the block. (Optional)
-no_dec : Do not display blocks with declarations only. (Optional)
-out_group_comment : Place comments outside the group. (Optional)
-no_disp_struct : Do not display struct and union. (Optional)
-no_reset : Do not delete result folder when script start. (Optional)
-true_false : Describe the judgment of if/else_if with True/False (described as Yes/No when not set)
-no_connection_point : No additional connection points (Optional. add connection points if not set)
-define <defA, defB, defC=1, etc>: Valid #define names. (Optional. To specify more than one, separate them by commas.)
-disp_invalid_def : Display invalid #define descriptions as comments. (Optional)
-left_flow_is_no: Left of IF branch is NO, right is YES. (Optional)
-add_extension <cxx, cs, etc> : Additional extension. (Optional)
-pj_name <your_project_name> : Project Name. (Optional)
-copy_org_code : Copy the input file source code to the results folder. (Optional)
```

Example:

```
yFlowGen.exe -f sample.c -out1file
```

- If you specify a file name after -f, create a folder of "result\_yFlowGen¥<source file name>" in the place where yFlowGen.exe is executed. Flowchart files are output for each function.  
The list of results is output to "result\_yFlowGen.html".
- If you specify a folder name after -d, yFlowGen searches for source files (.c, .cpp) written in C language under the specified folder. Then yFlowGen creates the same folder as the hierarchical structure of the found folder and stores the result.
- In addition to the execution method at the command prompt, a method to execute it with the GUI is also available separately.  
[\(→yFlowGenGUI\)](#)
- When using #define in C code, put a space after -define and list valid #define names separated by commas (,). (Example: -define AAA,BBB,CCC=1,DDD=-1)
- yFlowGen outputs the result list to "result\_yFlowGen.html" and "result\_yFlowGen¥index.html".

- Also, if you specify the project name using -pj\_name <project name>, result\_yFlowGne (folder, html) will be generated as <project name>\_result\_yFlowGen. Generates result\_yFlowGen if you do not use the -pj\_name option.

### 3.2 Restrictions

- It may not work if 2 byte characters (such as Japanese) are included in the path specified by "-f" or "-d". Therefore, please do not write 2 Byte characters in the path. If 2 Byte characters are included other than "place surrounded by" such as "printf, cout" and "comment", it may not work properly due to character code problem. If you want to use 2 Byte characters for variables etc, enclose Japanese with "". (Eg: for ("count" = 0; "count" <100; "count" ++) etc)
- Descriptions of enum, struct, typedef, and class are not output as flowcharts.
- In try-catch exception handling, the final block of catch and the next block are connected so that the figure can be easily seen.

### 3.3 Settings for generating GML files

#### 3.3.1 Open using yEd Graph Editor

The generated file is in gml file format. Therefore, please use Ed Graph Editor to open.

yEd Graph Editor can be downloaded for free from the following.

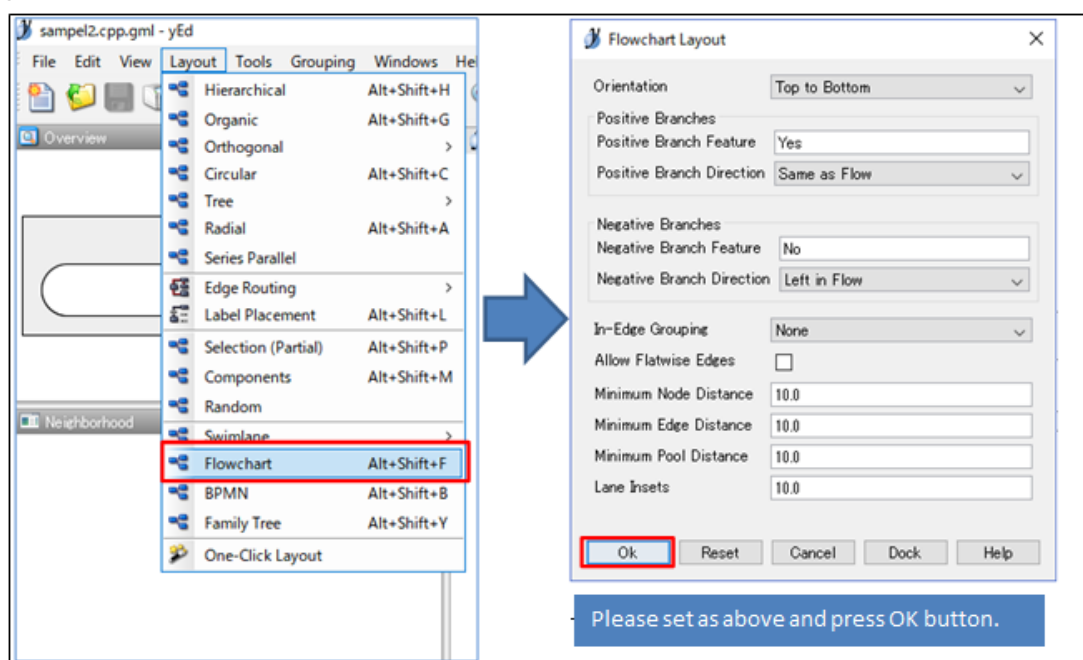
Since yEd Graph Editor is a wonderful graph editor, please use by all means when writing flow chart by hand.

[[yWorks] yEd Graph Editor - Downloads]

<https://www.yworks.com/downloads#yEd>

#### 3.3.2 Use yEd Graph Editor to auto-arrange the flow chart

Open the gml file. Then, using the automatic sorting function of yEd Graph Editor, arrange the figures automatically. Please click "Layout → Flowchart" on the tool bar. Then set as follows and execute auto alignment.



### 3.4 Settings for generating DOT+SVG files

#### 3.4.1 Graphviz is required when using DOT files

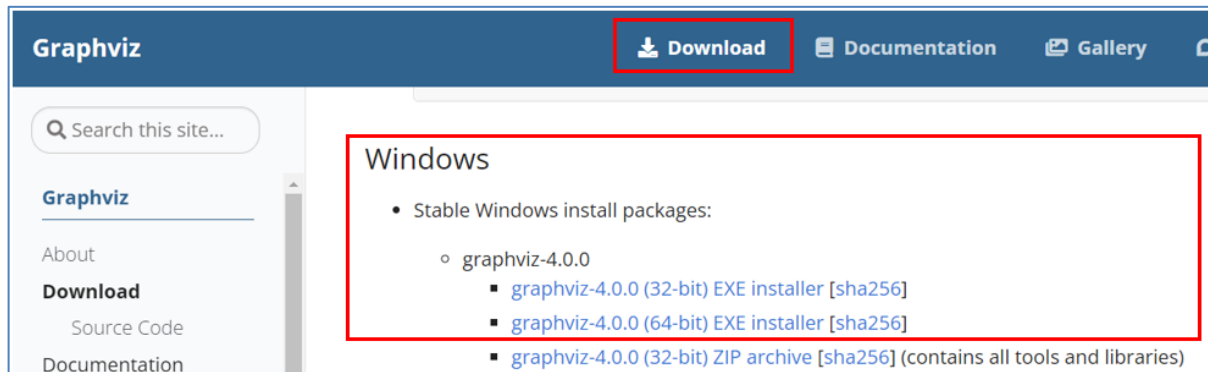
To generate DOT+SVG files, Graphviz must be installed beforehand.

yFlowGen converts the generated DOT file into an SVG file using Graphviz.

Graphviz can be downloaded from the following link.

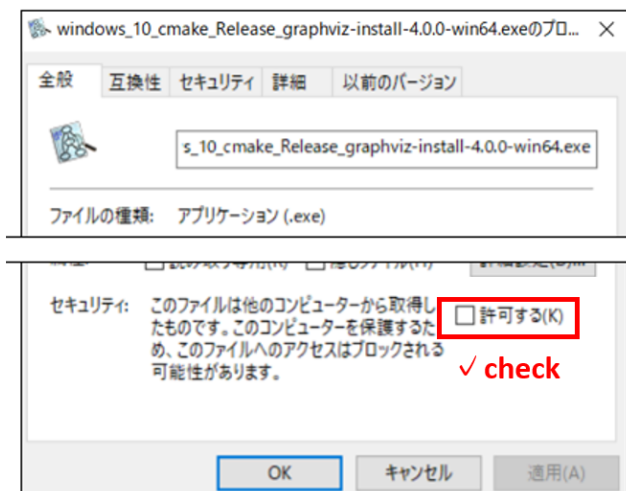
If you are using Windows 10, please use the 64-bit version of the Stable package.

[Download | Graphviz] <https://graphviz.org/download/>



The Graphviz installer may be blocked by security on some PCs.

In this case, right-click on the installer, open the properties, and select ✓ for "Allow" in the red box below.



## 4 About yFlowGenGUI

yFlowGenGUI is an auxiliary tool that can run yFlowGen.exe with GUI. For details, please see the "Useage" sheet of "yFlowGenGUI.xlsm" located at the same level as this file.

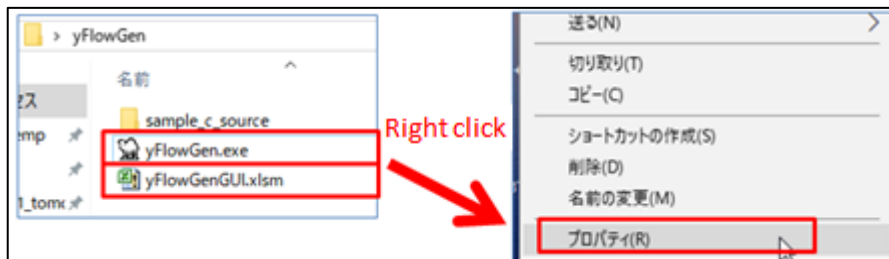
### 4.1 Unblock

Files downloaded from the Internet by Windows cannot be executed unblocked.

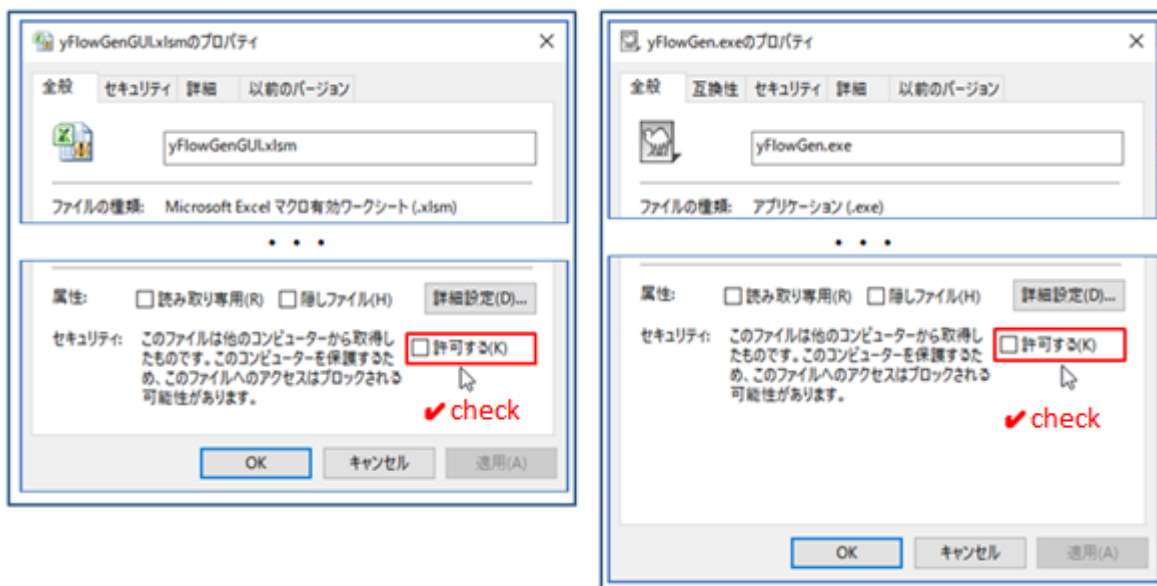
For this reason, please unblock in the following way.

(It seems that it can be executed without unblocking from the command prompt)

[Step 1] Right-click the file and press Properties.



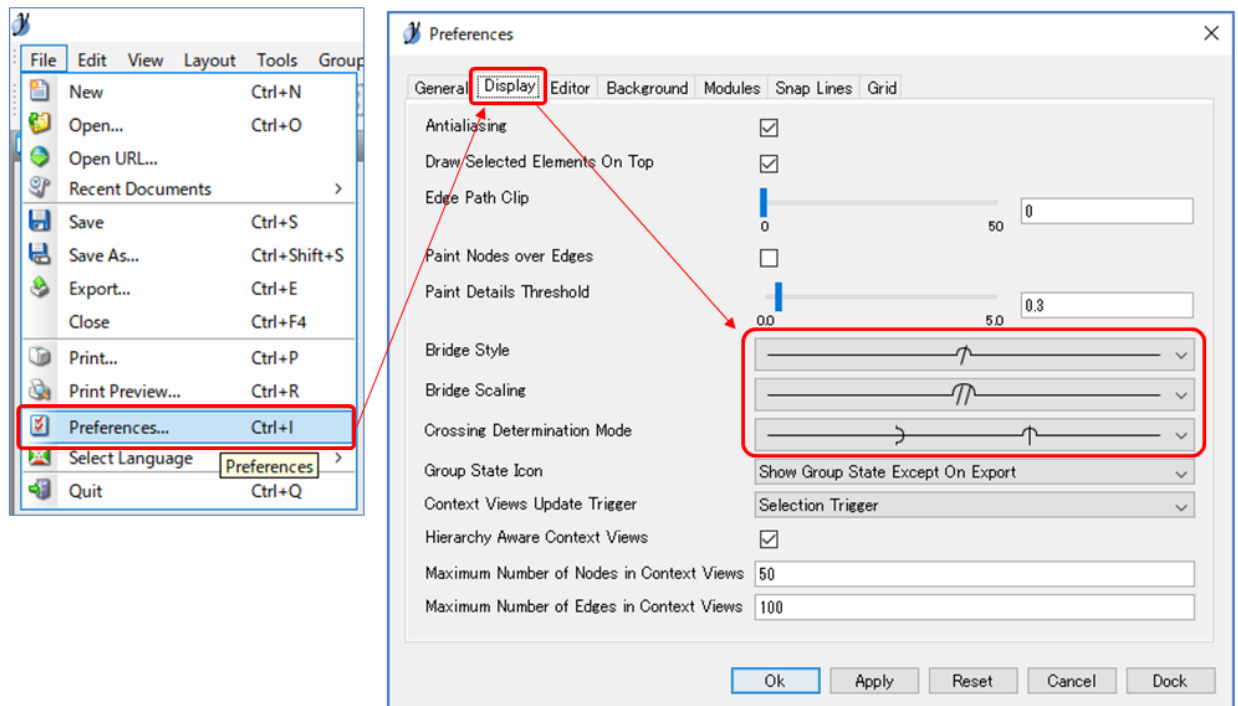
[Step 2] Please check the box of "Allow" in the security item.



## 5 Setting up to make it easier to see

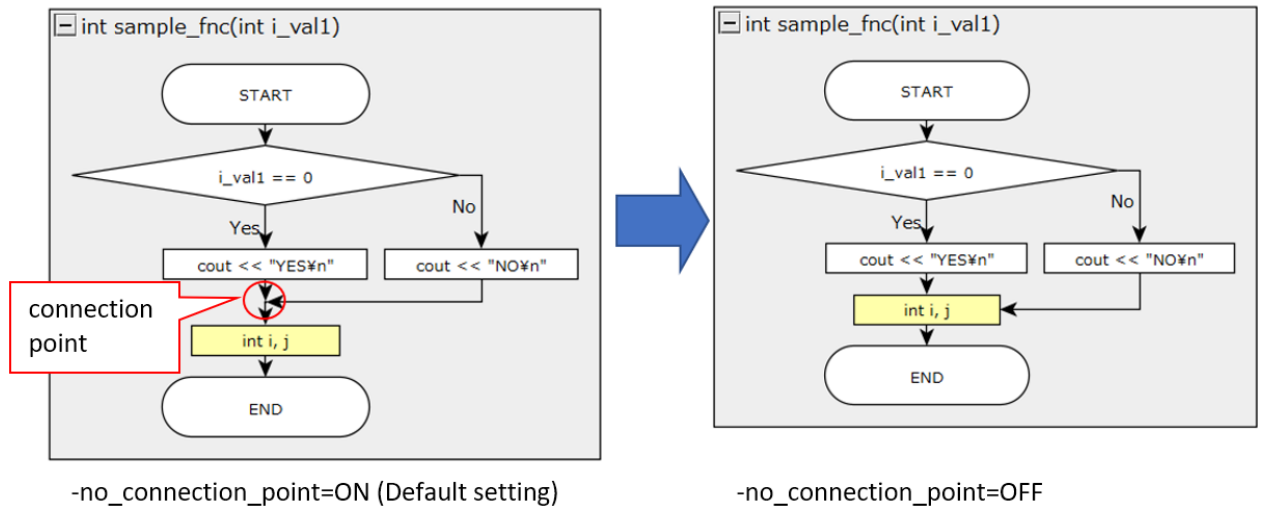
### 5.1 Bridge Style setting

You can choose the expression when the lines crossed. Please set "Bridge Style" in "Preferences" → "Display" tab of "yEd Graph Editor". The first setting is not a bridge expression, I think that it is easy to understand with this setting.



### 5.2 "No additional connection points" setting

If you want to display without adding connection points, turn on the "No additional connection point" setting (-no\_connection\_point).



## 6 About nkf32.exe

nkf32.exe is obtained from the following URL and redistributed. Various character codes of input files were supported using nkf32.exe. Please put it in the same hierarchy as yFlowGen.exe.

[nkf.exe : Vector] <https://www.vector.co.jp/soft/dl/win95/util/se295331.html>

(Author URL : <http://hp.vector.co.jp/authors/VA007219/>)

## 7 Operating environment

- OS: Windows 7 or higher is recommended

## 8 Terms of service

This program is freeware. toowaki has copyright. Please reprint freely.

Also, please do not change or change this software and attached Excel sheet.

## 9 Disclaimer

Regardless of damage caused by using this program, the author is not involved at all.

Please use this program at your own risk.

## 10 Contact information

If you have any requests, please email [toowaki.fc2@gmail.com](mailto:toowaki.fc2@gmail.com).



## 11 History

Date	Contents	Ver.	Editor
2018/1/5	Newly created.	1.0	toowaki
2018/5/2	When "printf("text");}" was executed, yFlowGen could not output correct result. So, this bug was collected.	1.1	
2018/11/8	Fixed about the following 3points. - "}" For indentation is continuous on the same line. - "}" Comes after ";". - When describing an else statement with one line and describing without using "{".	1.2	
2018/11/23	· When "(" or ")" exists in printf, cout, it was not working properly and it was fixed.	1.3	
2018/12/09	Some bugs were fixed.	1.5	
2018/12/30	Corresponded to class and struct.	2.0	
2019/01/03	Various character codes of input files were supported using nkf32.exe.	2.1	
2020/06/14	Fixed the connection with the break statement.	2.2	
2020/08/29	Fixed the connection with the default section of the switch.	2.3	
2020/08/30	Fixed the connection with the break statement.	2.4	
2021/06/26	Fixed the connection with the break statement of case.	2.5	
2021/07/17	In case of if-else description, it had been changed so that the process works properly even if it is not described in else.	2.7	
2021/07/17	Fixed connection when using if.	2.8	
2022/03/21	Added the option "-no_connection_point "	2.9	
2022/04/10	Fixed the connection with the break statement.	2.10	
2022/07/03	- Generates DOT+SVG files using Graphviz. - Generates a list of generated results as "result_yFlowGen.html.	3.0	
2022/09/11	- Added the option "-ignoredef" (specifying an invalid #define name).	3.1	
2022/09/18	- Added the option "-disp_invalid_def" (display invalid #define descriptions as comments).	3.2	
2023/02/12	- Added the option "-left_flow_is_no" (set IF branch left to NO, right to YES).	3.3	
2023/02/18	- Support for uppercase letter extensions - Added option "Additional extension".	3.4	
2023/07/01	- The exit and _exit functions are now connected to the end block.	3.5	
2023/07/18	- Added the option "-pj_name" (project name). - Supported the case where multiple processes are described using ";" in one line. For example, "if (a) sub01(); else sub2();".	4.1	
2023/07/22	- Fixed blocking connection when handling if condition is empty.	4.2	
2023/08/12	- Supported #if-#elif.	4.4	
2023/11/04	Added option "-copy_org_code". If "-copy_org_code" is set, it will copy the input source code to the results folder. This allows you to move and open just the results folder to another environment and still be able to see the code.	5.0	
2023/11/11	- Support when the input file path specified with options -d or -f contains double-byte characters.	5.1	