

Programming without coding technology

تكنولوجيا البرمجة بدون كود



Version 1.0
(Stable) Rev. 8

الاصدارة الاولى
المراجعة الثامنة

(1) Mahmoud Programming Language

(2) RPWI Environment

(3) DoubleS (Super Server) Paradigm

(١) لغة البرمجة محمود

(٢) بيئة البرمجة بدون كود

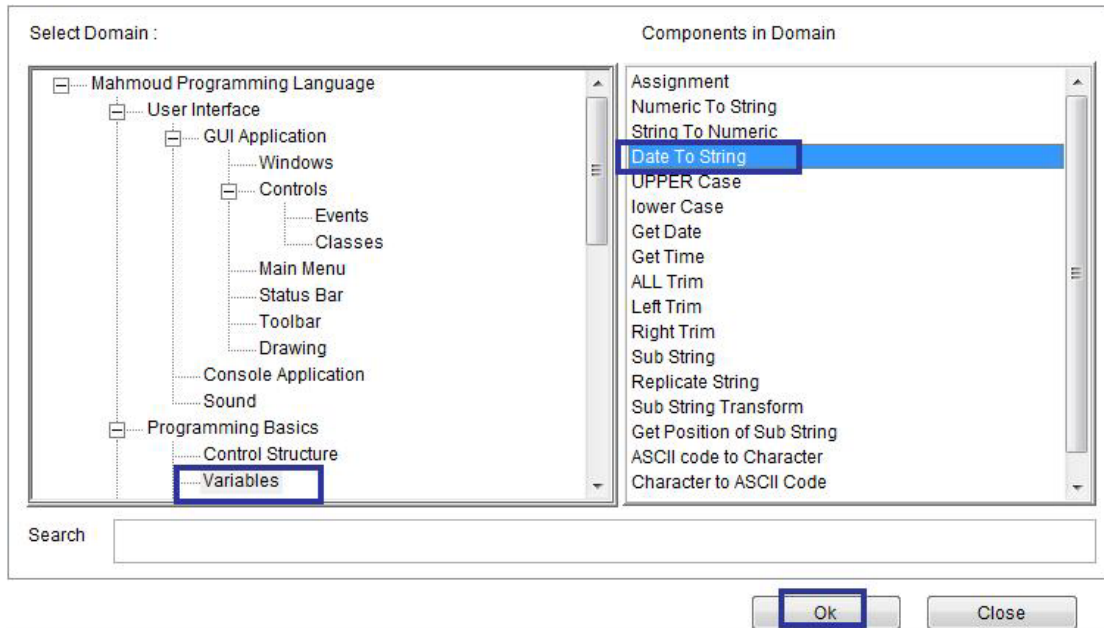
(٣) نمط البرمجة الخادم الممتاز

<http://www.sourceforge.net/projects/doublesvsoop>

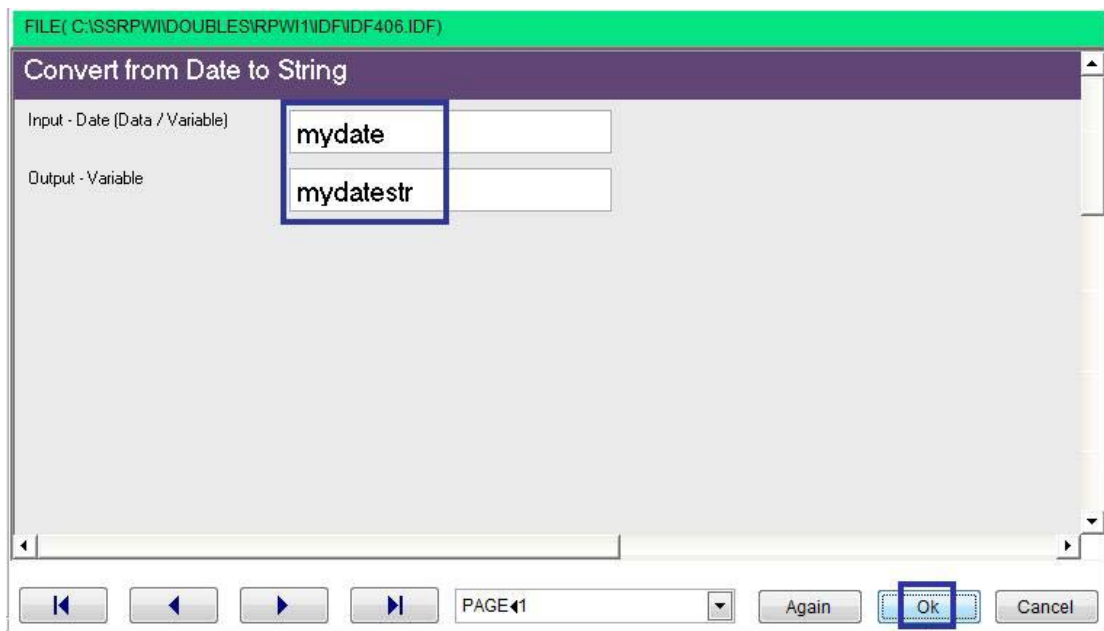
By
Mahmoud Fayed
msfclipper@users.sourceforge.net

جدول المحتويات

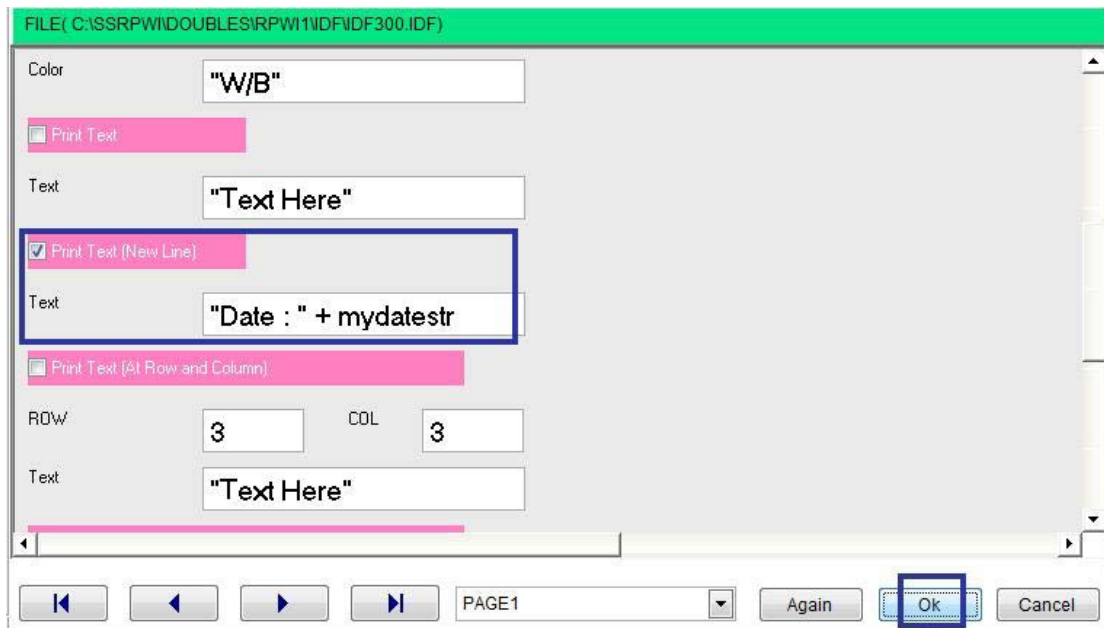
الموضوع	رقم الصفحة
Introduction مقدمة	3
Mahmoud Programming Language لغة البرمجة محمود	12
Hello World مرحبا بالعالم	14
Setting Colors & Clearing Screen اختيار الالوان ومسح الشاشة	22
Clearing a rectangle area, drawing a box مسح مساحة ورسم مستطيل	26
Variables Assignment ضبط المتغيرات	29
Strings العبارات الحرفية	33
Numerical variables and arithmetic operations المتغيرات الرقمية	54
Logical Variables and logical operations المتغيرات المنطقية	71
Expressions & Macro التعبيرات والماكرو	83
Date and Time الوقت والتاريخ	90
Converting between data types التحويل بين انواع البيانات	94
ASCII code كود الاسكي	103
Getting Input from User استقبال المدخلات من المستخدم	107
Menus القوائم	113
IF Statement الجملة الشرطية اذا	118
For Loop الحلقة التكرارية باستخدام العداد	128
While Loop الحلقة التكرارية باستخدام شرط	133
Loop and Exit اللف والخروج	141
Error Handling (Try – Catch) معالجة الاخطاء	142
Memo variables متغيرات الملاحظات	143
Arrays المصفوفات	155
Files الملفات	162
Structure Programming البرمجة الهيكلية	170
Database Files ملفات قواعد البيانات	177
GUI Applications التطبيقات الرسومية	203
GUI – Controls (Objects, Events & Classes) عناصر التحكم	206
Form Designer صمم التماذج	216
Language Extension امتداد اللغة	218



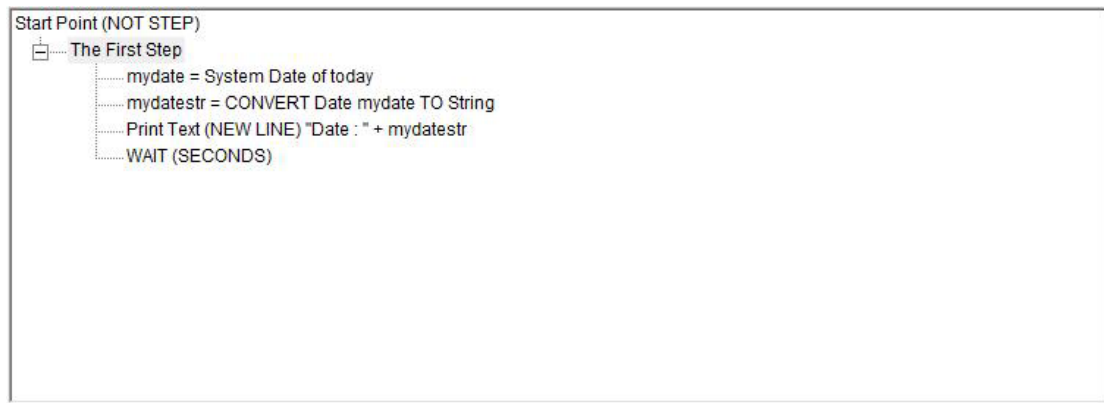
Domain (Variables) – Component (Date to String)



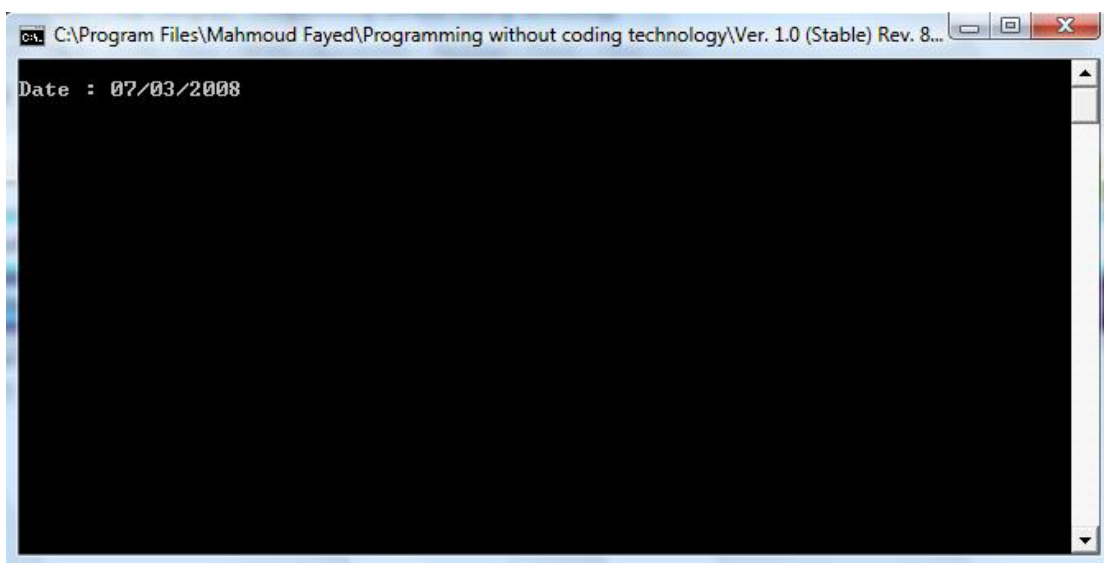
Interaction Page



Interaction Page



Final Steps Tree



The Final Application

ASCII Code

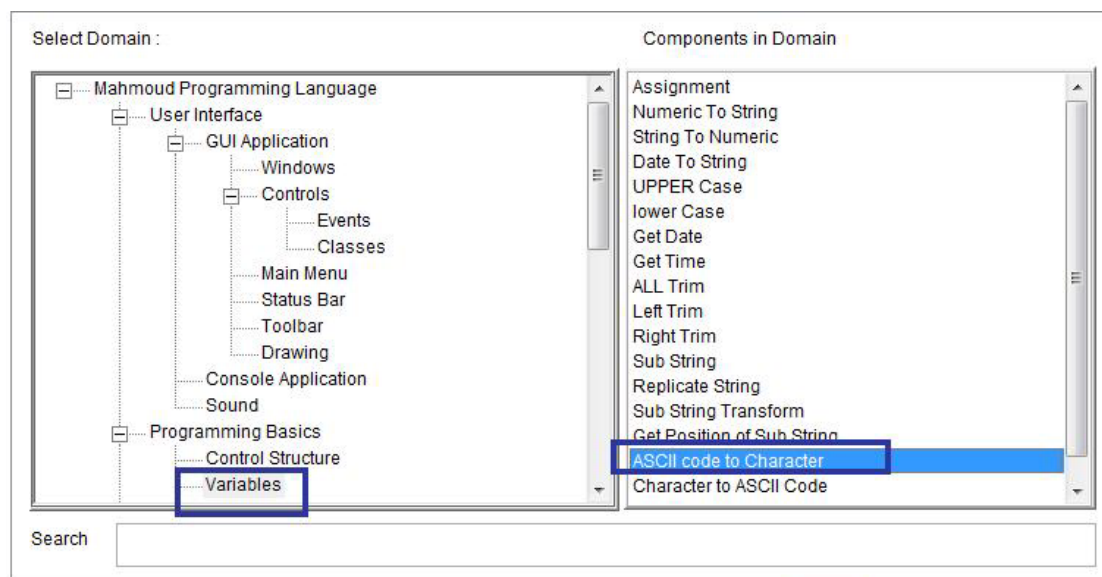
Components

- ASCII code to character
- Character to ASCII code

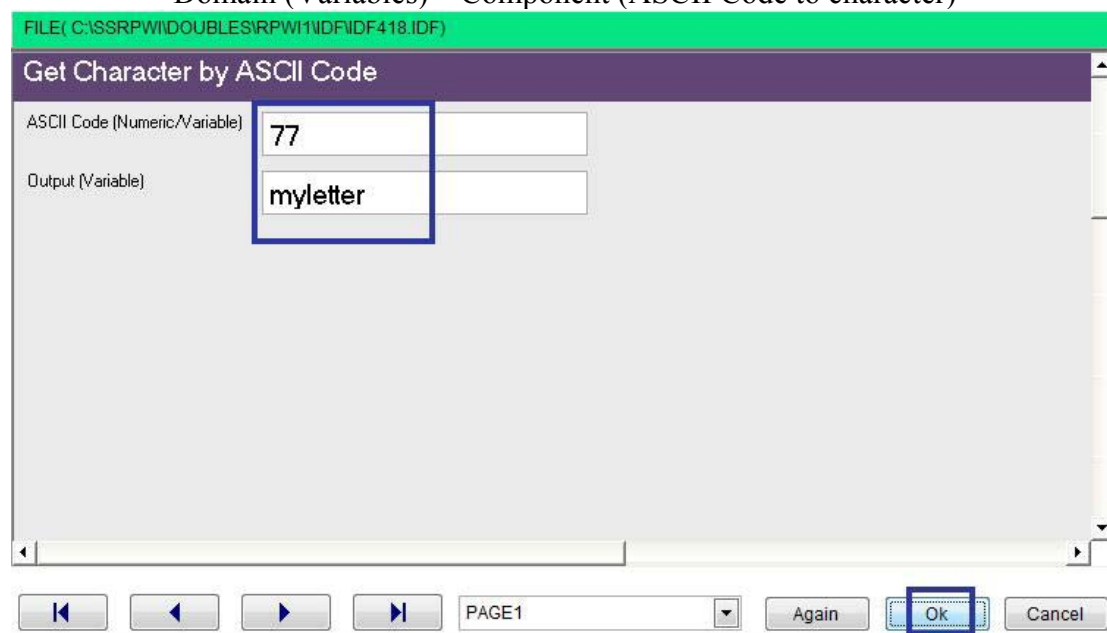
ASCII Code to character

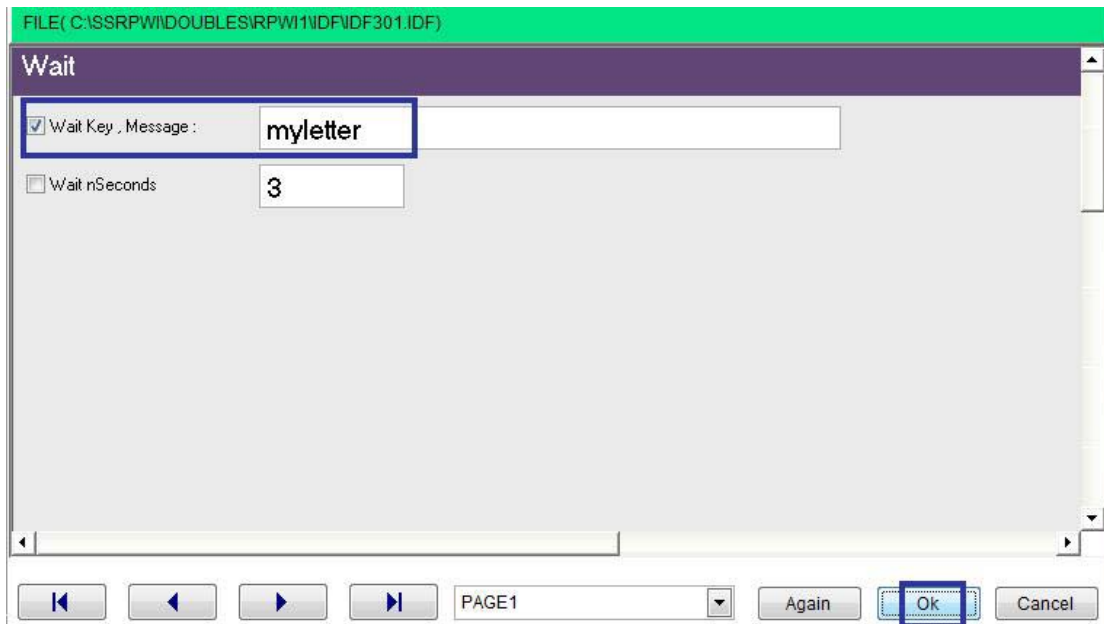
- Domain (Variables)
- Component (ASCII Code to character)

Example - Screen shots:-



Domain (Variables) – Component (ASCII Code to character)

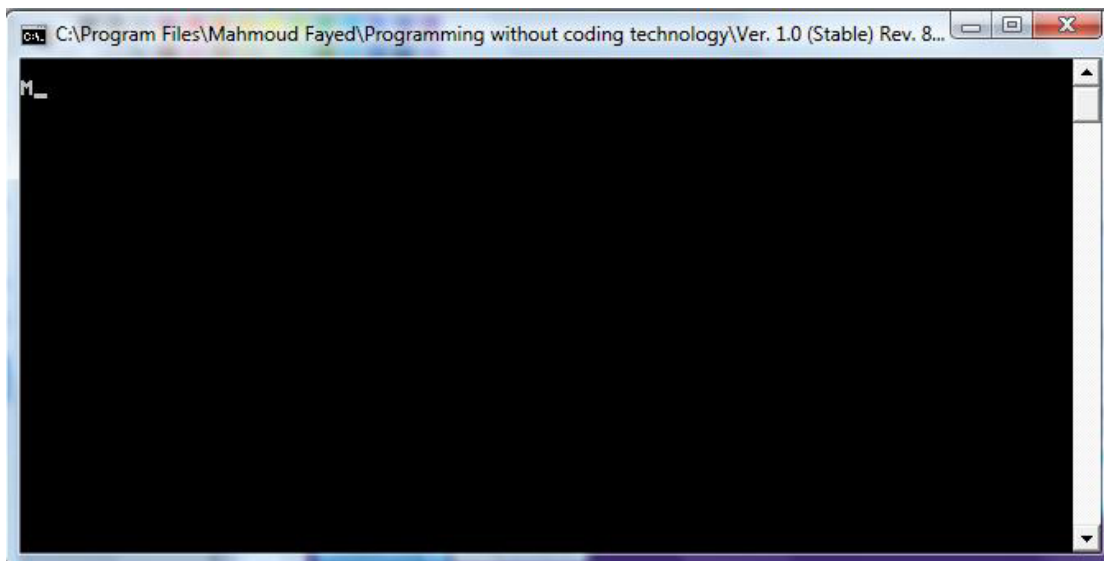




Interaction Page



Steps Tree

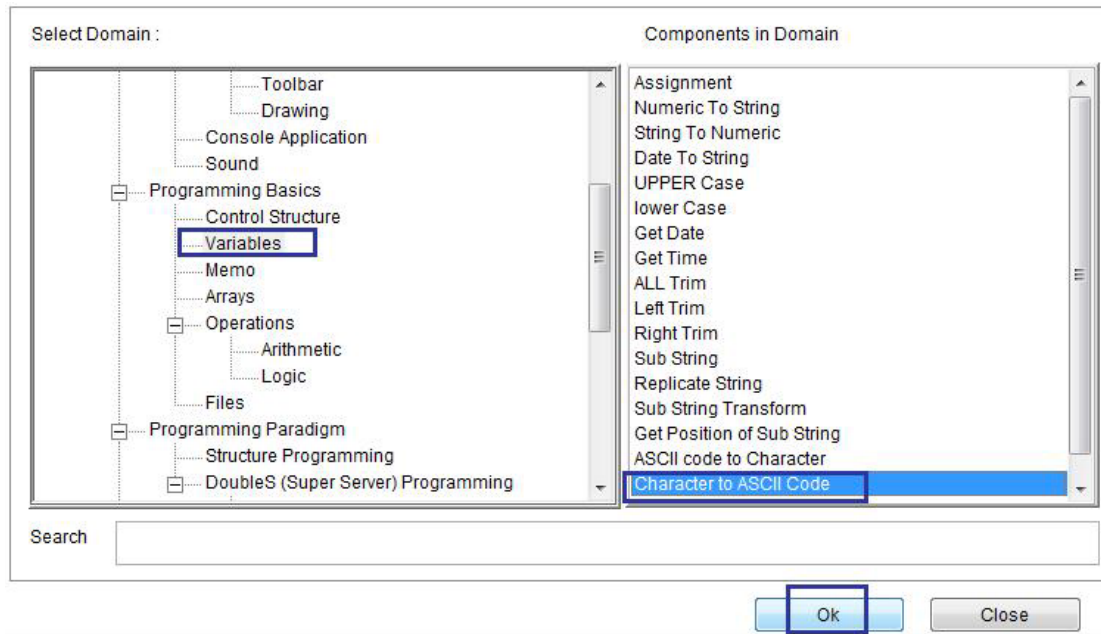


The final application

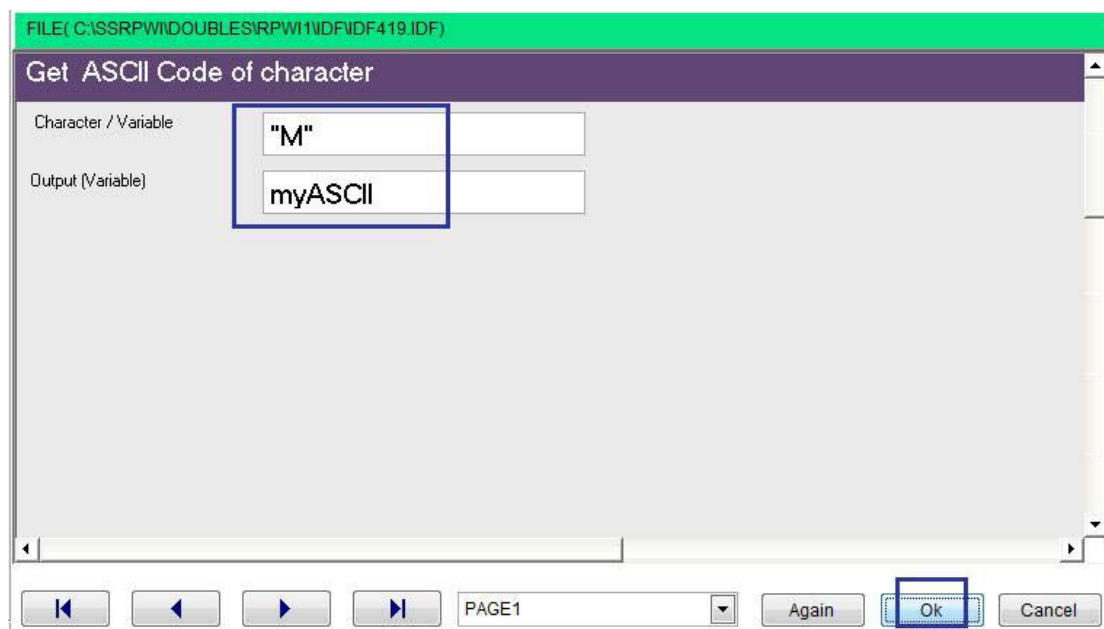
Character to ASCII Code

- Domain (Variables)
- Component (ASCII Code to character)

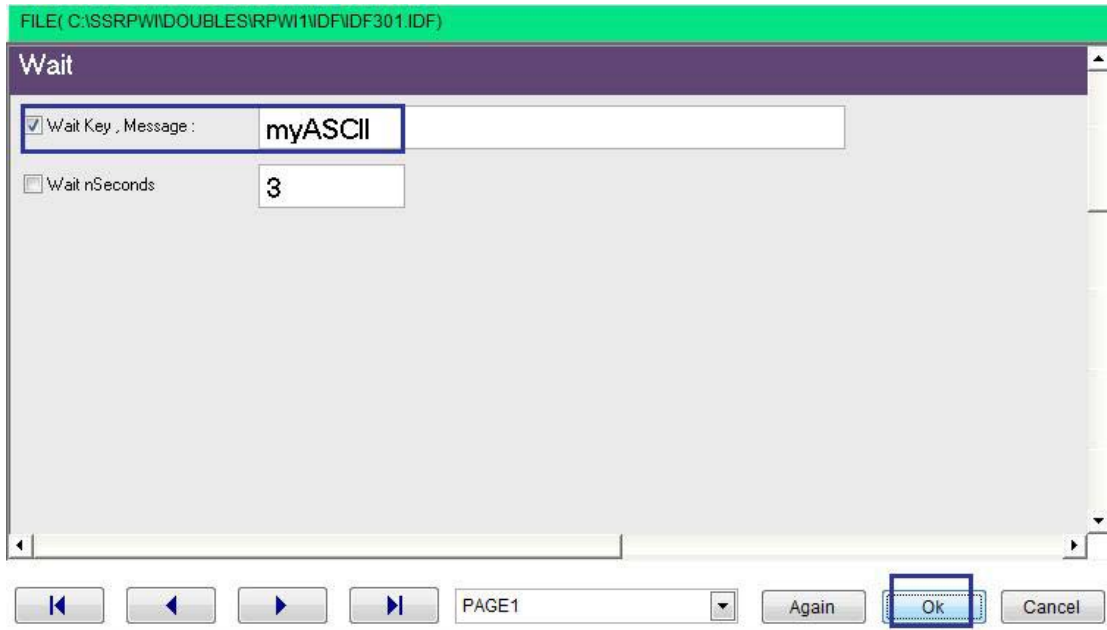
Example - Screen shots:-



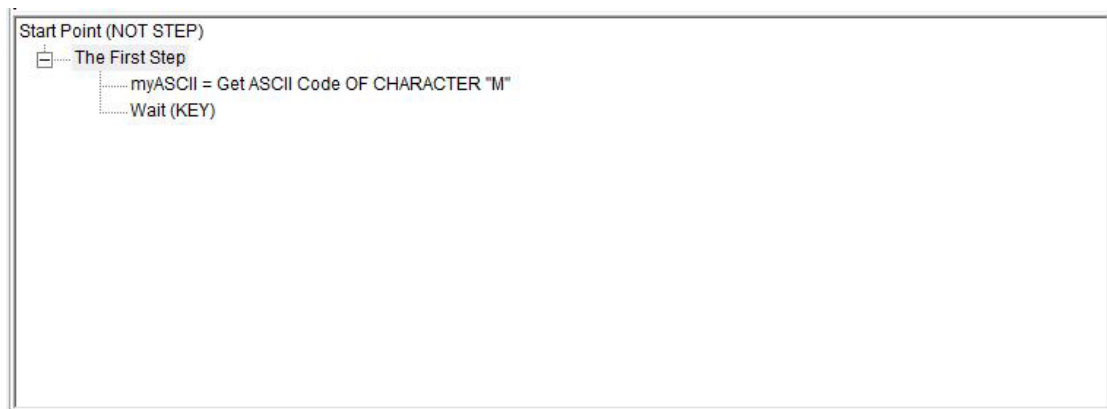
Domain (Variables) – Component (Character to ASCII Code)



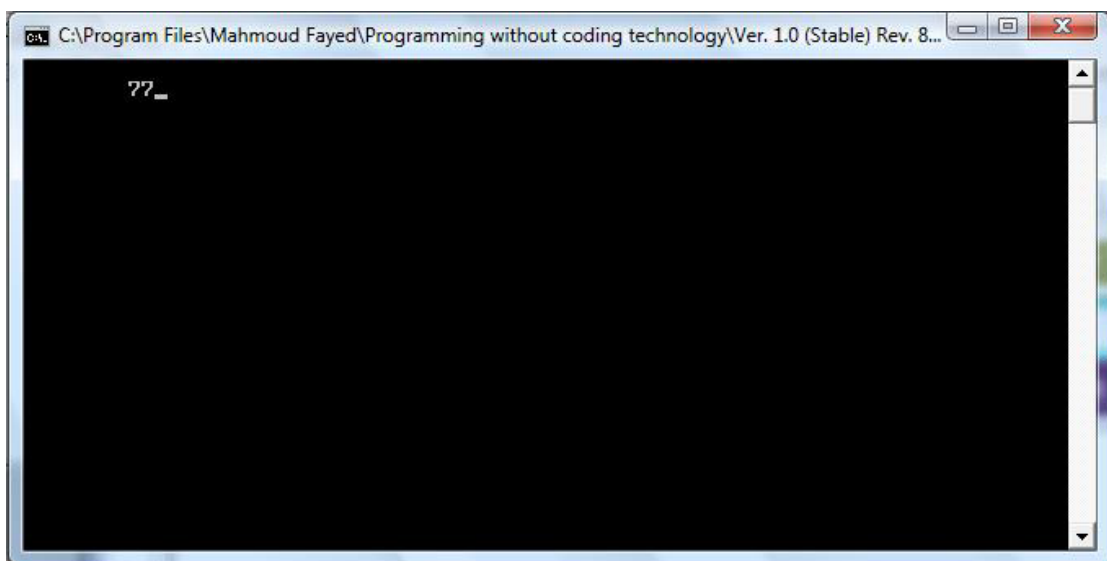
Interaction Page



Interaction Page



Steps Tree

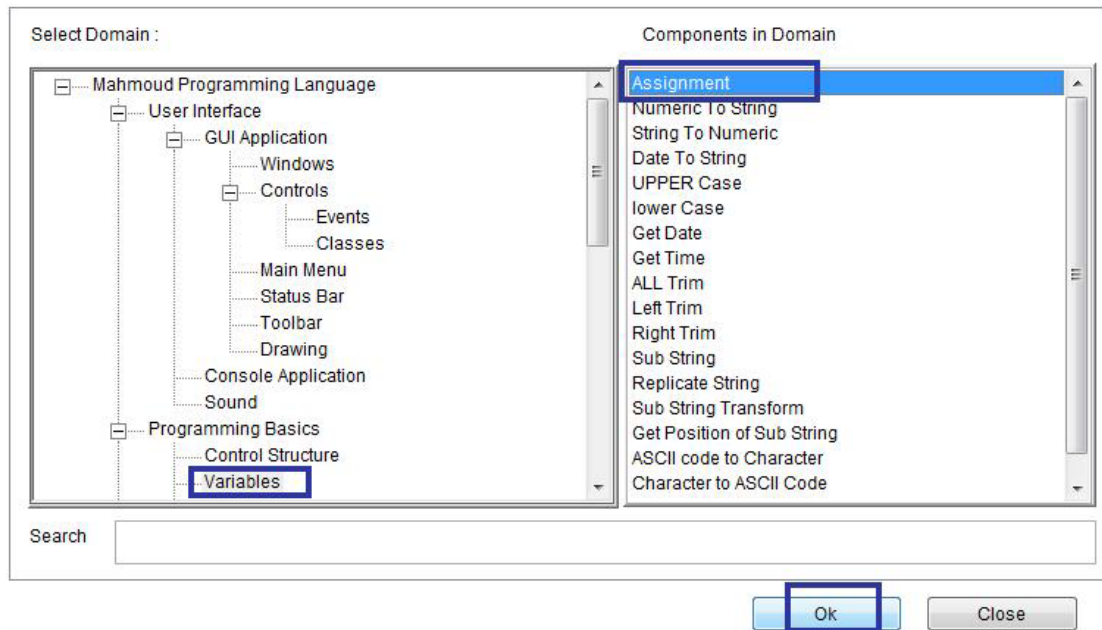


Final Application

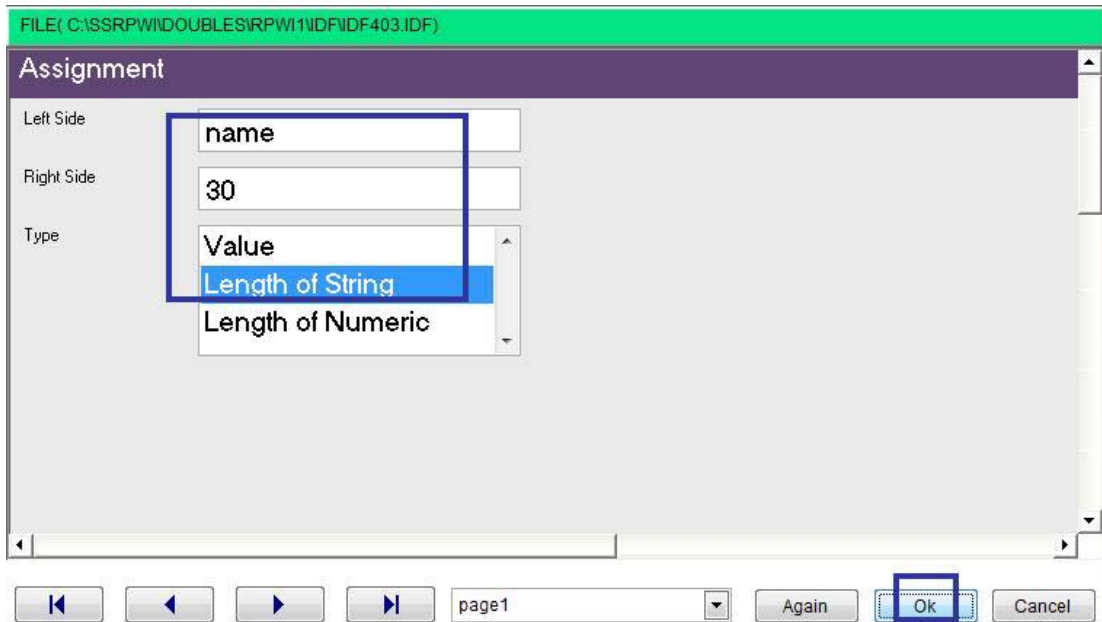
Getting Input from User

- Domain (Console Application)
- Component (Basic Input/Output)

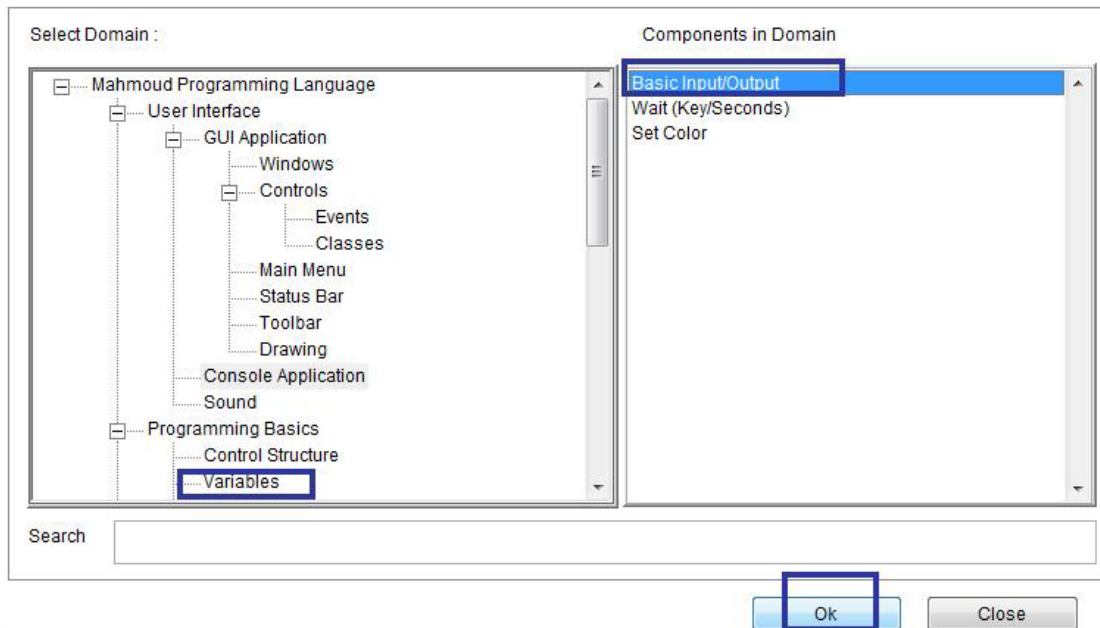
Example - Screen shots:-



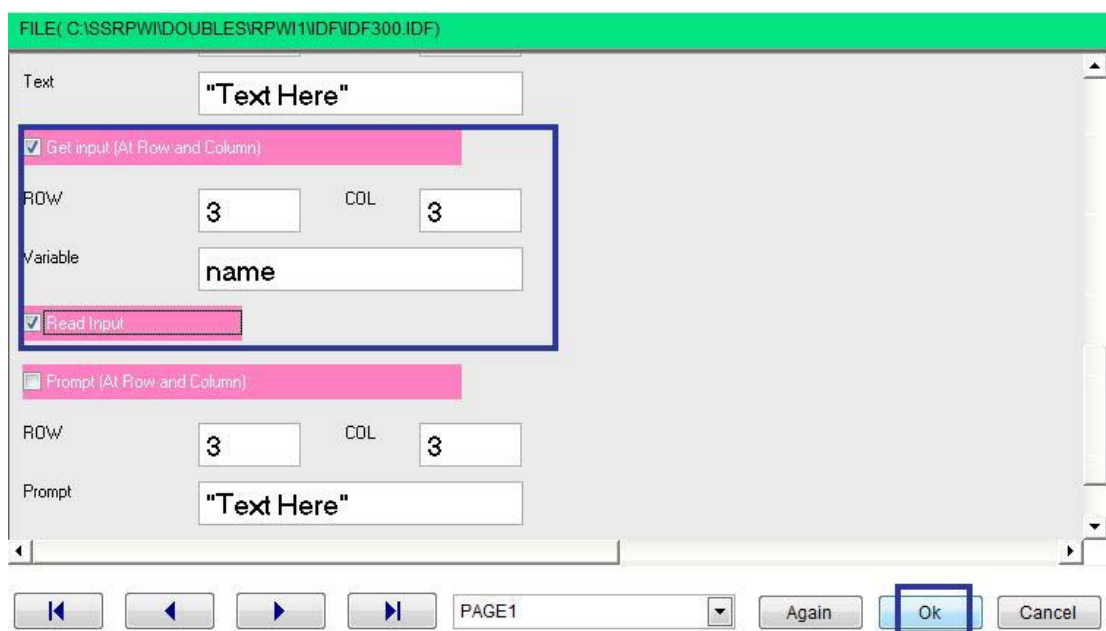
Domain (Variables) – Component (Assignment)



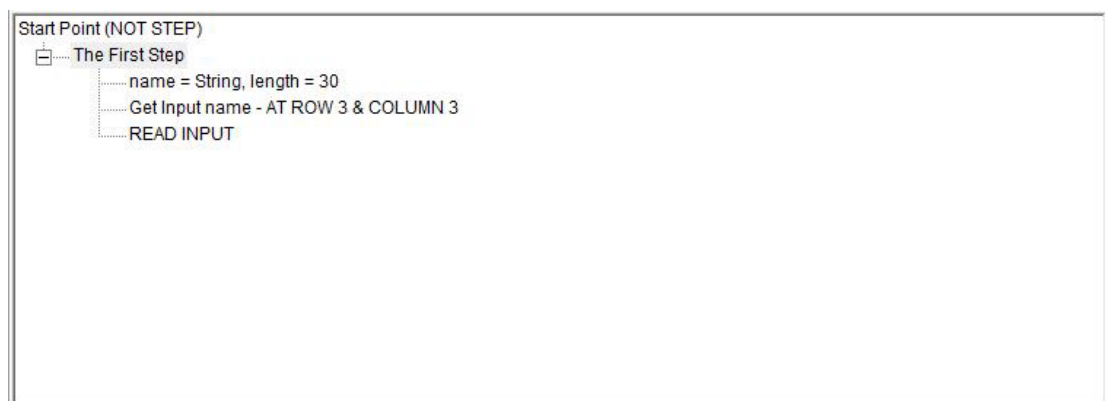
Interaction Page



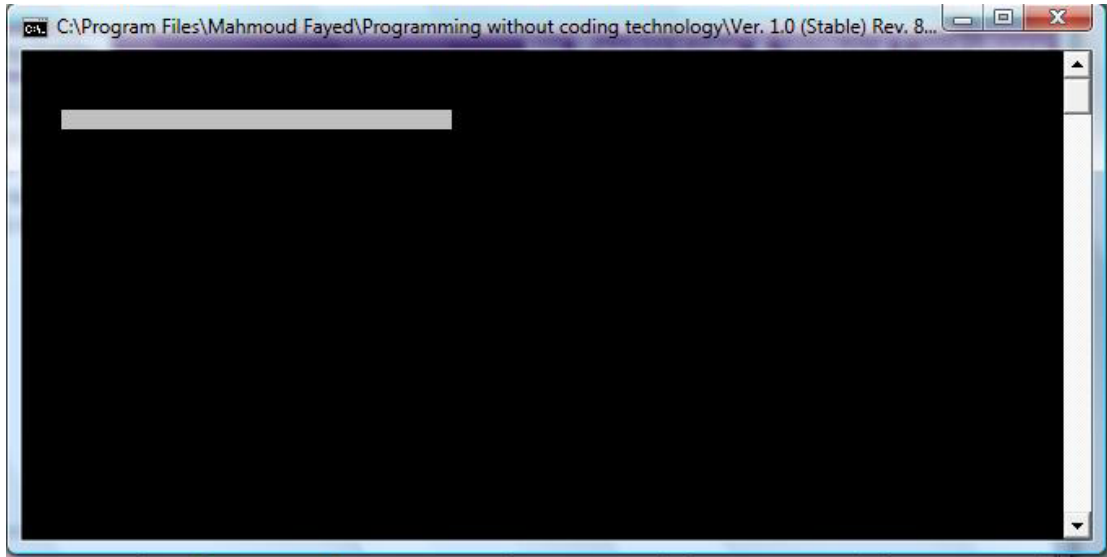
Domain (Variables) – Component (Basic Input/Output)



Interaction Page



Steps Tree

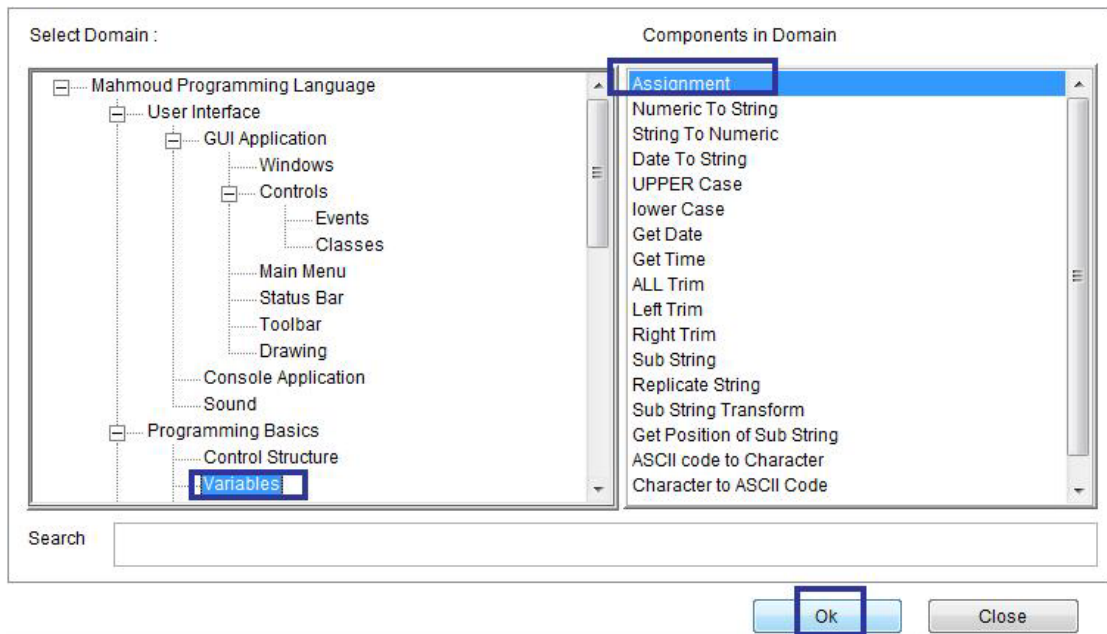


The final application

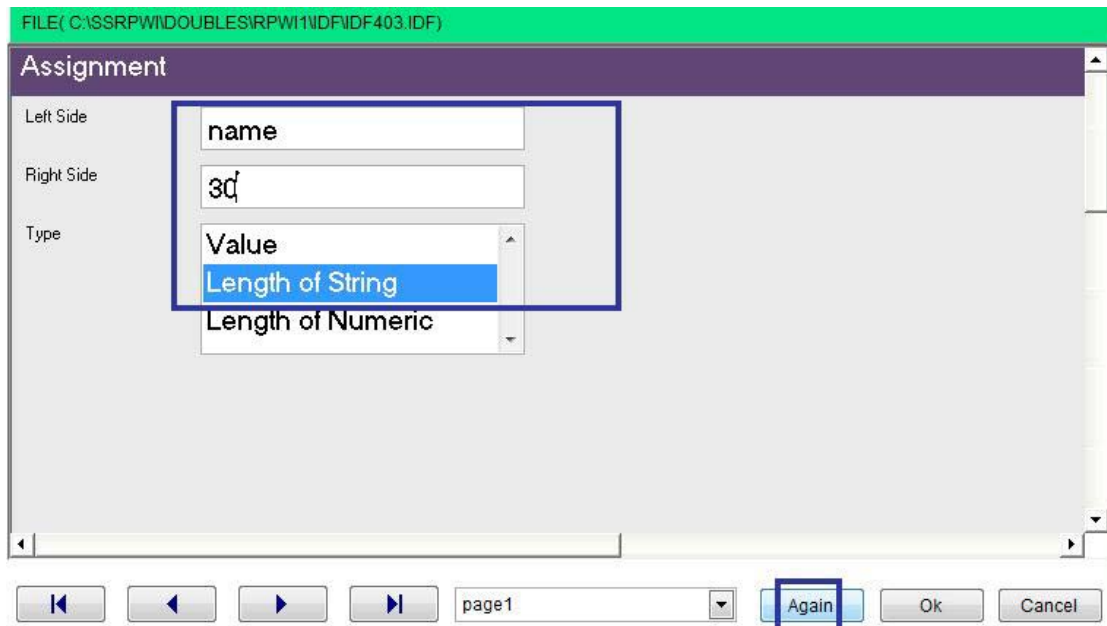
Note:

You can use more than one input line in the same time
And the user uses arrows to move between lines
Also you need to check (Read Input) just one time after input lines

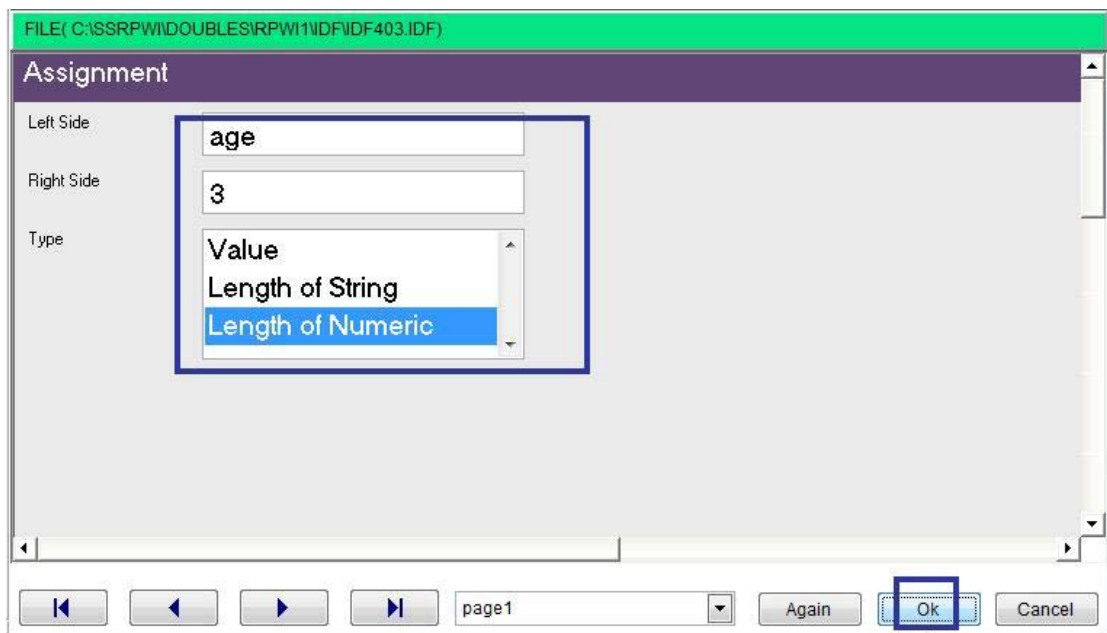
Example - Screen shots:-



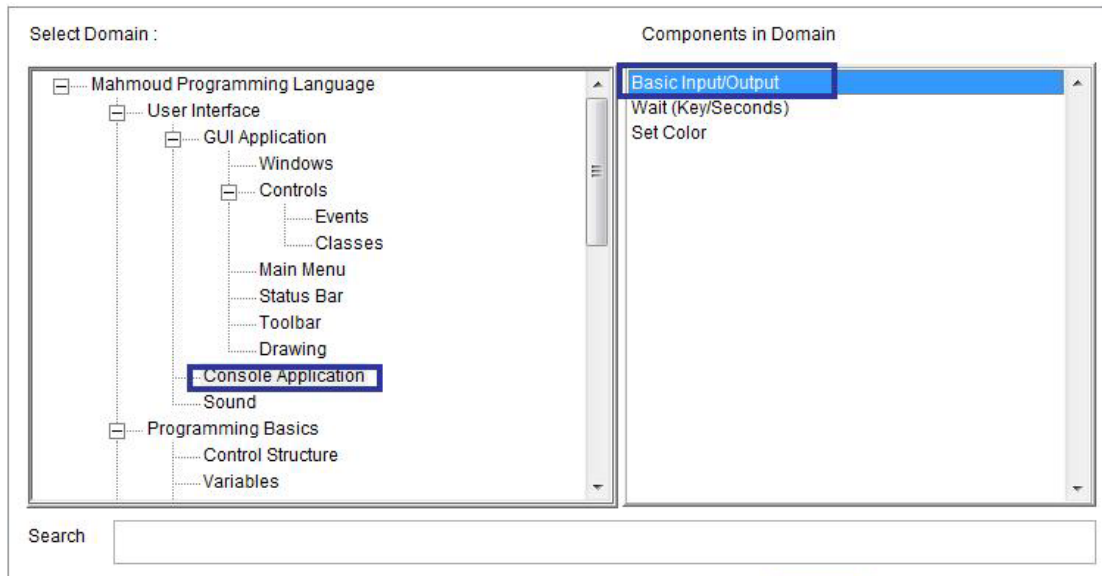
Domain (Variables) – Component (Assionment)



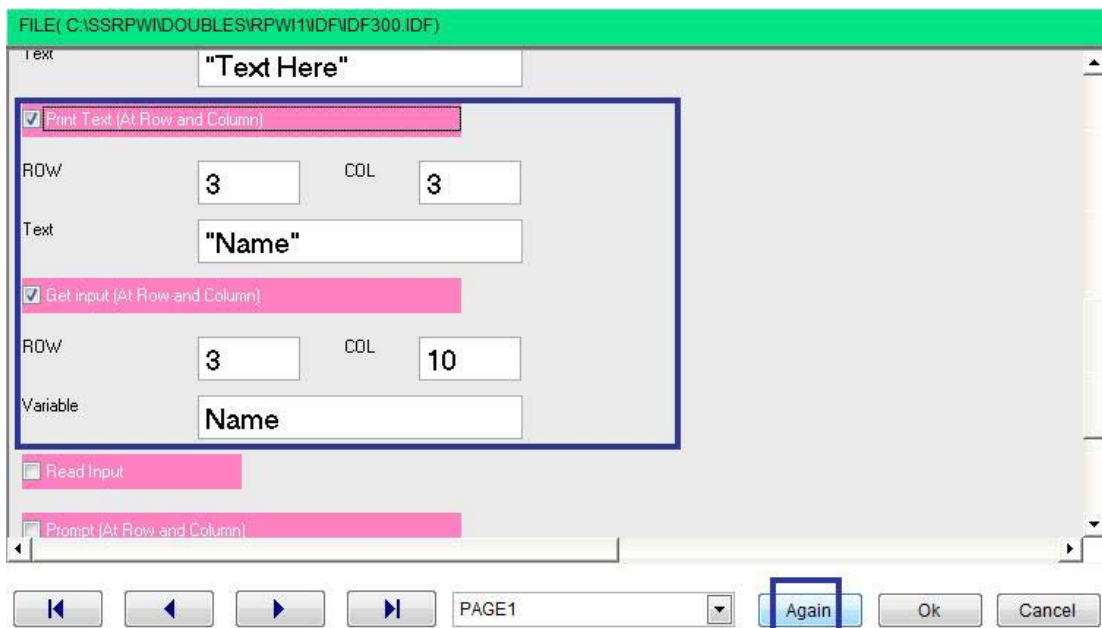
Interaction Page



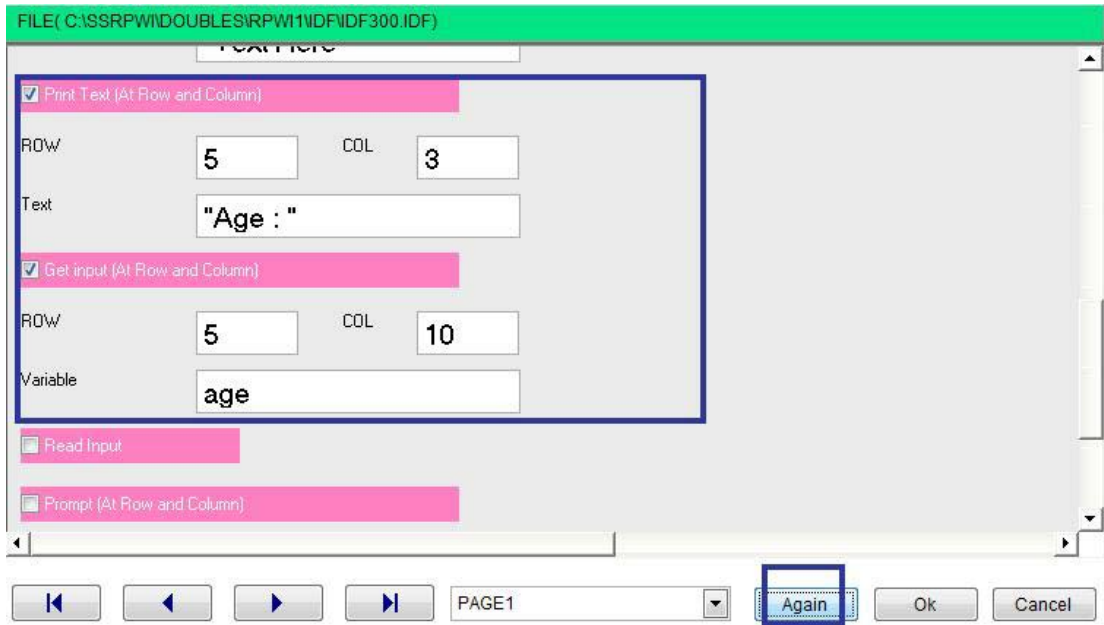
Interaction Page



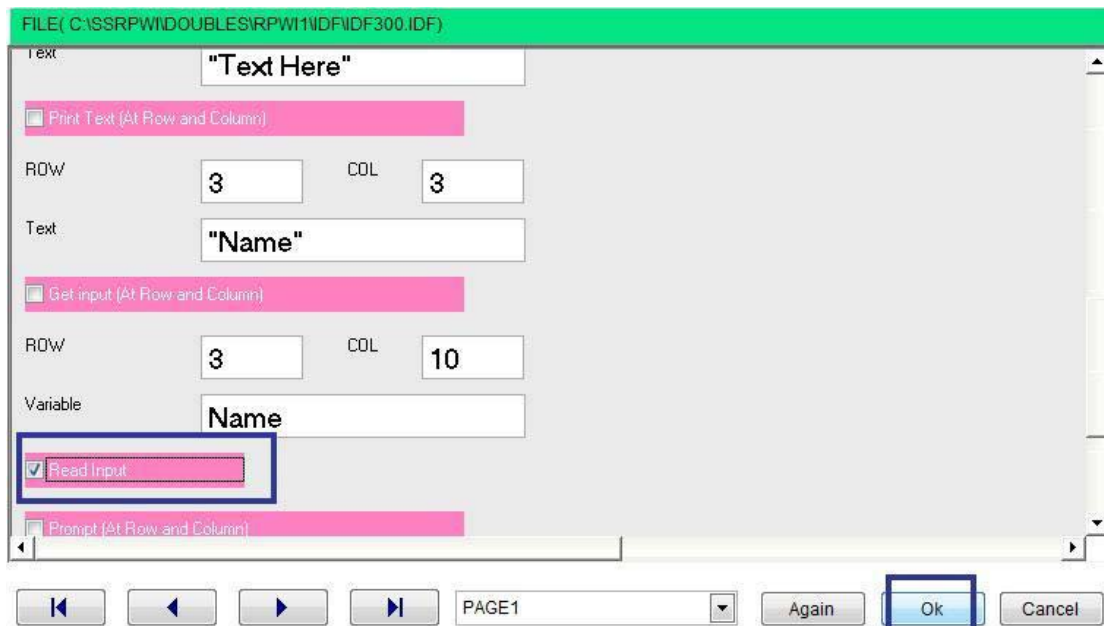
Domain (Console Application) – Component (Basic Input/Output)



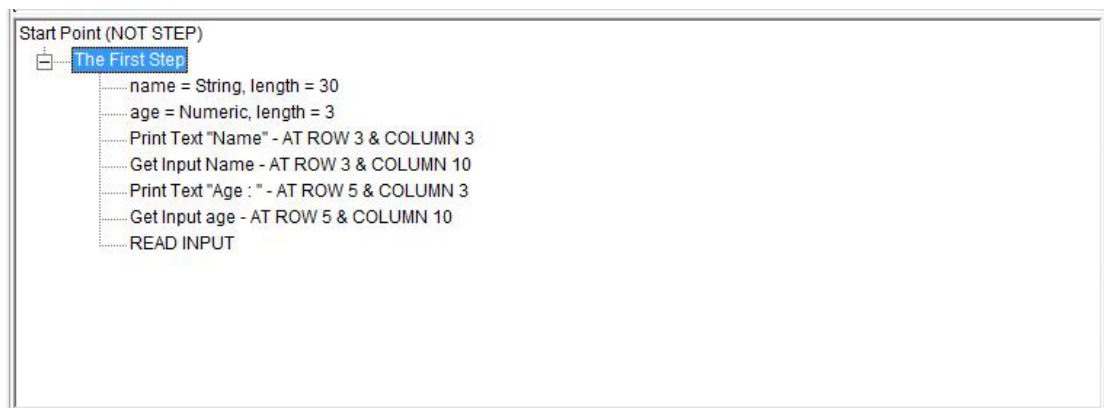
Interaction Page



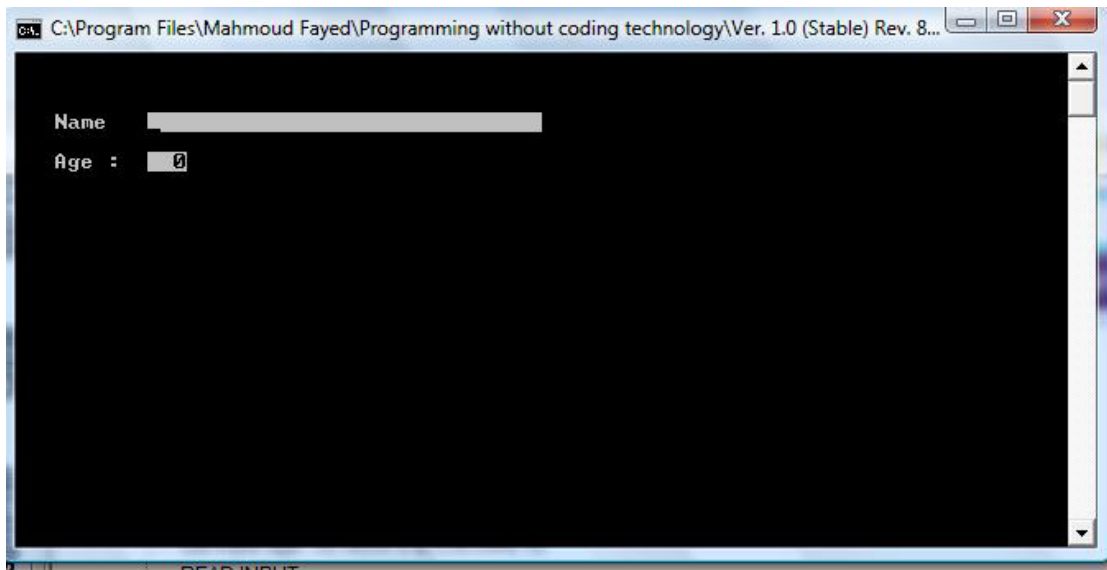
Interaction Page



Interaction Page



Steps Tree

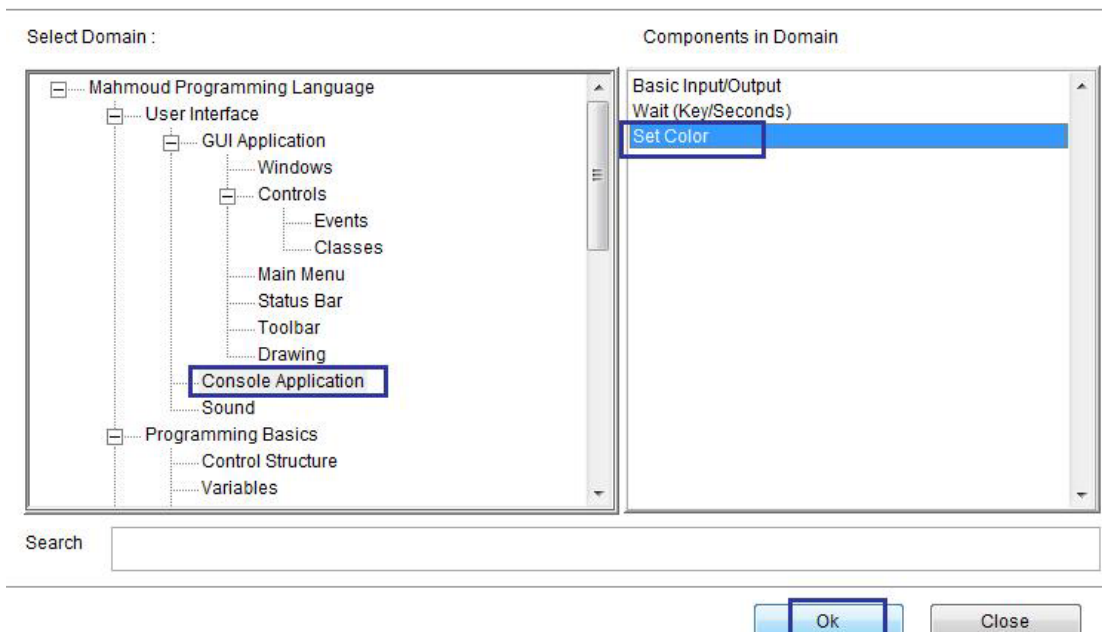


The final application

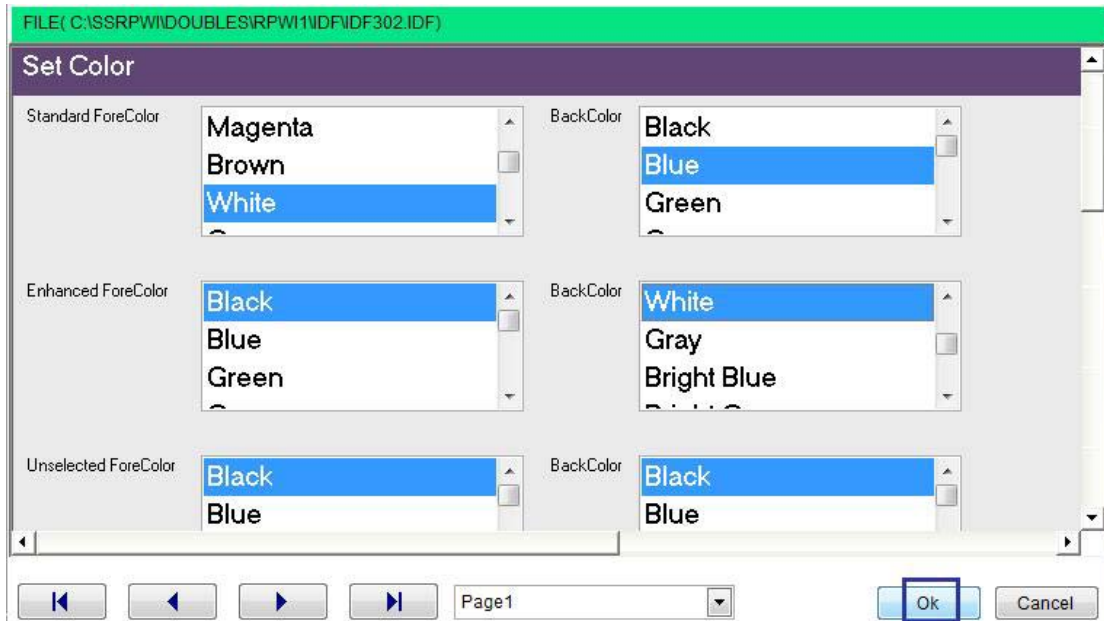
Menus

- Domain (Console Application)
- Component (Basic Input/output)

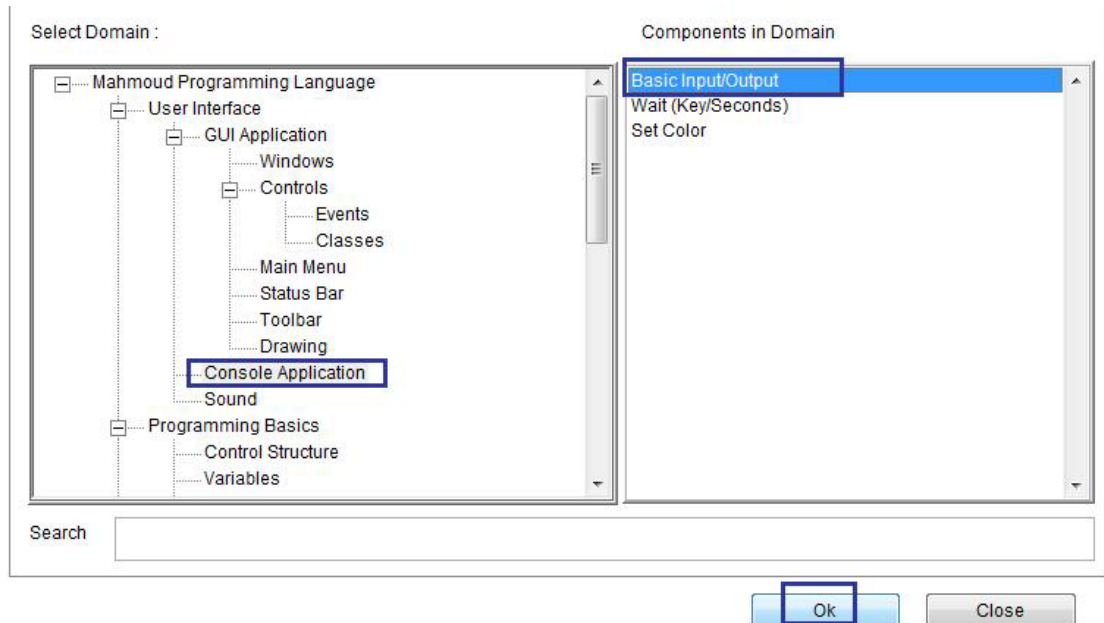
Example - Screen shots:-



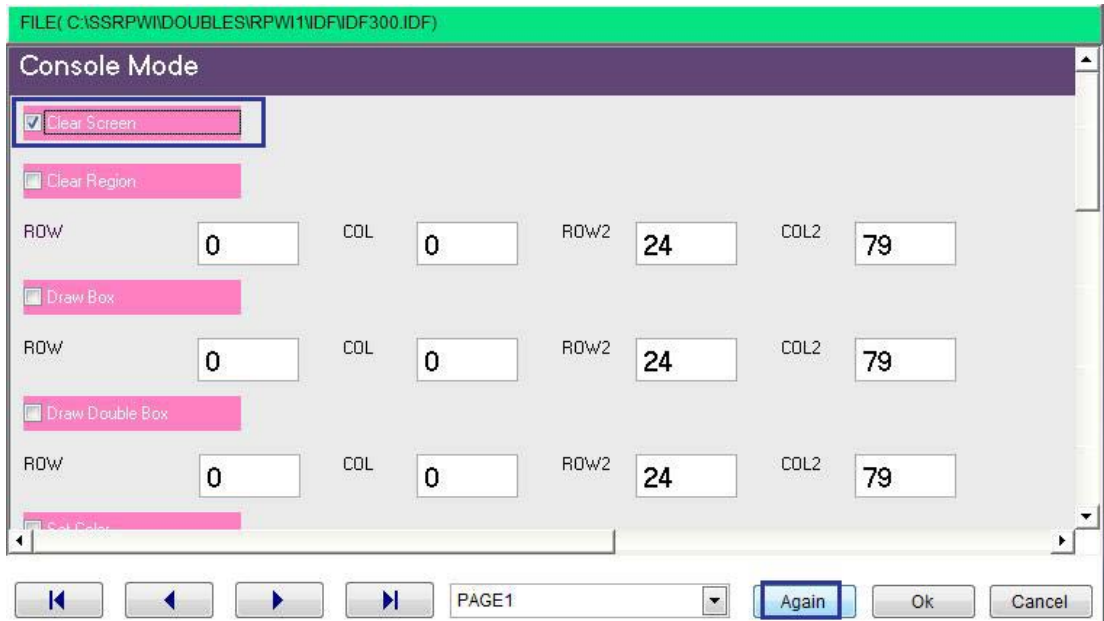
Domain (Console Application) – Component (Set color)



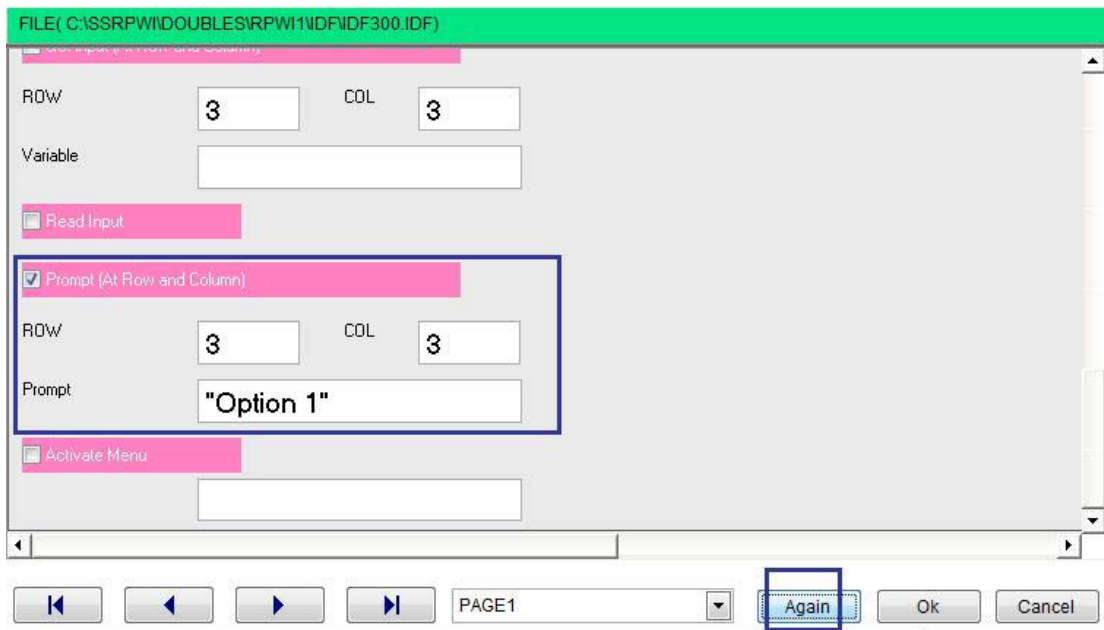
Interaction Page



Domain (Console Application) – Component (Basic Input/Output)



Interaction Page



Interaction Page

FILE(C:\SSRPW\DOUBLES\RPWI1\DF\DF300.IDF)

Read Input (At Row and Column)

ROW COL

Variable

Read Input

Prompt (At Row and Column)

ROW COL

Prompt

Activate Menu

Interaction Page

FILE(C:\SSRPW\DOUBLES\RPWI1\DF\DF300.IDF)

Read Input (At Row and Column)

ROW COL

Variable

Read Input

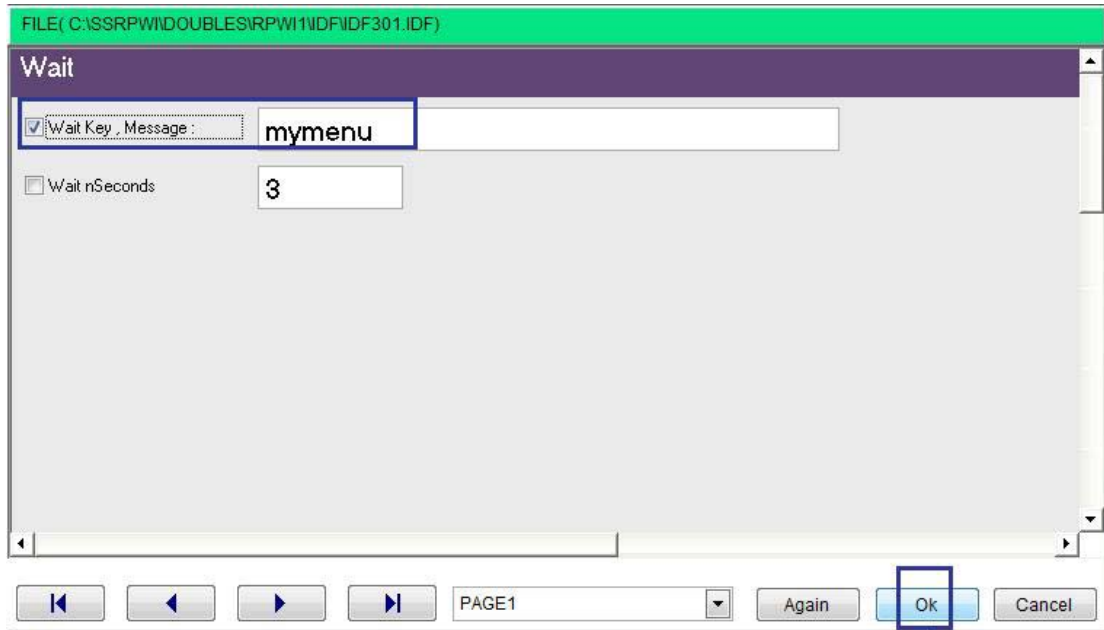
Prompt (At Row and Column)

ROW COL

Prompt

Activate Menu

Interaction Page



Interaction Page



Steps Tree



The Final Application

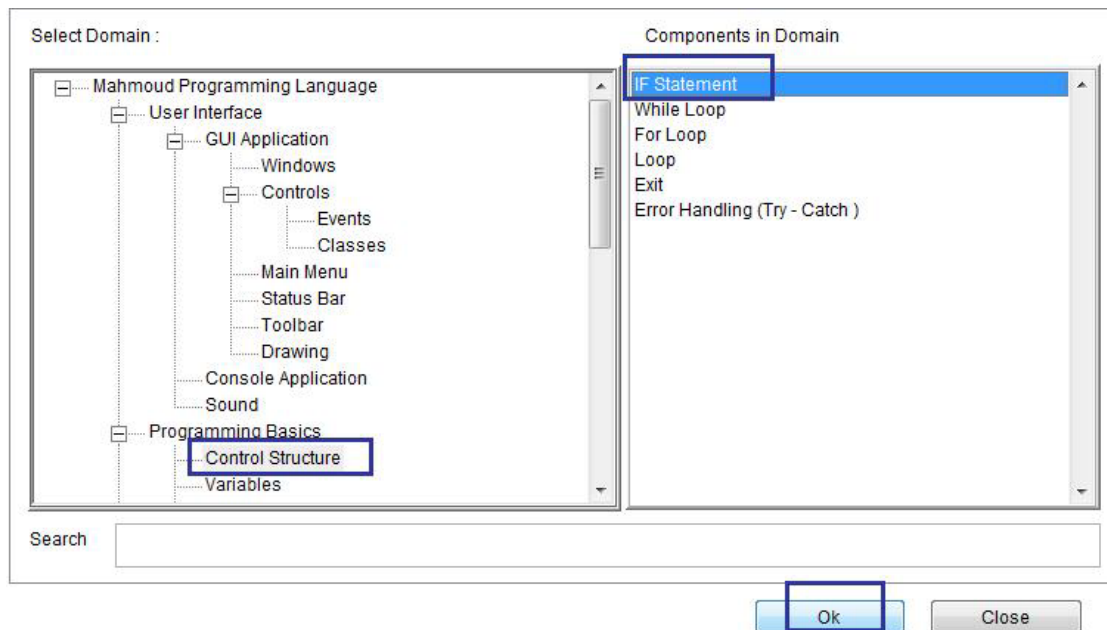


The Final Application

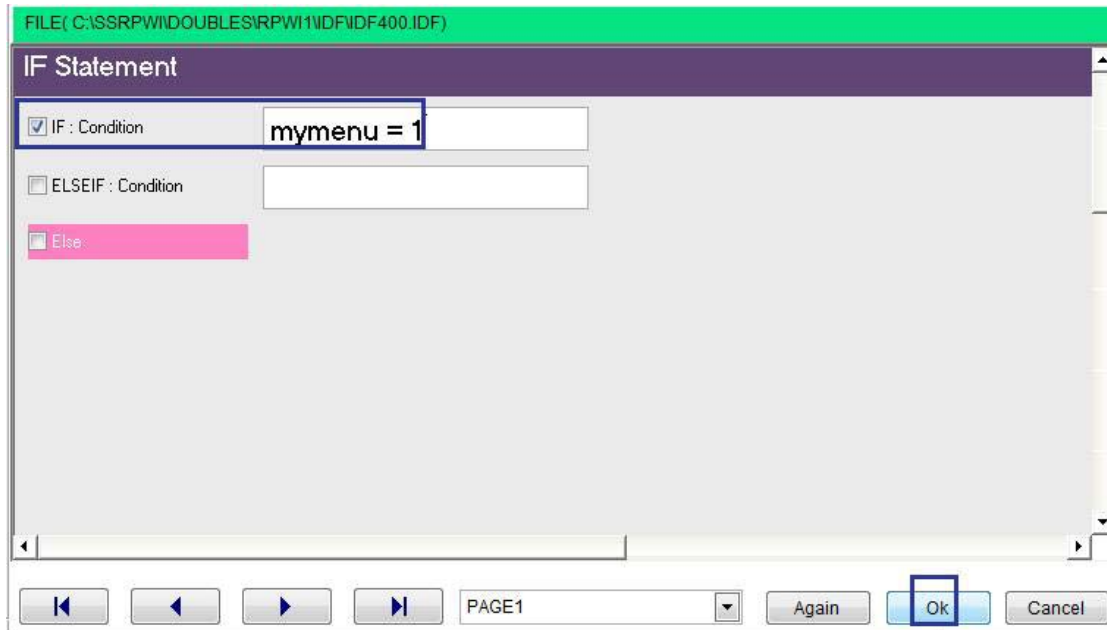
IF Statement

- Domain (Control Structure)
- Component (IF Statement)

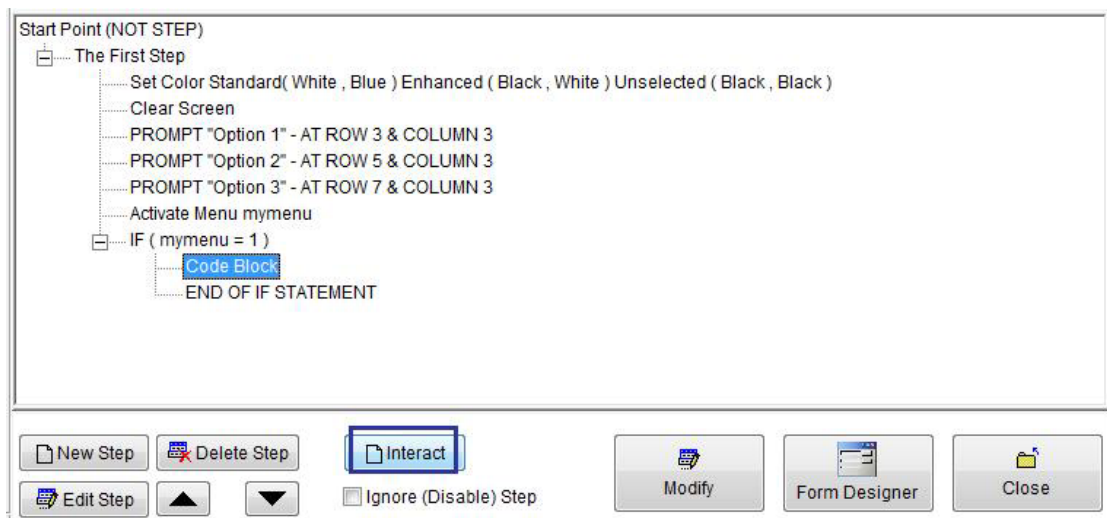
Example - Screen shots:-



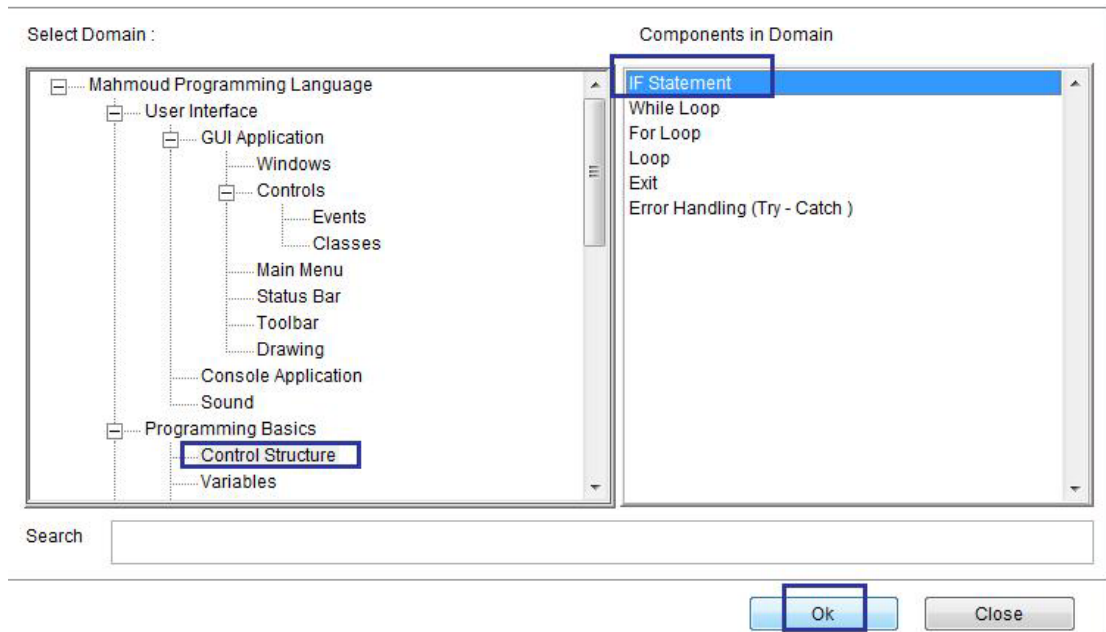
Domain (Control Structure) Component (IF Statement)



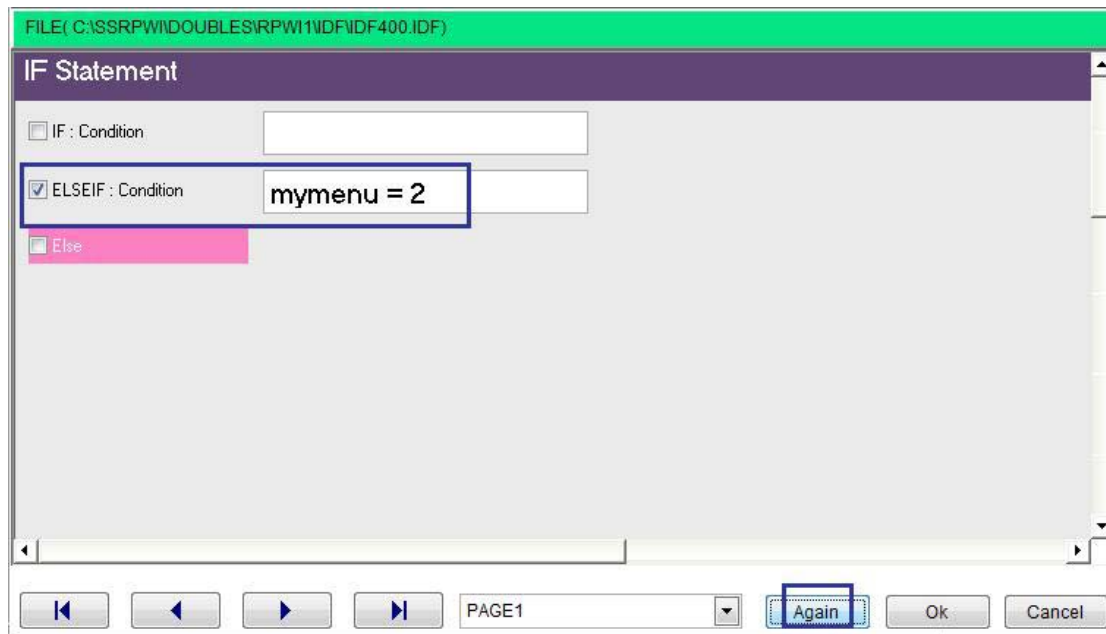
Interaction Page



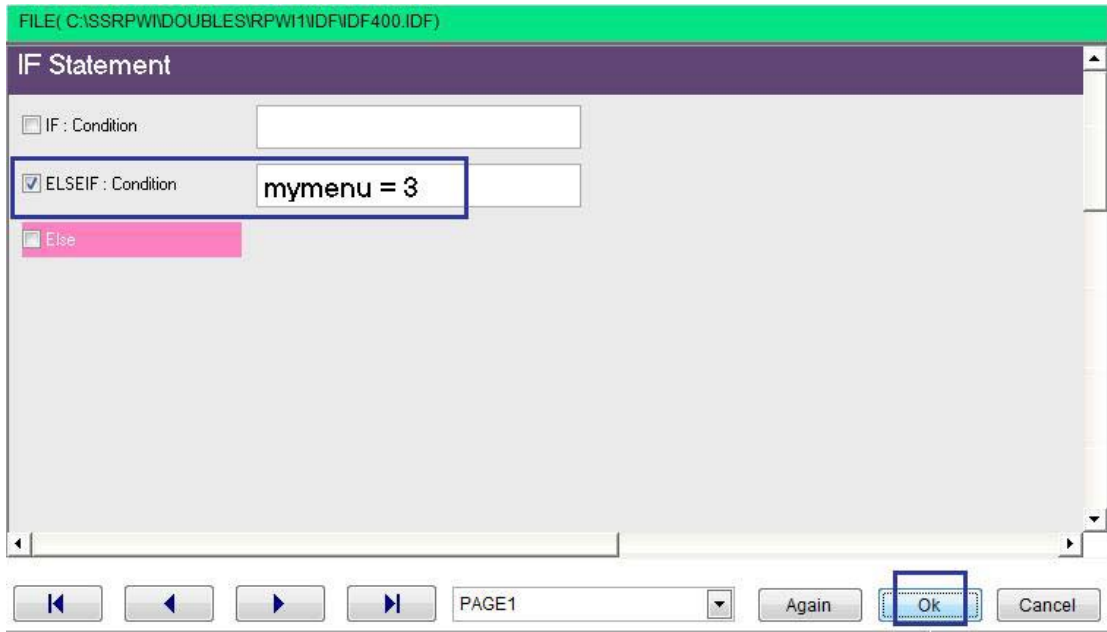
Steps Tree



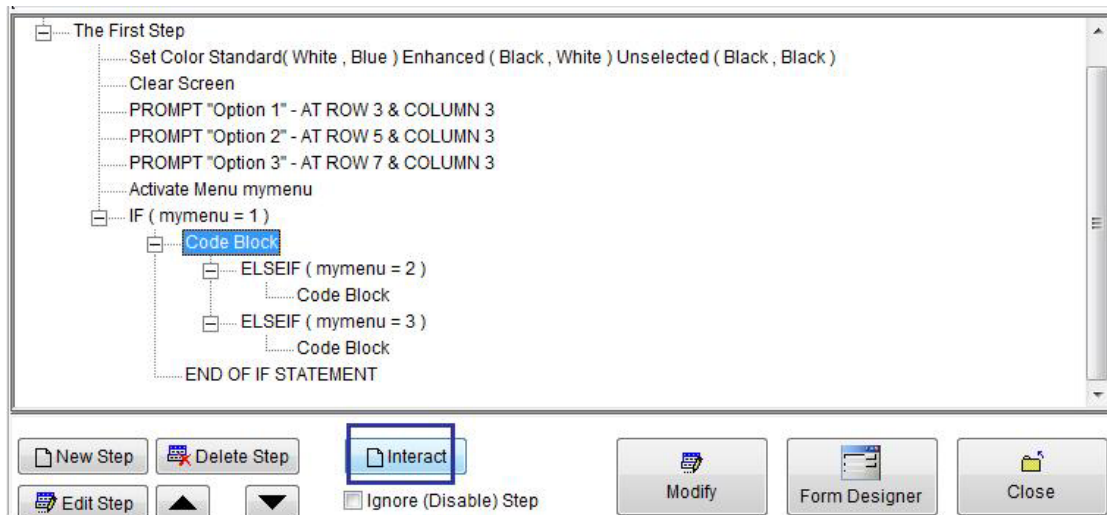
Steps Tree



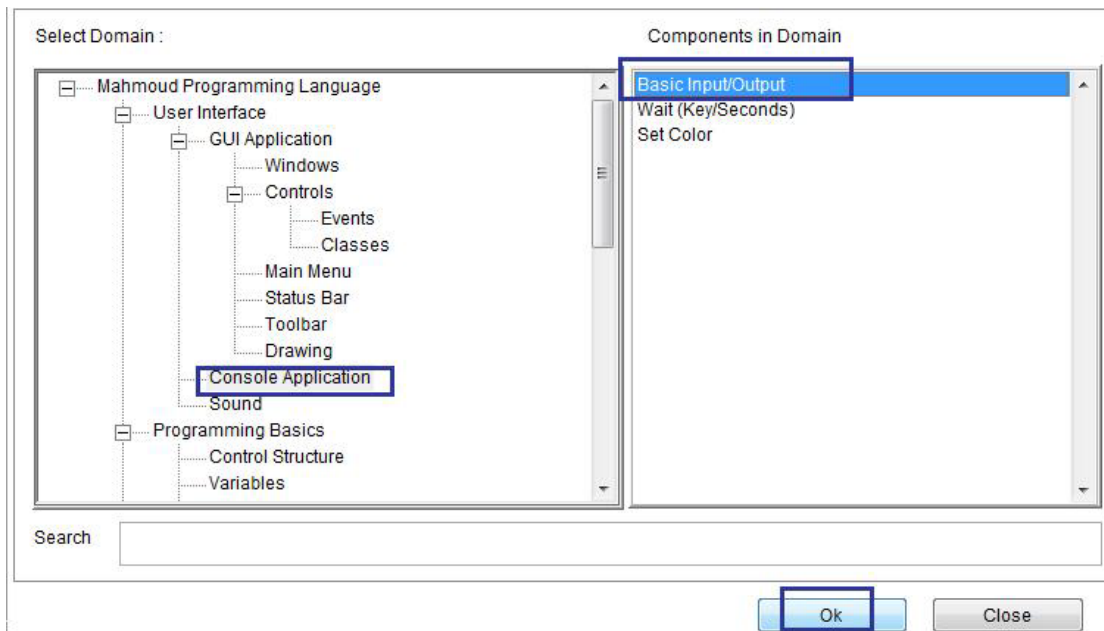
Interaction Page



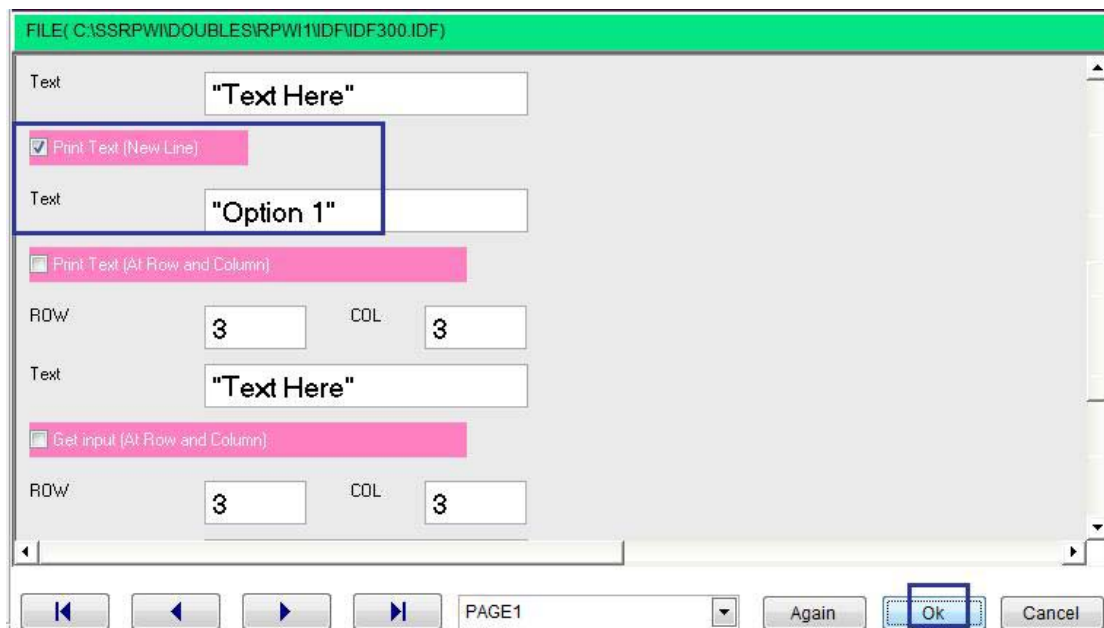
Interaction Page



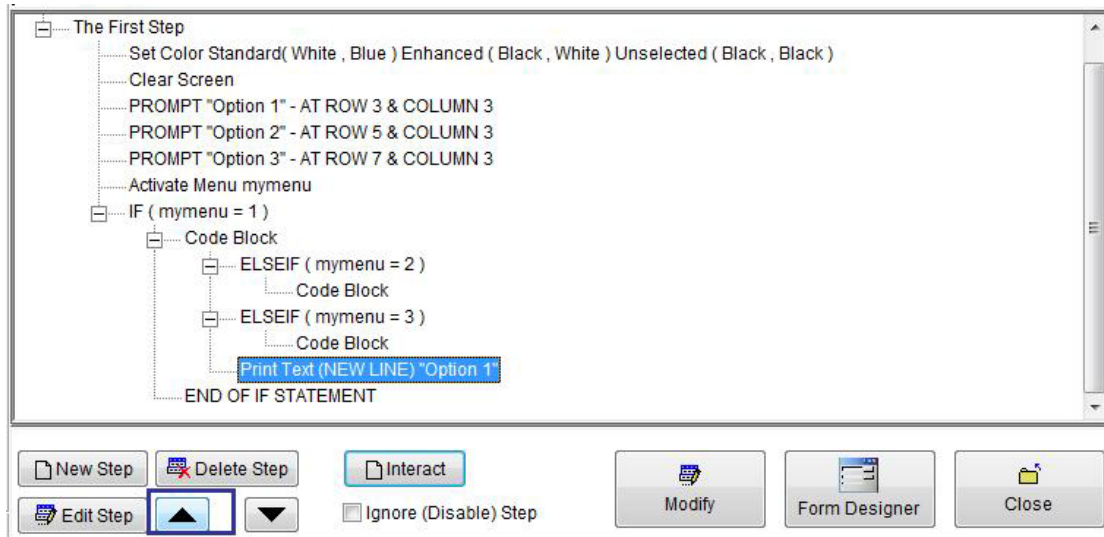
Steps Tree



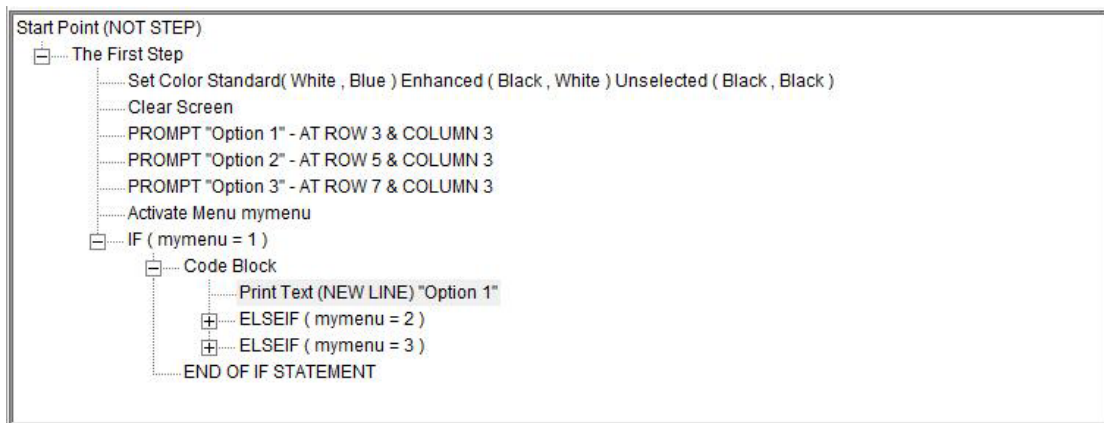
Steps Tree



Interaction Page



Steps Tree



Steps Tree

Edit Data Code Veto RPWI Transporter Package Tools Help

Goal : Main Copy **Goal Designer**

Start Point (NOT STEP)\The First Step\IF (mymenu = 1)\Code Block\Print Text (NEW LINE) "Option 1"

Steps Tree Step Details **No Restrictions & Without Limitations**

Start Point (NOT STEP)

- [-] The First Step
 - Set Color Standard(White , Blue) Enhanced (Black , White) Unselected (Black , Black)
 - Clear Screen
 - PROMPT "Option 1" - AT ROW 3 & COLUMN 3
 - PROMPT "Option 2" - AT ROW 5 & COLUMN 3
 - PROMPT "Option 3" - AT ROW 7 & COLUMN 3
 - Activate Menu mymenu
 - [-] IF (mymenu = 1)
 - [-] Code Block
 - Print Text (NEW LINE) "Option 1"**
 - [-] ELSEIF (mymenu = 2)
 - Code Block
 - [-] ELSEIF (mymenu = 3)
 - Code Block
 - END OF IF STATEMENT

Steps Tree

Edit Data Code Veto RPWI Transporter Package Tools Help

Goal : Main Paste **Goal Designer**

Start Point (NOT STEP)\The First Step\IF (mymenu = 1)\Code Block\ELSEIF (mymenu = 2)\Code Block

Steps Tree Step Details **No Restrictions & Without Limitations**

Start Point (NOT STEP)

- [-] The First Step
 - Set Color Standard(White , Blue) Enhanced (Black , White) Unselected (Black , Black)
 - Clear Screen
 - PROMPT "Option 1" - AT ROW 3 & COLUMN 3
 - PROMPT "Option 2" - AT ROW 5 & COLUMN 3
 - PROMPT "Option 3" - AT ROW 7 & COLUMN 3
 - Activate Menu mymenu
 - [-] IF (mymenu = 1)
 - [-] Code Block
 - Print Text (NEW LINE) "Option 1"
 - [-] ELSEIF (mymenu = 2)
 - Code Block**
 - [-] ELSEIF (mymenu = 3)
 - Code Block
 - END OF IF STATEMENT

Steps Tree

Goal : Main **Goal Designer**

Start Point (NOT STEP)The First StepIF (mymenu = 1)Code BlockELSEIF (mymenu = 3)Code Block

Steps Tree Step Details **No Restrictions & Without Limitations**

```

Start Point (NOT STEP)
├── The First Step
│   ├── Set Color Standard( White , Blue ) Enhanced ( Black , White ) Unselected ( Black , Black )
│   ├── Clear Screen
│   ├── PROMPT "Option 1" - AT ROW 3 & COLUMN 3
│   ├── PROMPT "Option 2" - AT ROW 5 & COLUMN 3
│   ├── PROMPT "Option 3" - AT ROW 7 & COLUMN 3
│   ├── Activate Menu mymenu
│   └── IF ( mymenu = 1 )
│       ├── Code Block
│       │   ├── Print Text (NEW LINE) "Option 1"
│       │   ├── ELSEIF ( mymenu = 2 )
│       │   │   └── Code Block
│       │   │       └── Print Text (NEW LINE) "Option 1"
│       │   └── ELSEIF ( mymenu = 3 )
│       │       └── Code Block
│       └── END OF IF STATEMENT
    
```

Steps Tree

```

Start Point (NOT STEP)
├── The First Step
│   ├── Set Color Standard( White , Blue ) Enhanced ( Black , White ) Unselected ( Black , Black )
│   ├── Clear Screen
│   ├── PROMPT "Option 1" - AT ROW 3 & COLUMN 3
│   ├── PROMPT "Option 2" - AT ROW 5 & COLUMN 3
│   ├── PROMPT "Option 3" - AT ROW 7 & COLUMN 3
│   ├── Activate Menu mymenu
│   └── IF ( mymenu = 1 )
│       ├── Code Block
│       │   ├── Print Text (NEW LINE) "Option 1"
│       │   ├── ELSEIF ( mymenu = 2 )
│       │   │   └── Code Block
│       │   │       └── Print Text (NEW LINE) "Option 1"
│       │   └── ELSEIF ( mymenu = 3 )
│       │       └── Code Block
│       │           └── Print Text (NEW LINE) "Option 1"
│       └── END OF IF STATEMENT
    
```

Ignore (Disable) Step

Steps Tree

FILE(C:\SSRPW\DOUBLES\RPW1\DF\DF300.IDF)

Color

Print Text

Text

Print Text (New Line)

Text

Print Text (At Row and Column)

ROW COL

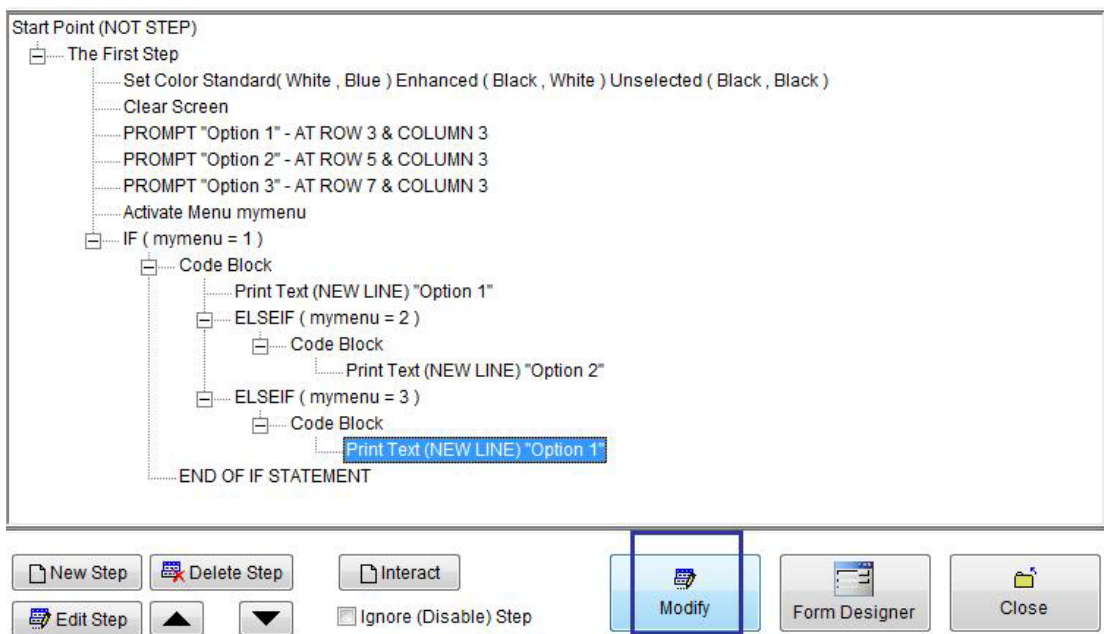
Text

Get input (At Row and Column)

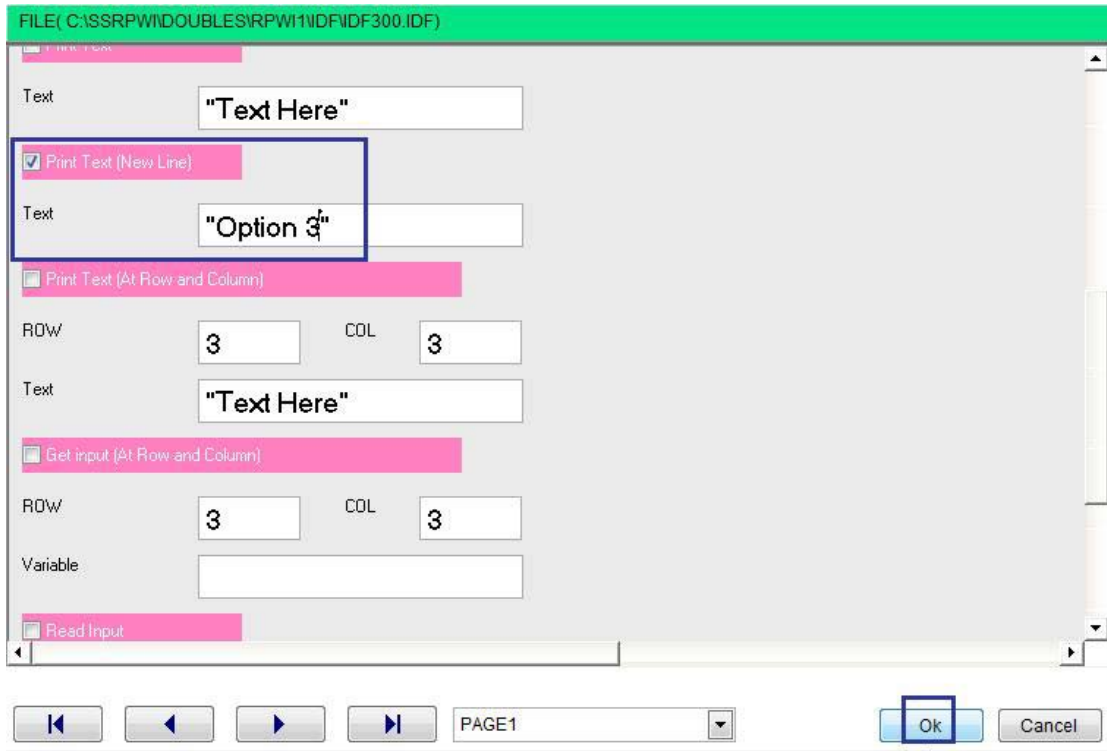
ROW COL

Navigation: [Back] [Left] [Right] [Forward] PAGE1 [Ok] [Cancel]

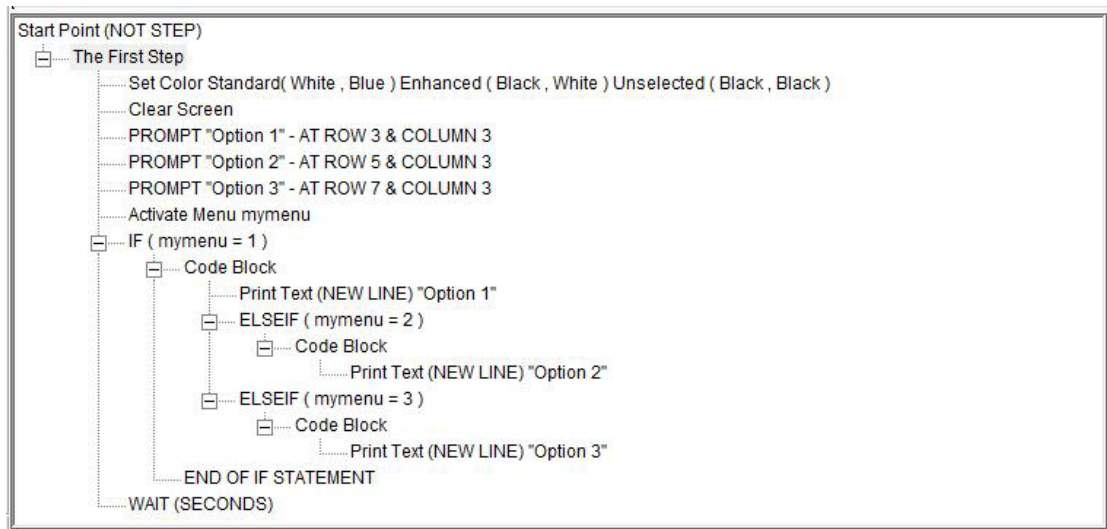
Interaction Page



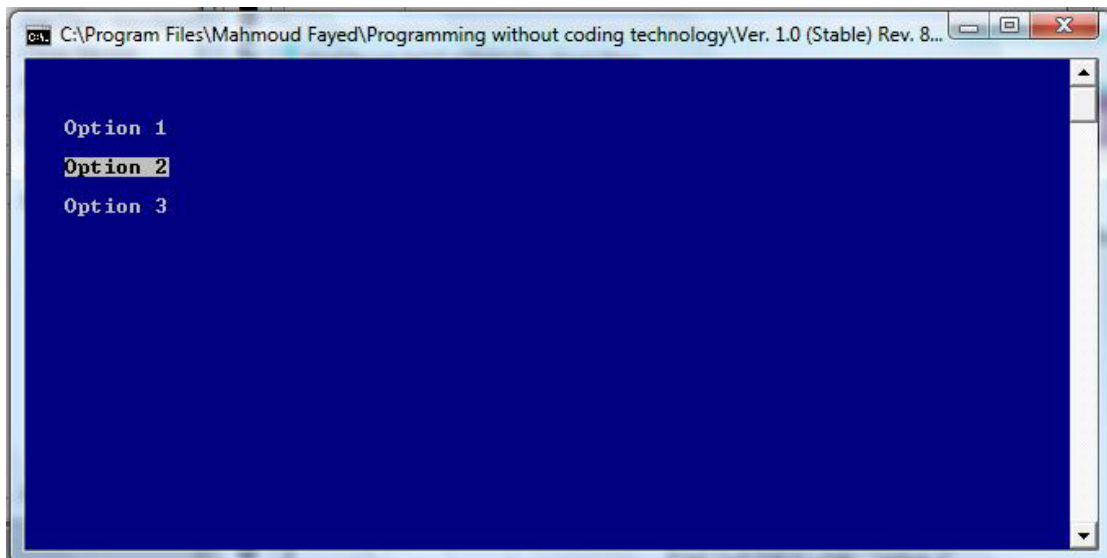
Steps Tree



Interaction Page



Final Steps Tree



Final Application

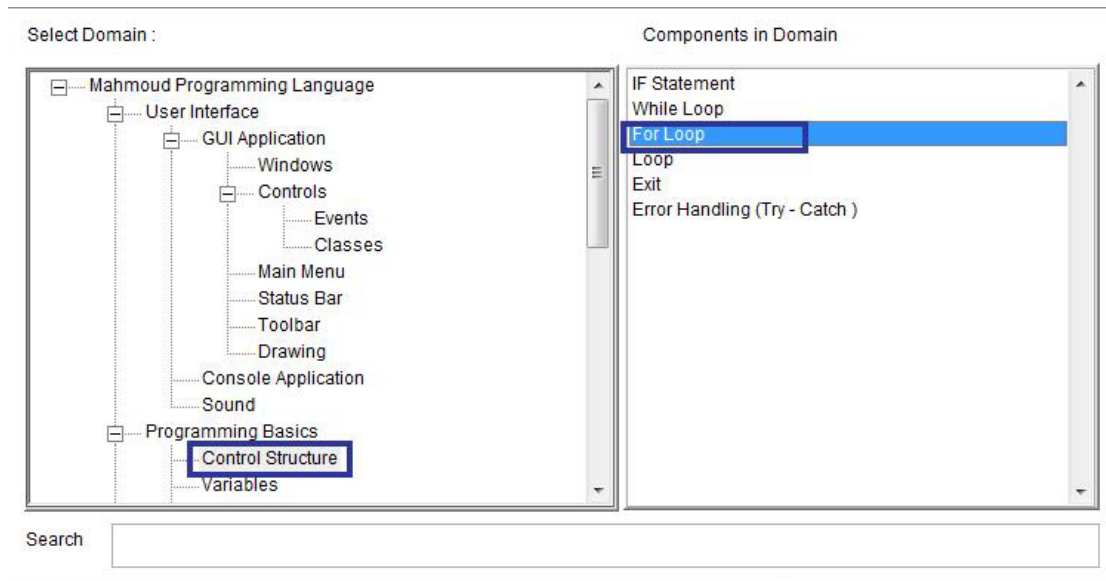


Final Application

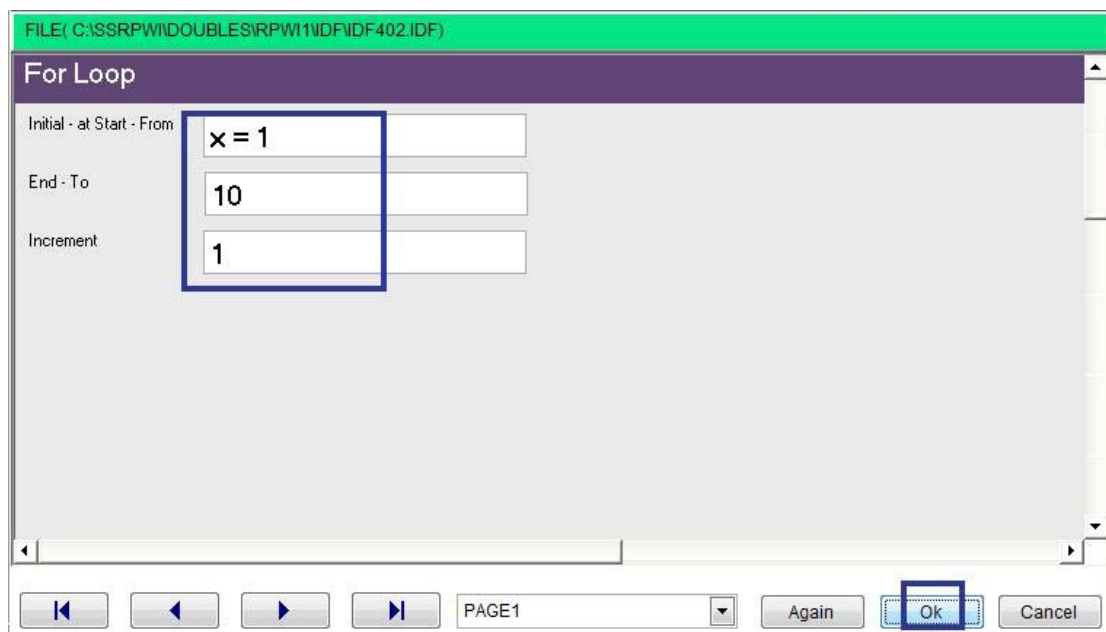
For Loop

- Domain (Control Structure)
- Component (For Loop)

Example - Screen shots:-



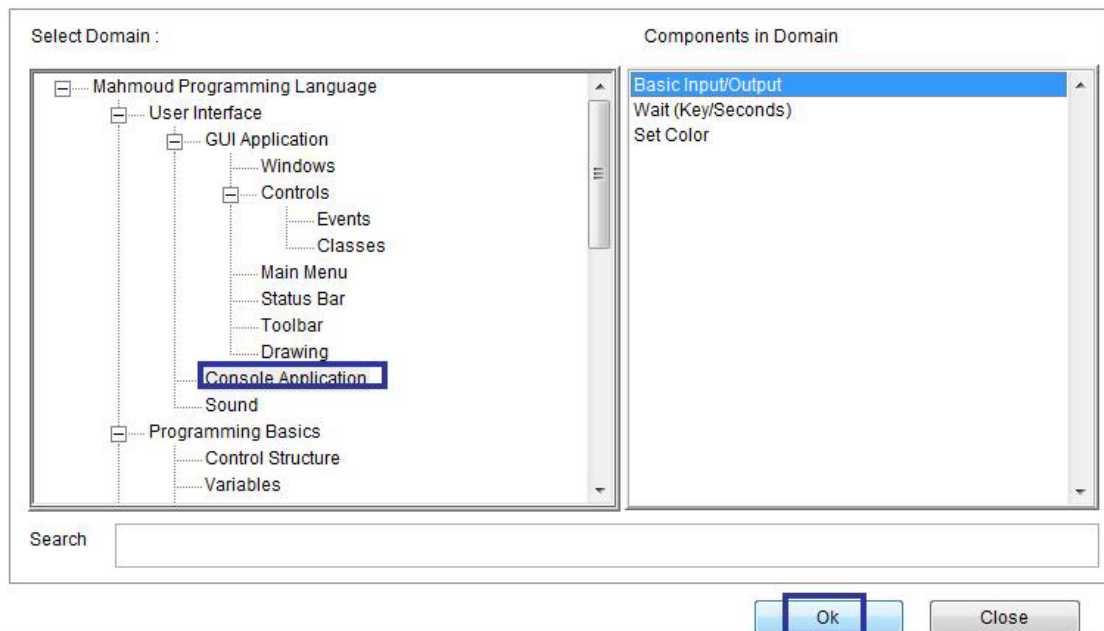
Domain (Control Structure) Component (For Loop)



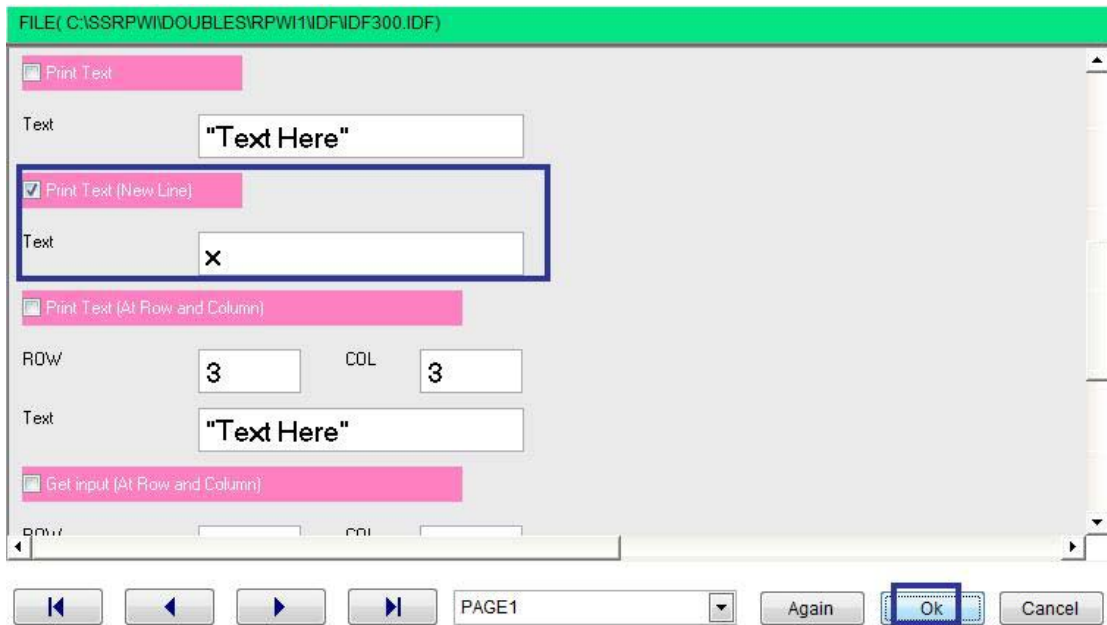
Interaction Page



Steps Tree



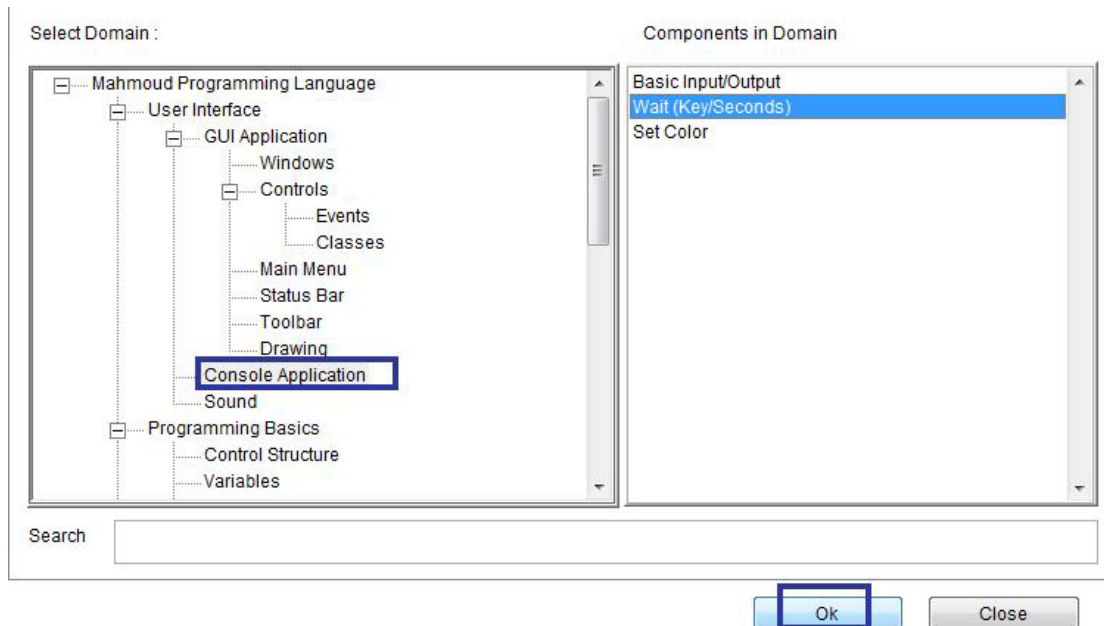
Domain (Console Application) Component (Basic Input/output)



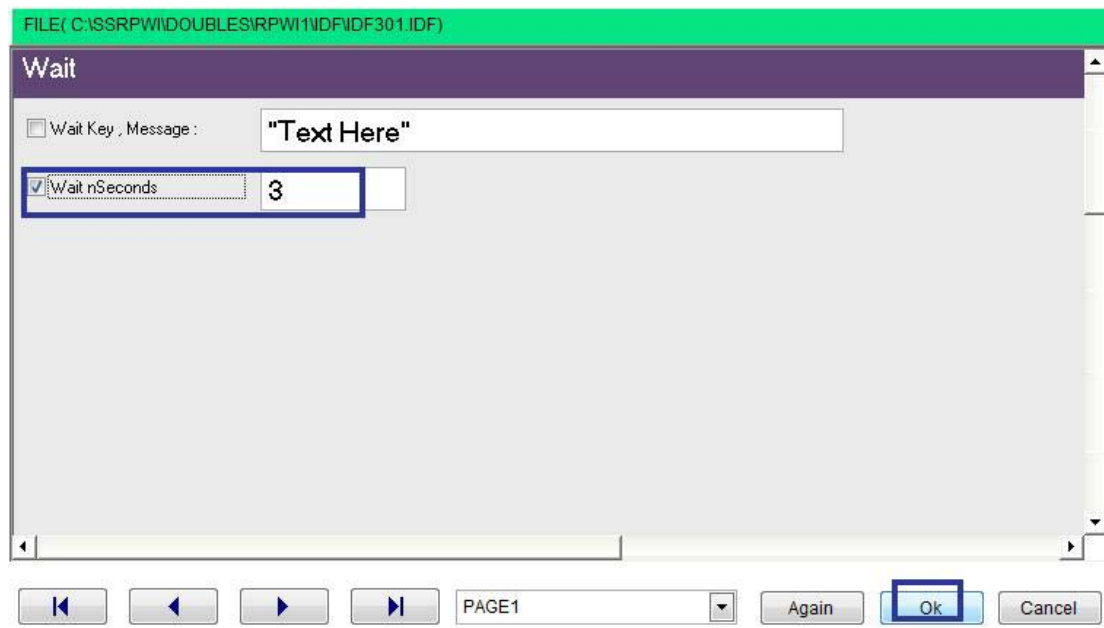
Interaction Page



Steps Tree



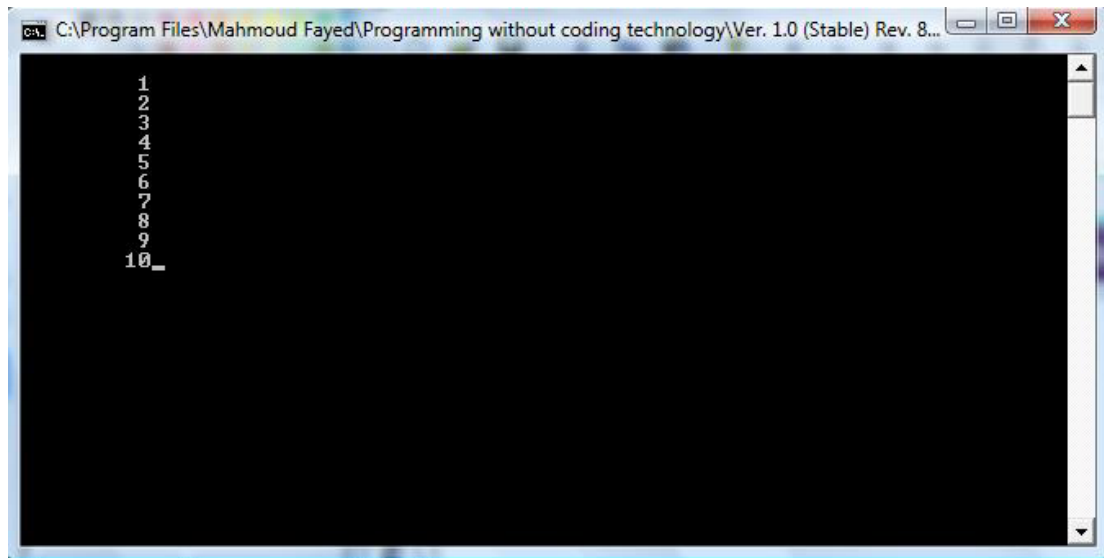
Domain (Console Application) Component (Wait Key/Seconds)



Interaction Page



Final Steps Tree

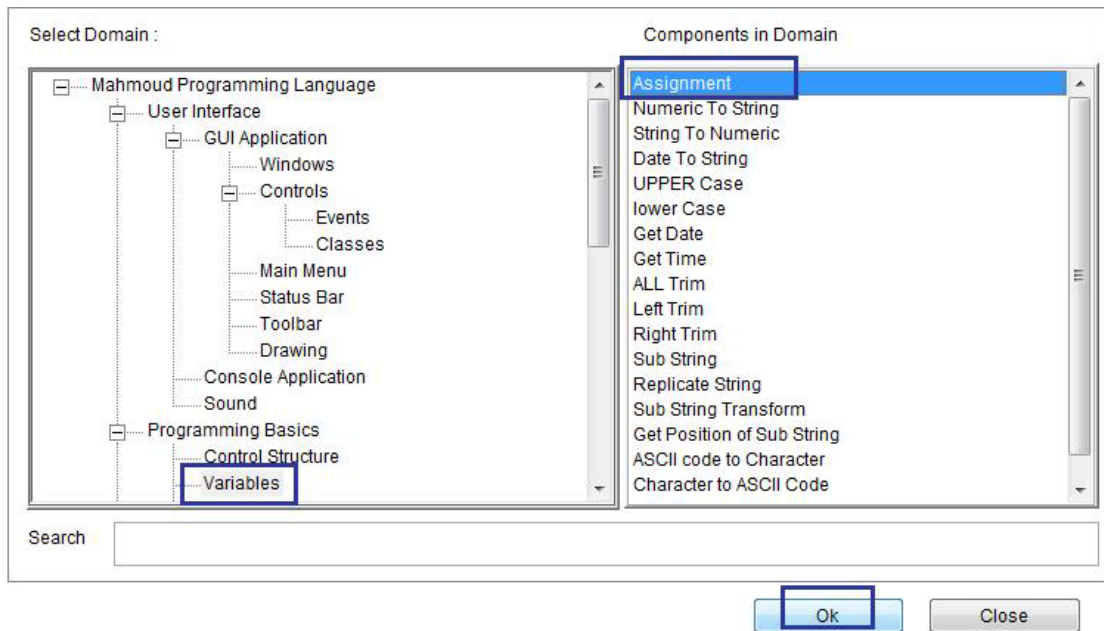


Final Application

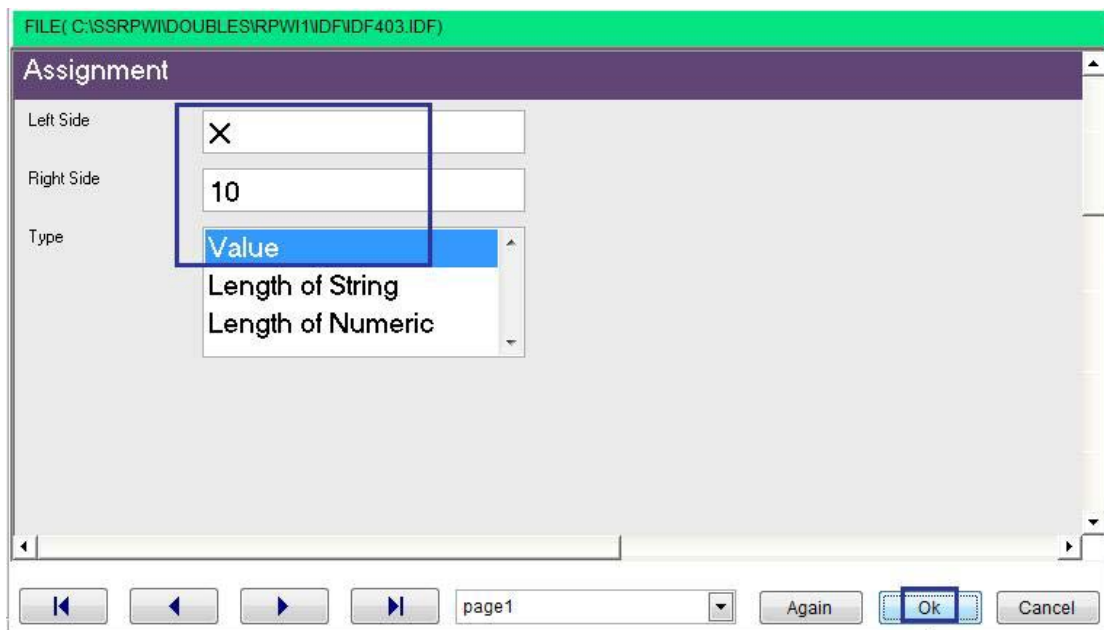
While Loop

- Domain (Control Structure)
- Component (While Loop)

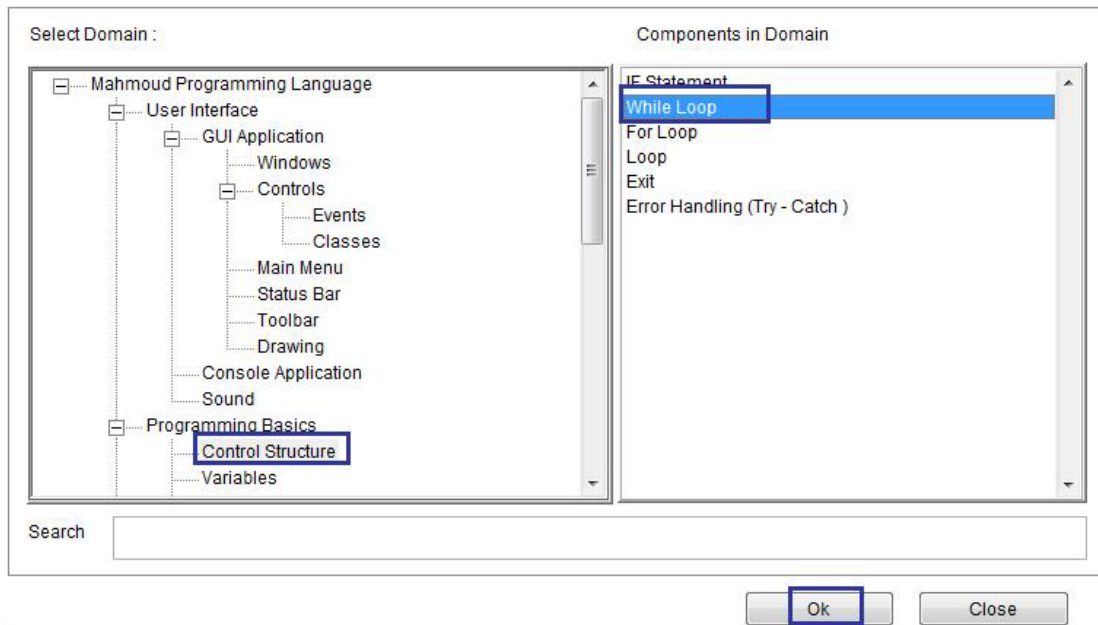
Example - Screen shots:-



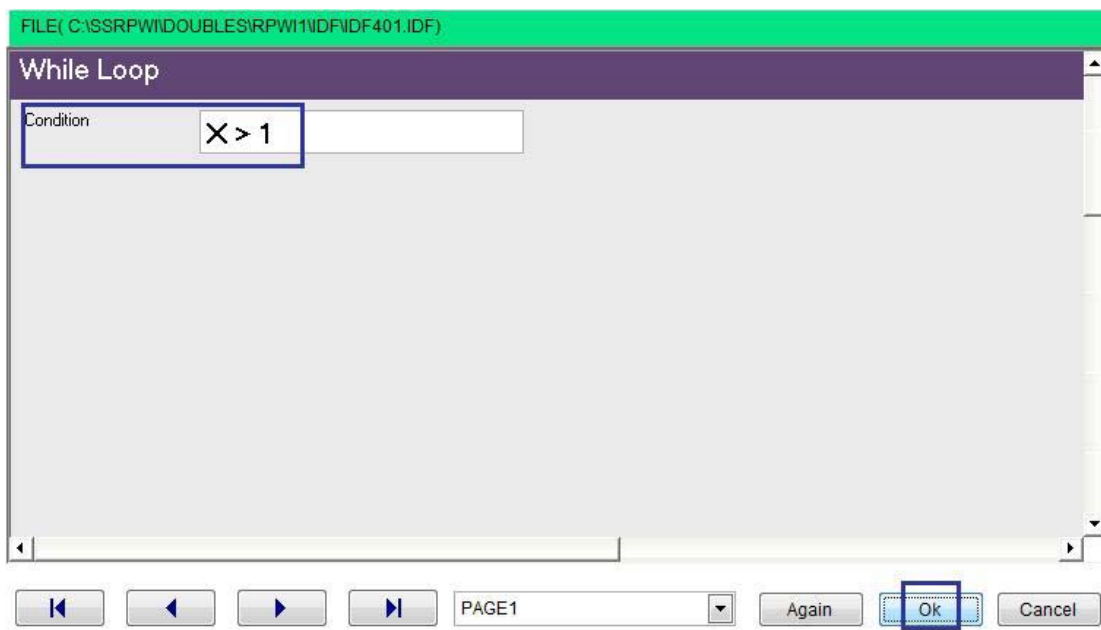
Domain (Variables) Component (Assignment)



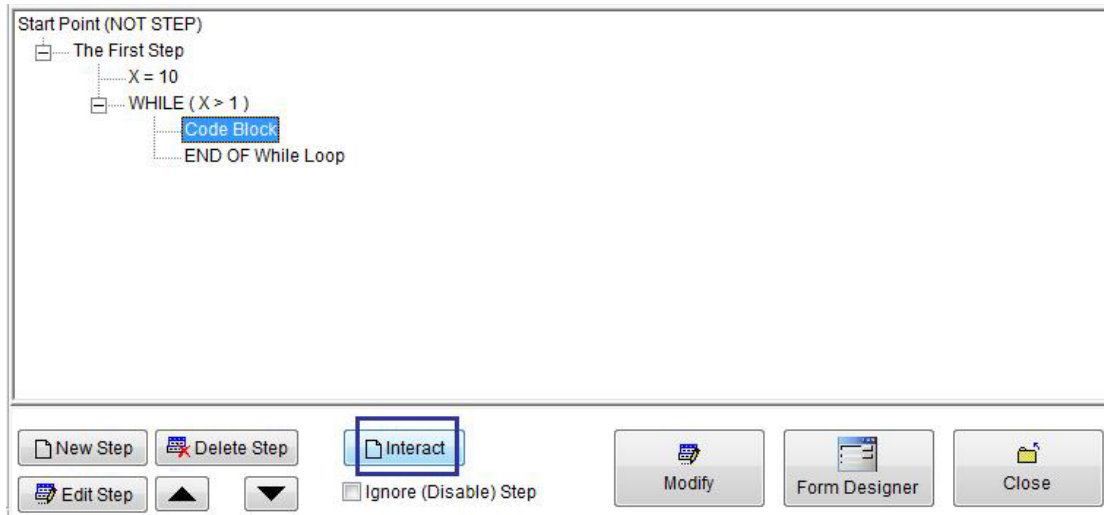
Interaction Page



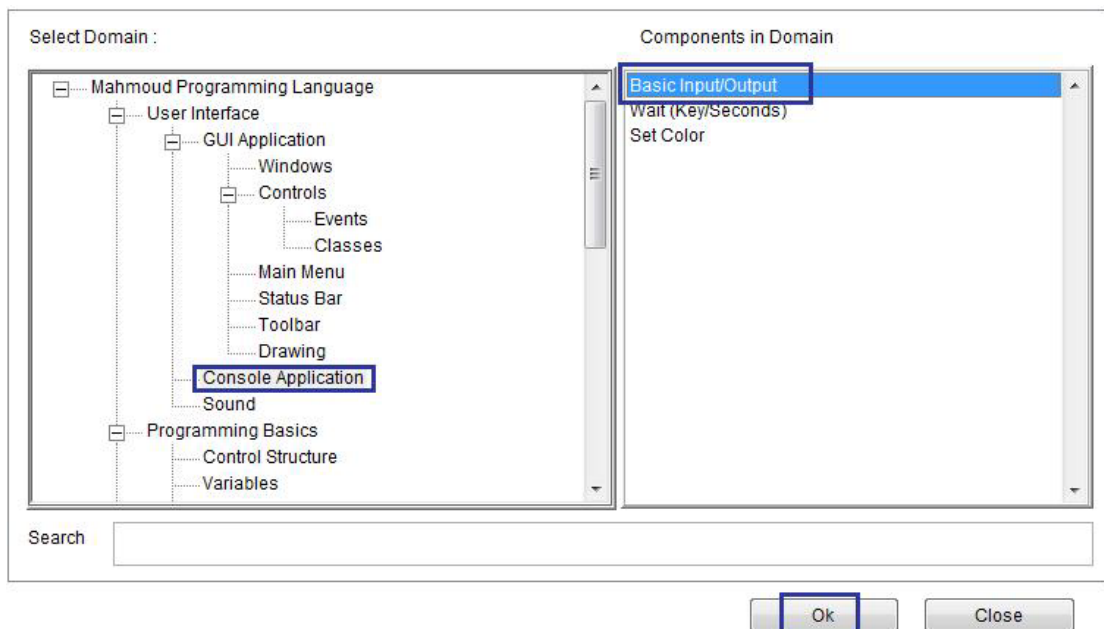
Domain (Control Structure) Component (While Loop)



Interaction Page



Steps Tree



Domain (Console Application) Component (Basic Input/output)

FILE(C:\SSRPW\IDOUBLES\IRPW1\IDF\IDF300.IDF)

Print Text

Text "Text Here"

Print Text (New Line)

Text X

Print Text (At Row and Column)

ROW 3 COL 3

Text "Text Here"

Get input (At Row and Column)

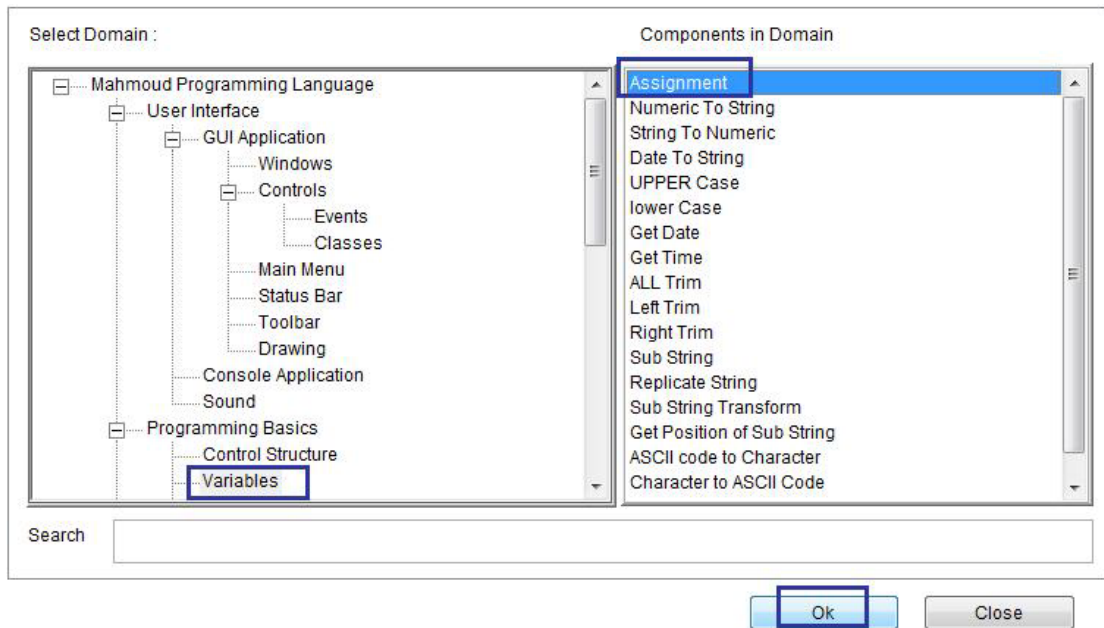
ROW COL

PAGE1

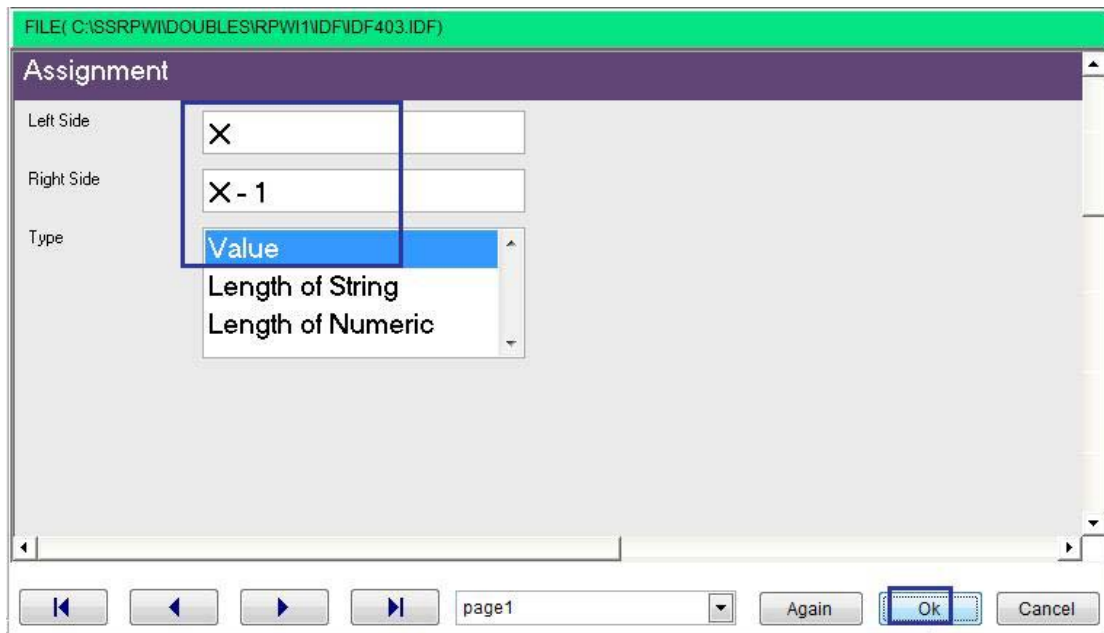
Interaction Page



Steps Tree



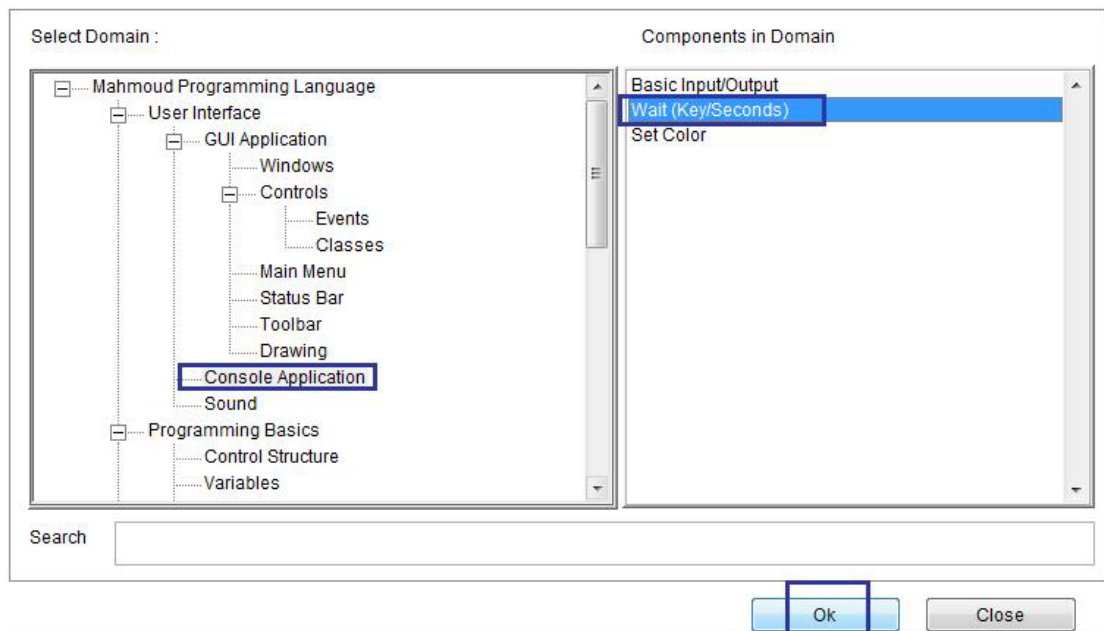
Domain (Variables) Component (Assignment)



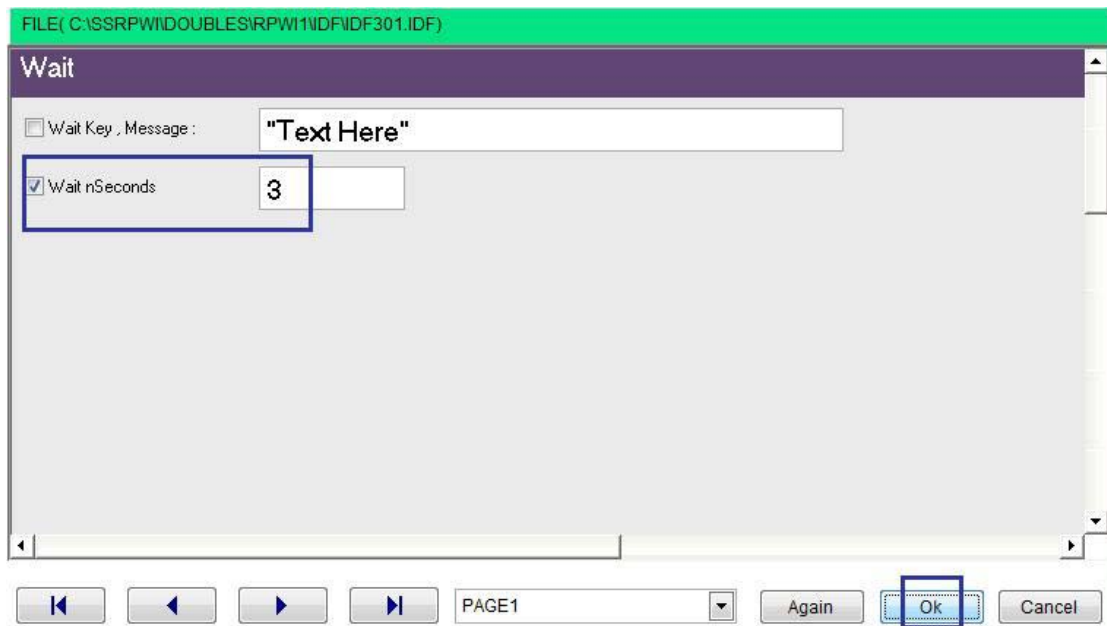
Interaction Page



Steps Tree



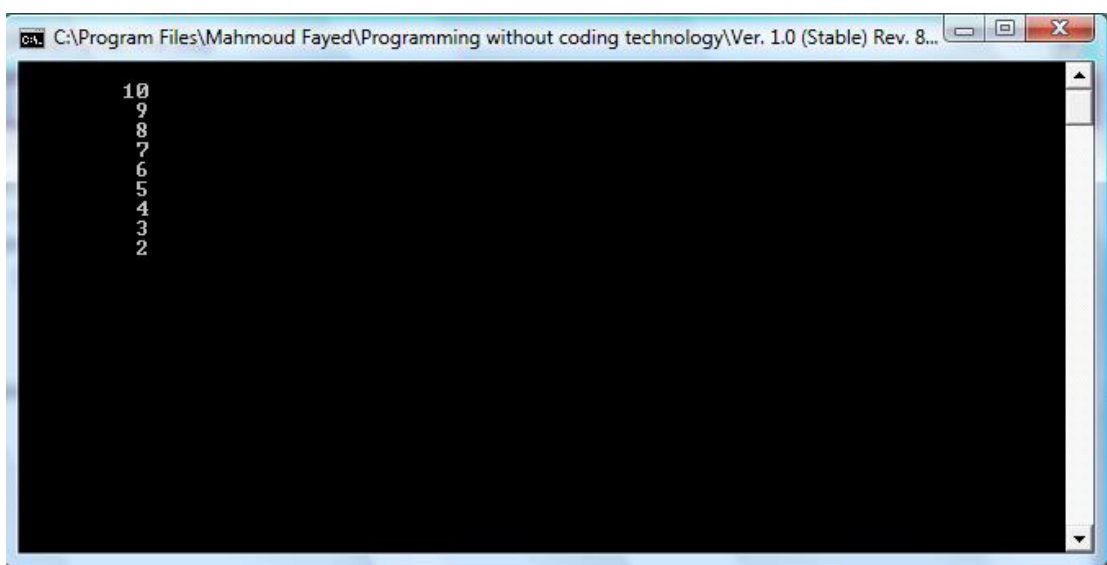
Domain (Console Application) Component (Wait Key/Seconds)



Interaction Page



Final Steps Tree



Final Application

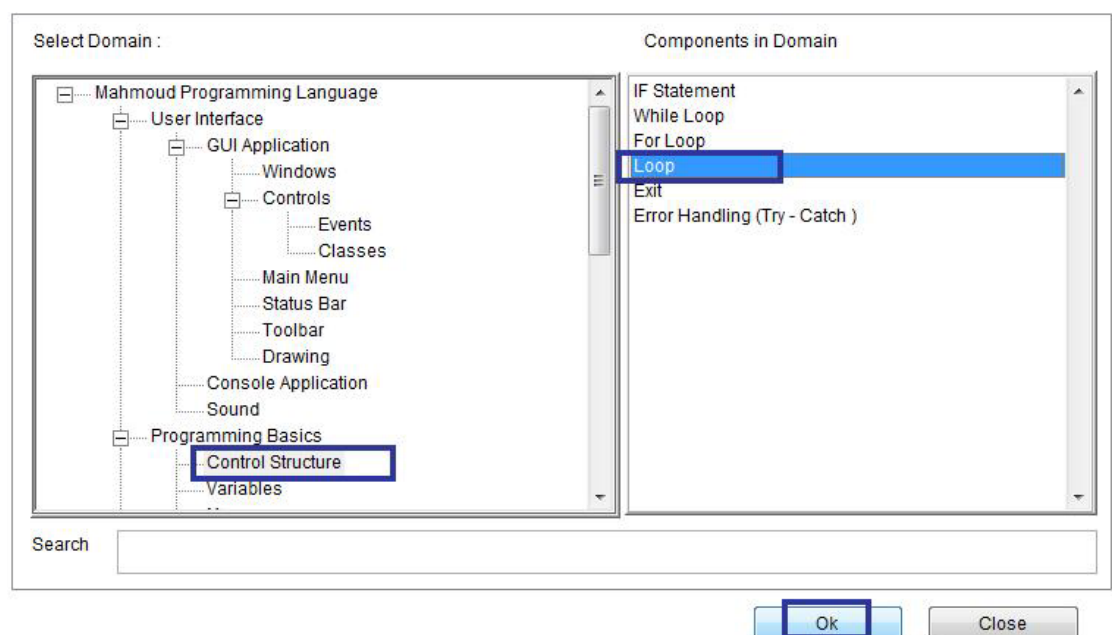
Loop and Exit

Components

- Loop component
- Exit component

Loop Component

- Domain (Control Structure)
- Component Name (Loop)

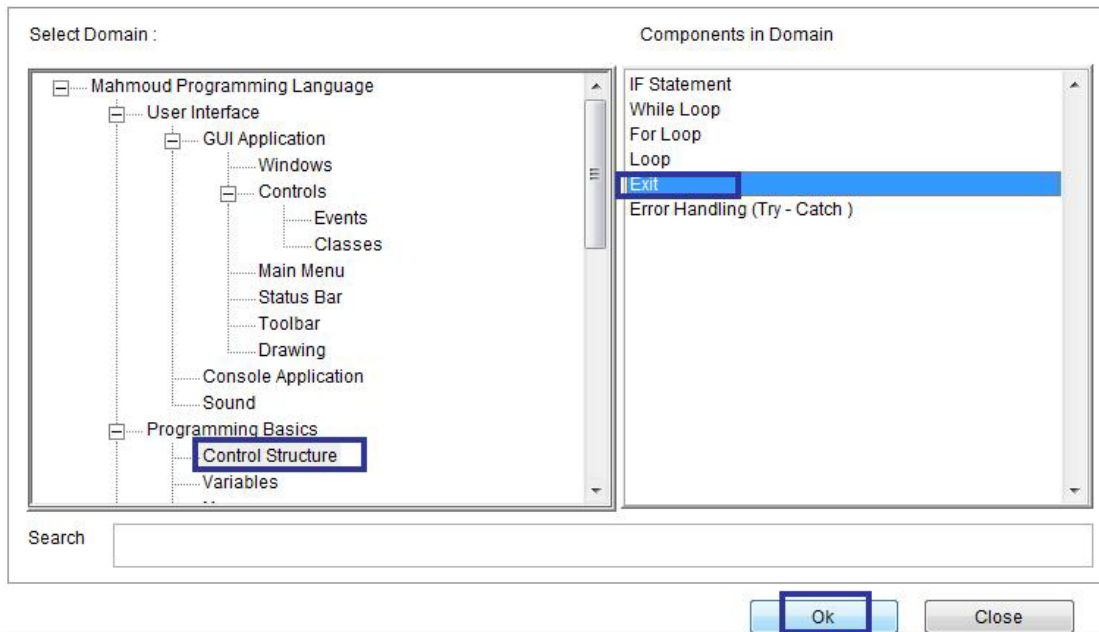


Domain (Control Structure) Component (Loop)

The Loop step restarts the current iteration of the enclosing Loop structure, and if the enclosing loop is a FOR Loop, it changes the loop variable (Increase or decrease) to the value of the next iteration of the loop

Exit Component

- Domain (Control Structure)
- Component Name (Exit)



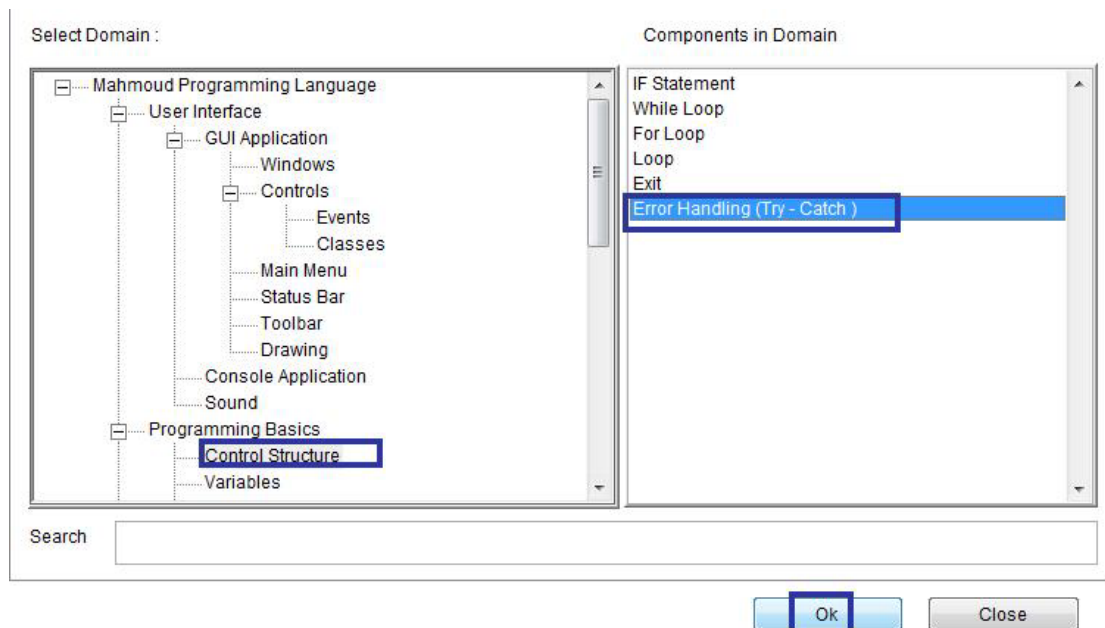
Domain (Control Structure) Component (Exit)

The EXIT step immediately terminates execution of the enclosing loop structure.

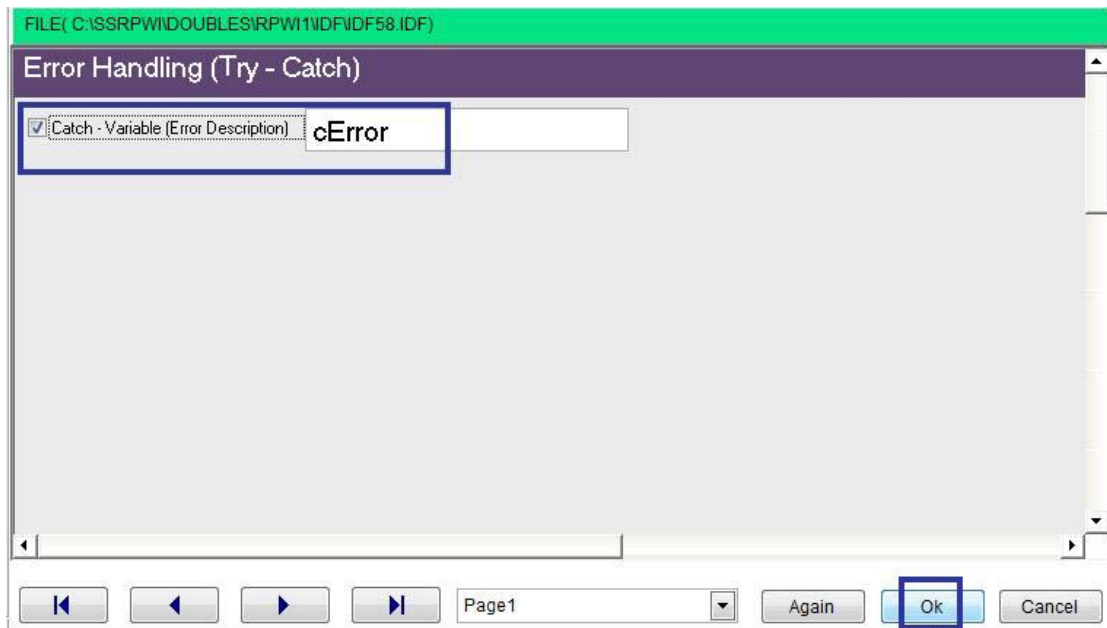
Error Handling (Try – Catch)

- Domain (Control Structure)
- Component Name (Error Handling - Try - Catch)

Automatically integrate error handling, so that any error will be intercepted, and recovered by means of the CATCH step or ignored



Domain (Control Structure) – Component (Error handling – try – catch)



Interaction Page



Steps Tree

Memo Variables

Memo variables are multi line strings.

Domain (Memo)

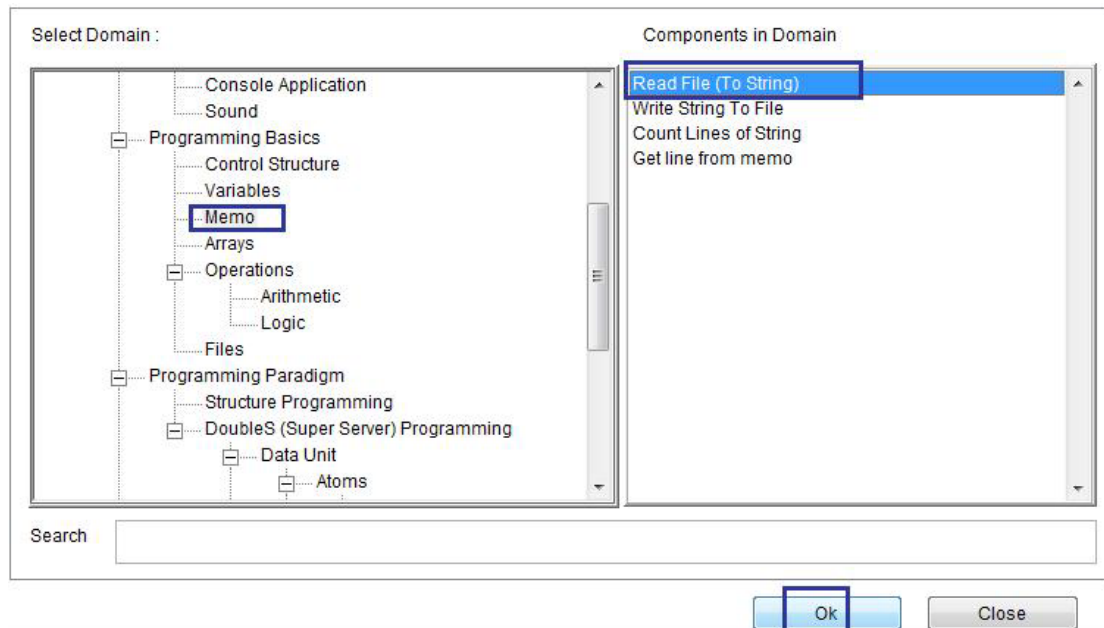
Components:

- Read File (To String)
- Write String (To File)
- Count Lines of String
- Get Line from memo

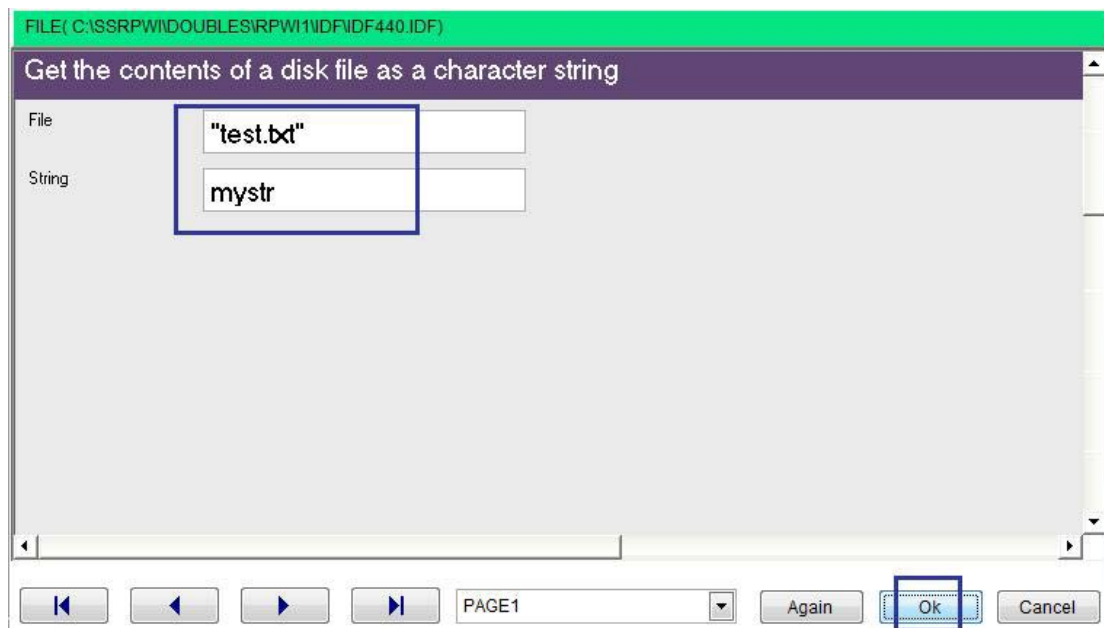
Read File (To String)

- Domain (Memo)
- Component (Read File To String)

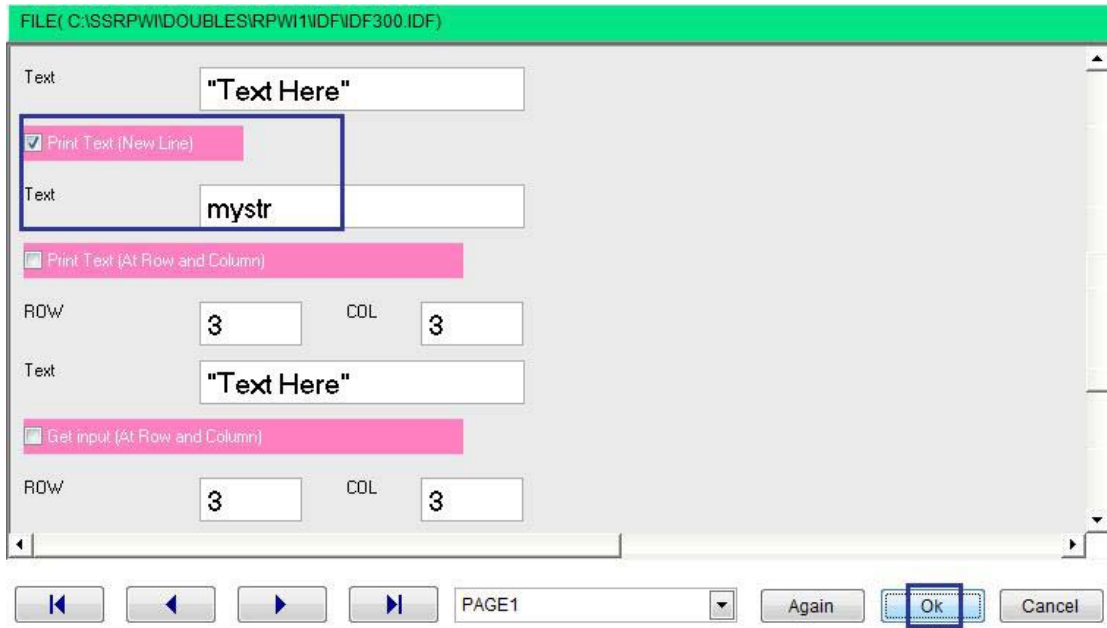
Example - Screen shots:-



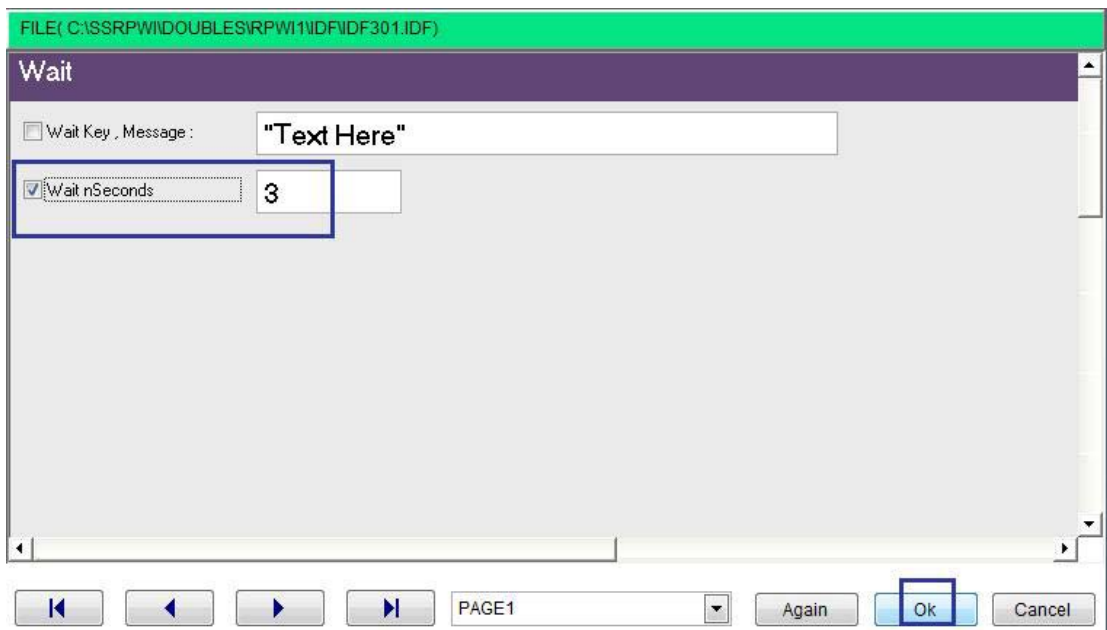
Domain (Memo) Component (Read File To String)



Interaction Page



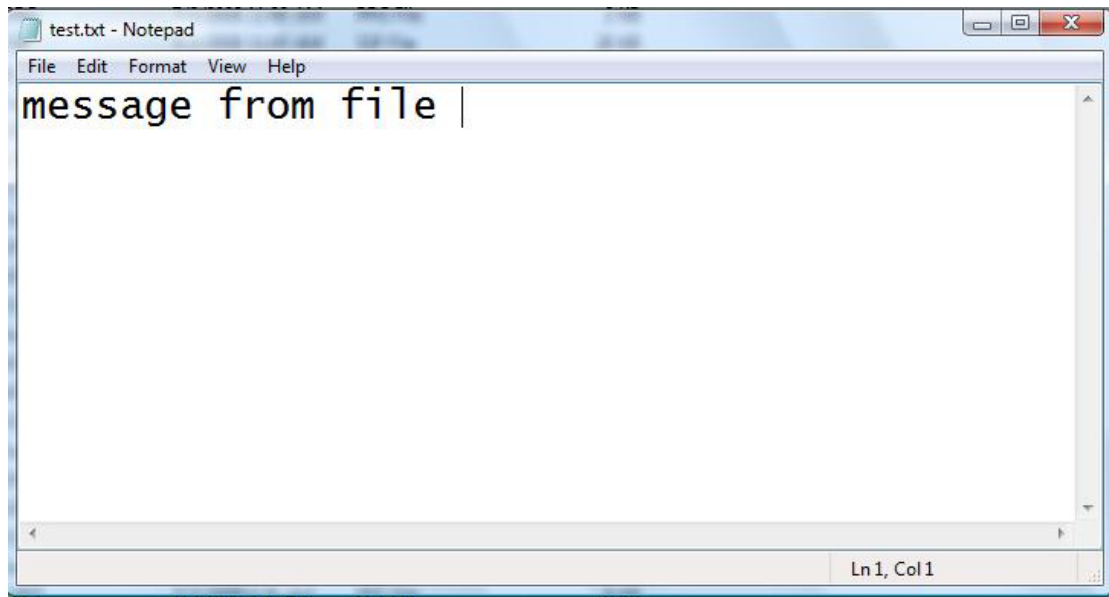
Interaction Page



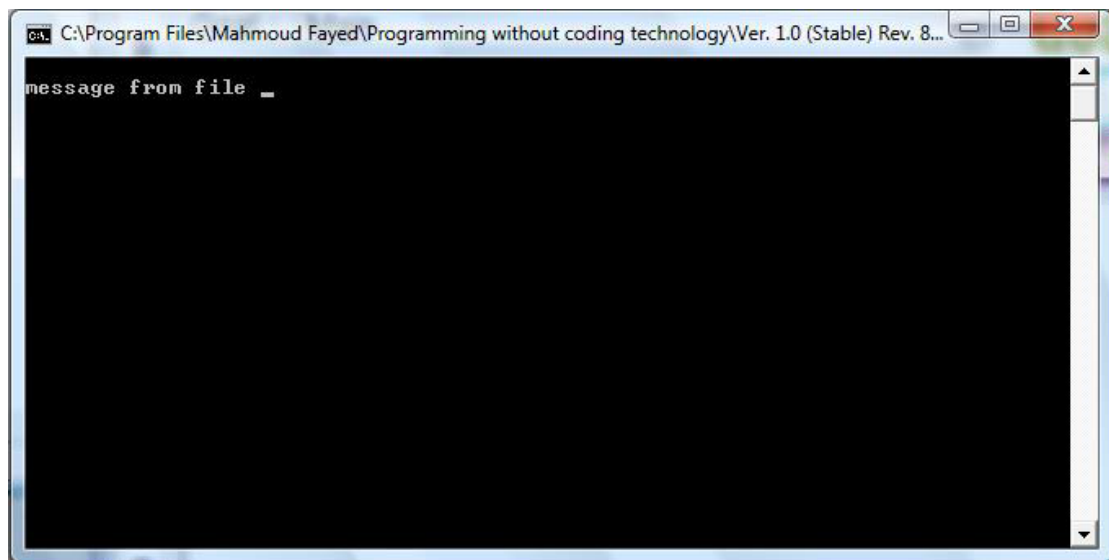
Interaction Page



Steps Tree



File: test.txt

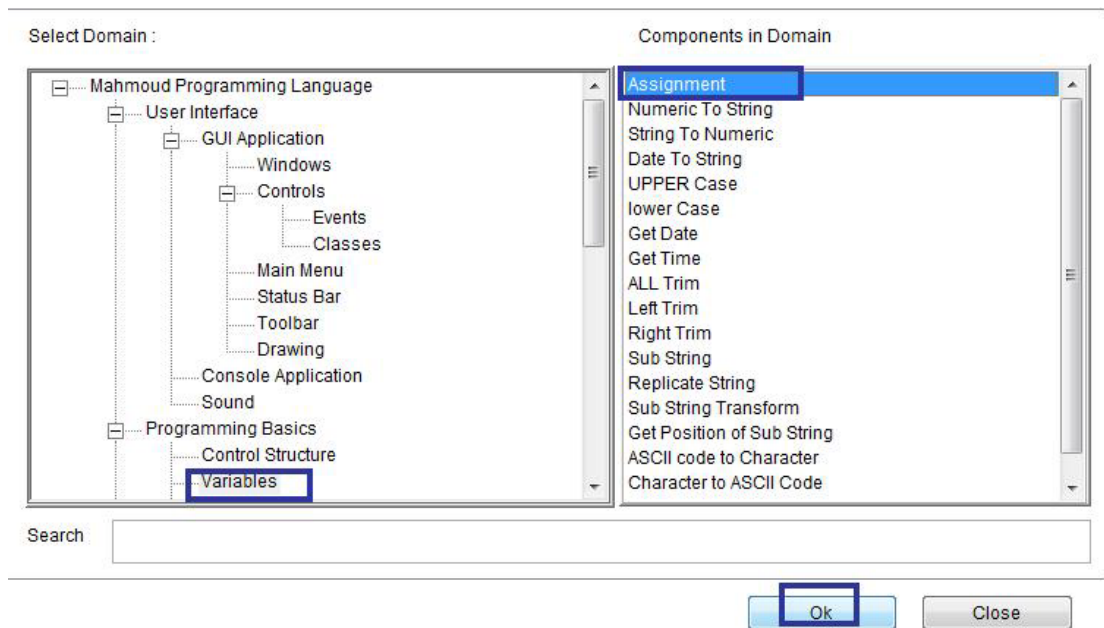


The Final application

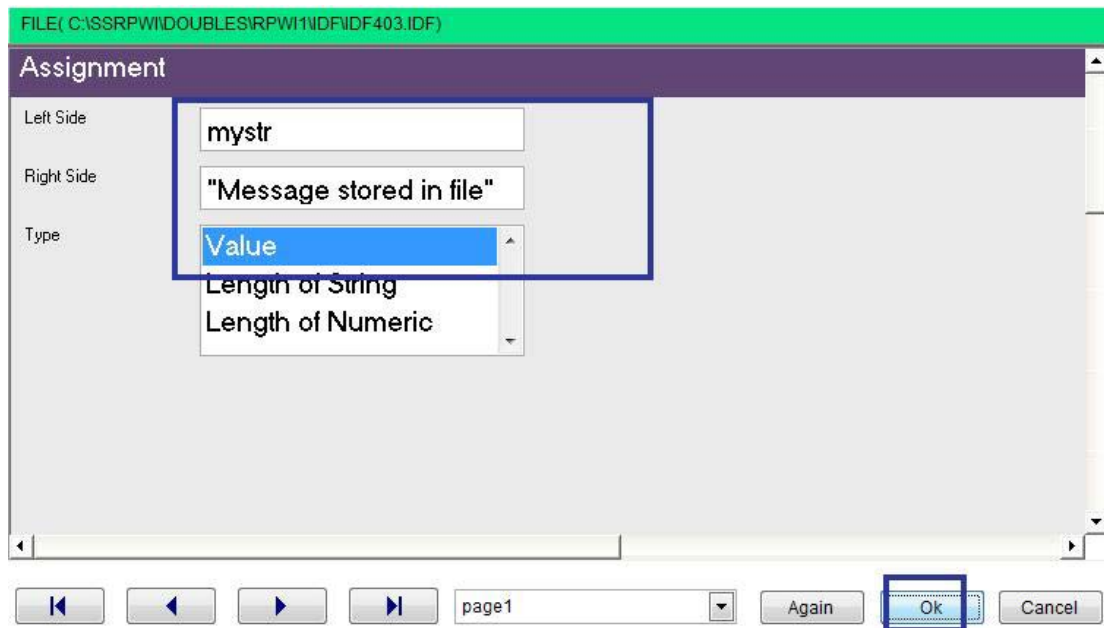
Write string to file

- Domain (Memo)
- Component (Write string to file)

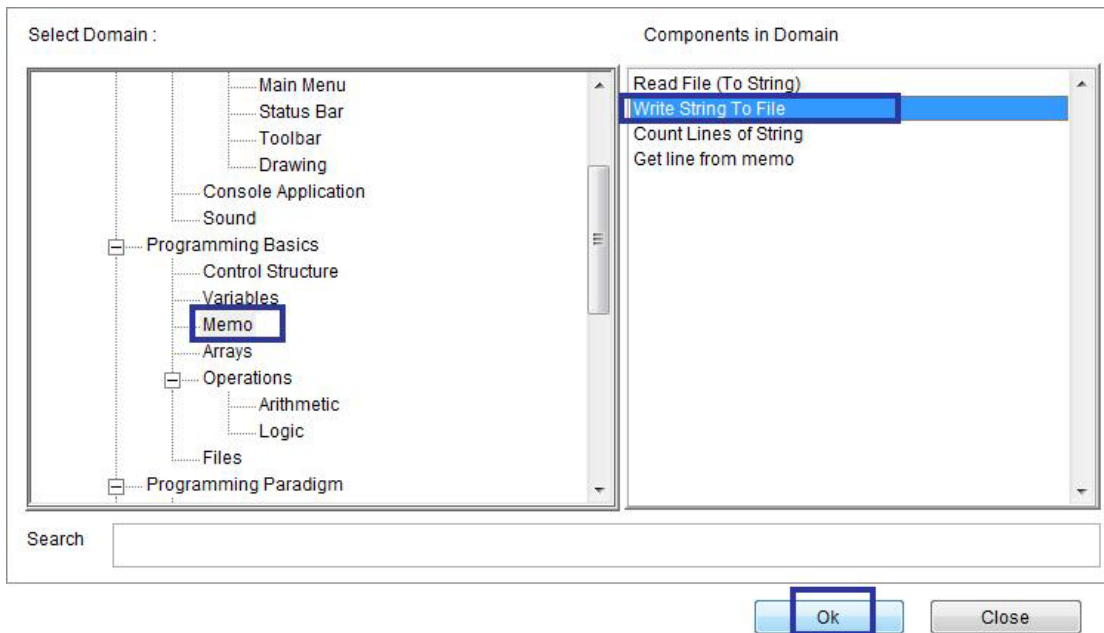
Example - Screen shots:-



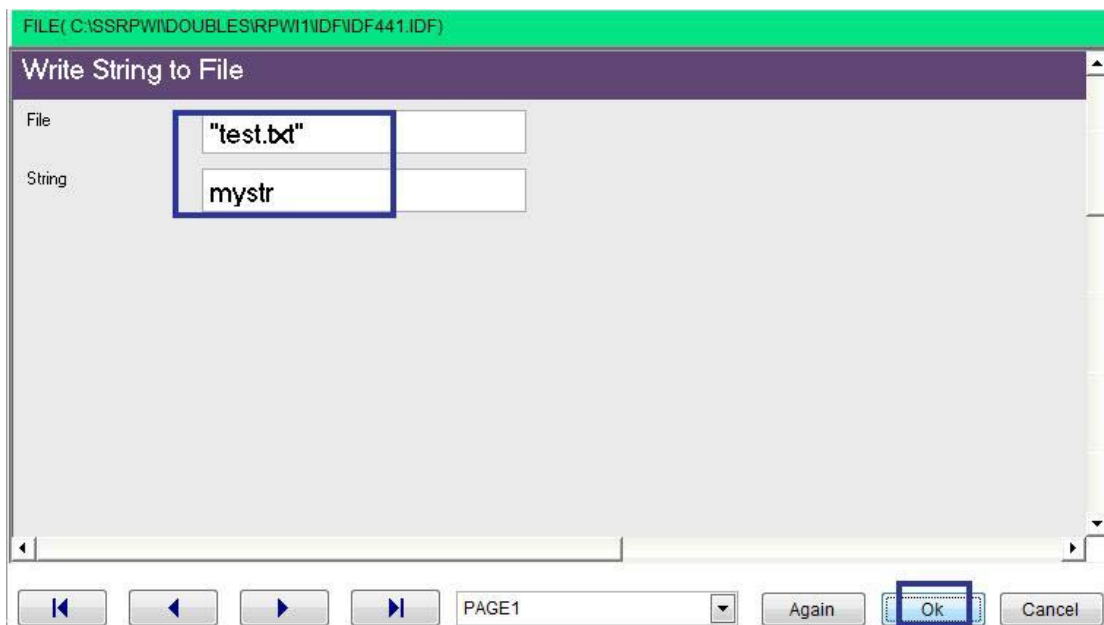
Domain (Variables) Component (Assignment)



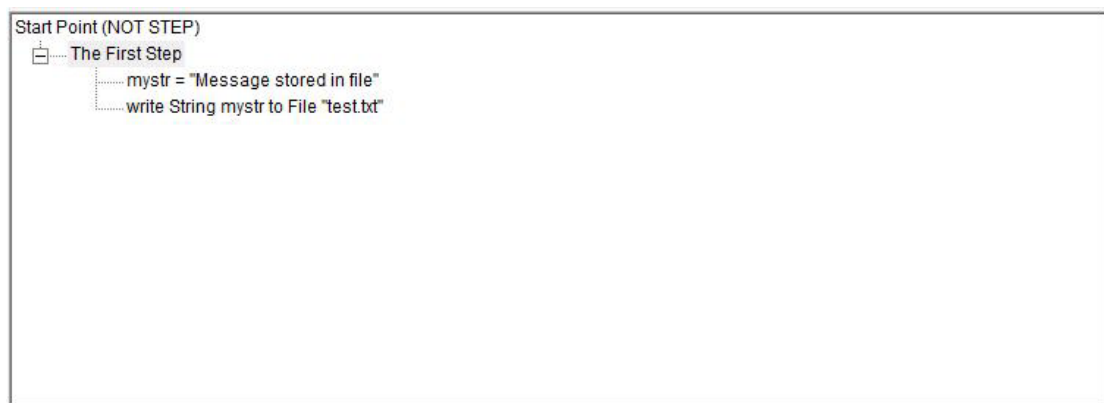
Interaction Page



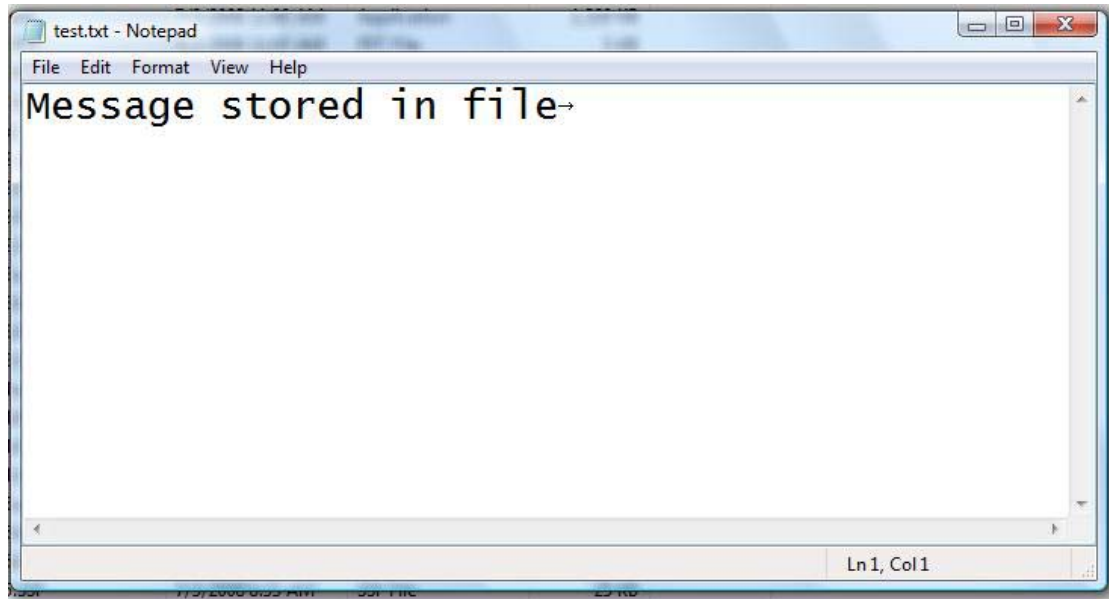
Domain (Memo) Component (Write string to File)



Interaction Page



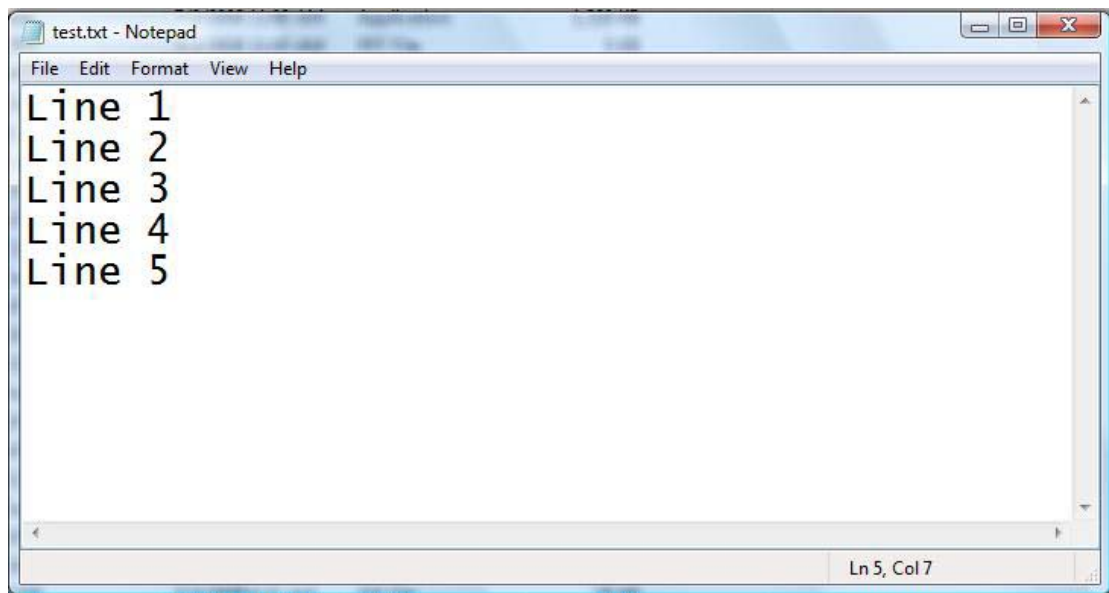
Steps Tree



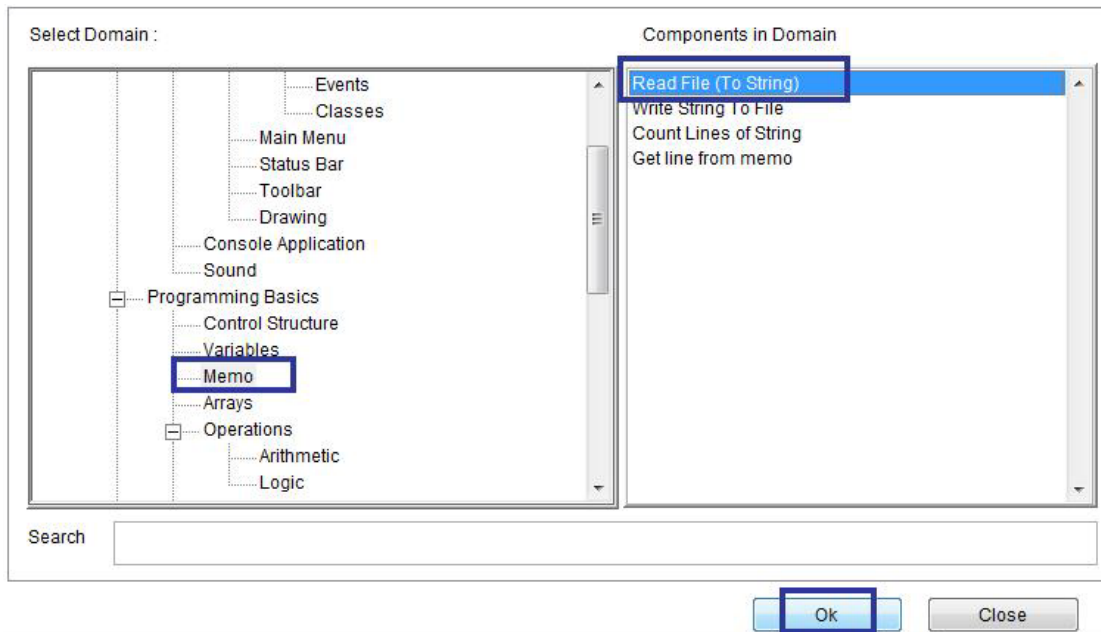
Test.txt

Count Lines of String & Getting Line from Memo

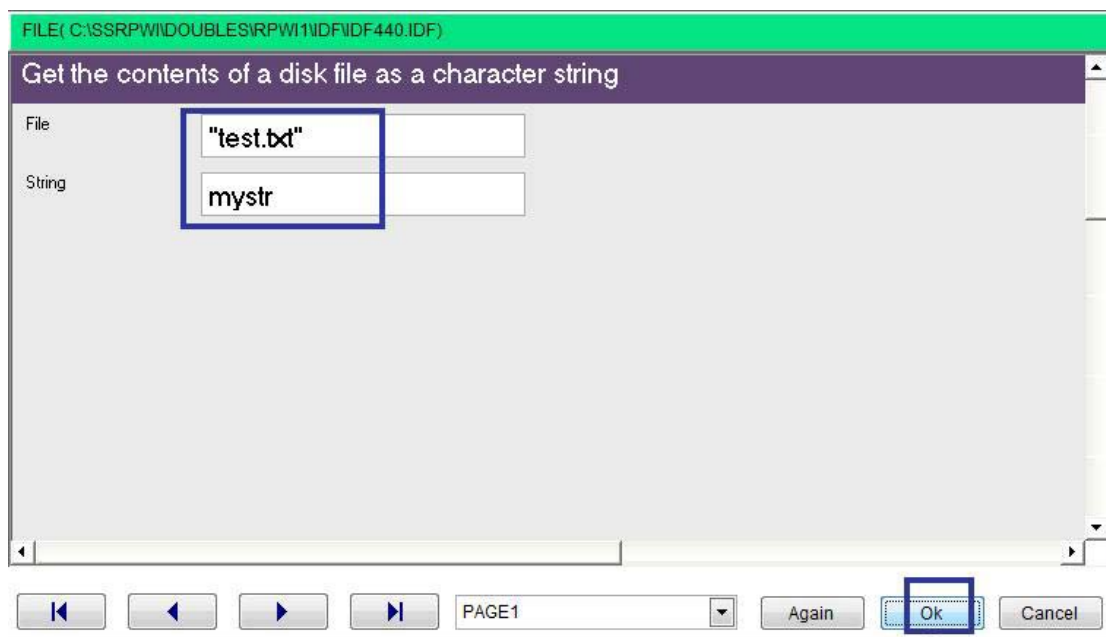
- Domain (Memo)
- Component (Count Lines of String)
- Component (Getting Line From Memo)



Test.txt



Domain (Memo) Variable (Read File to string)



Interaction Page