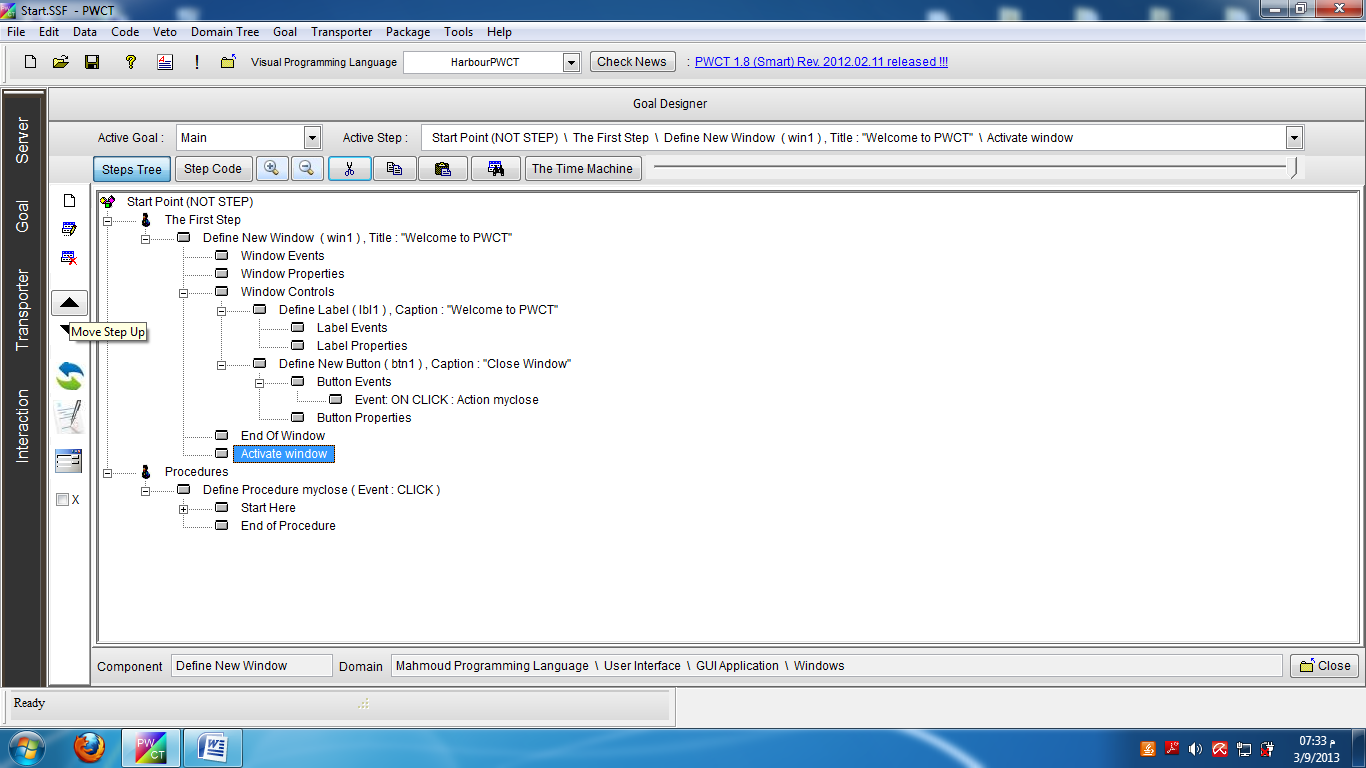
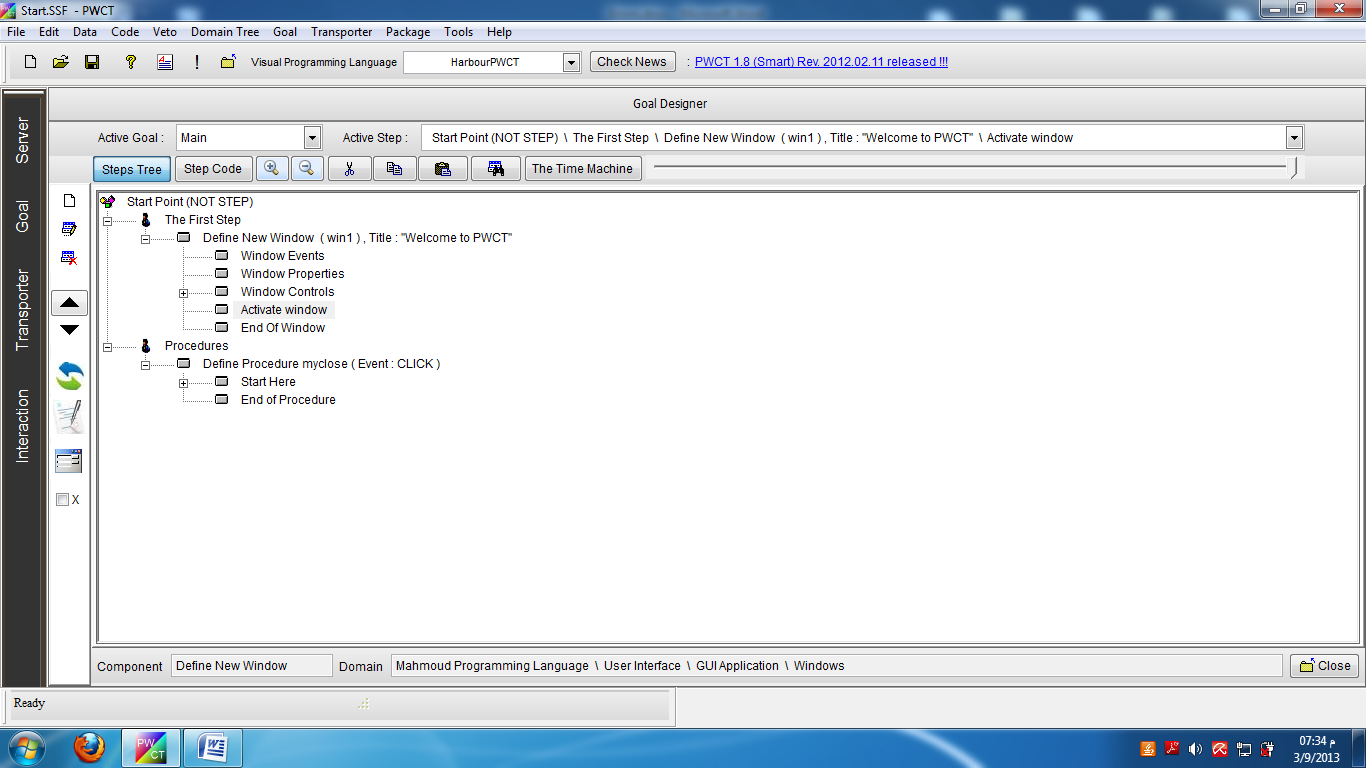
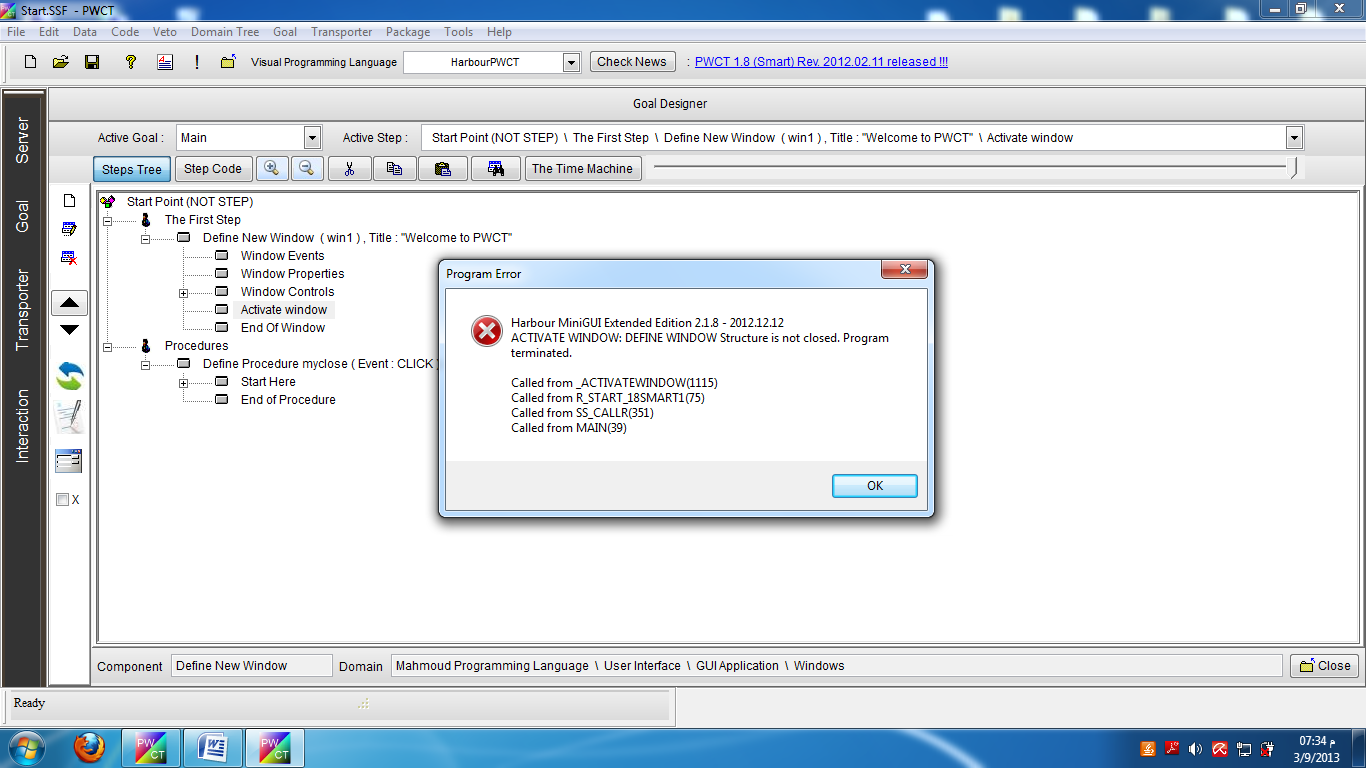
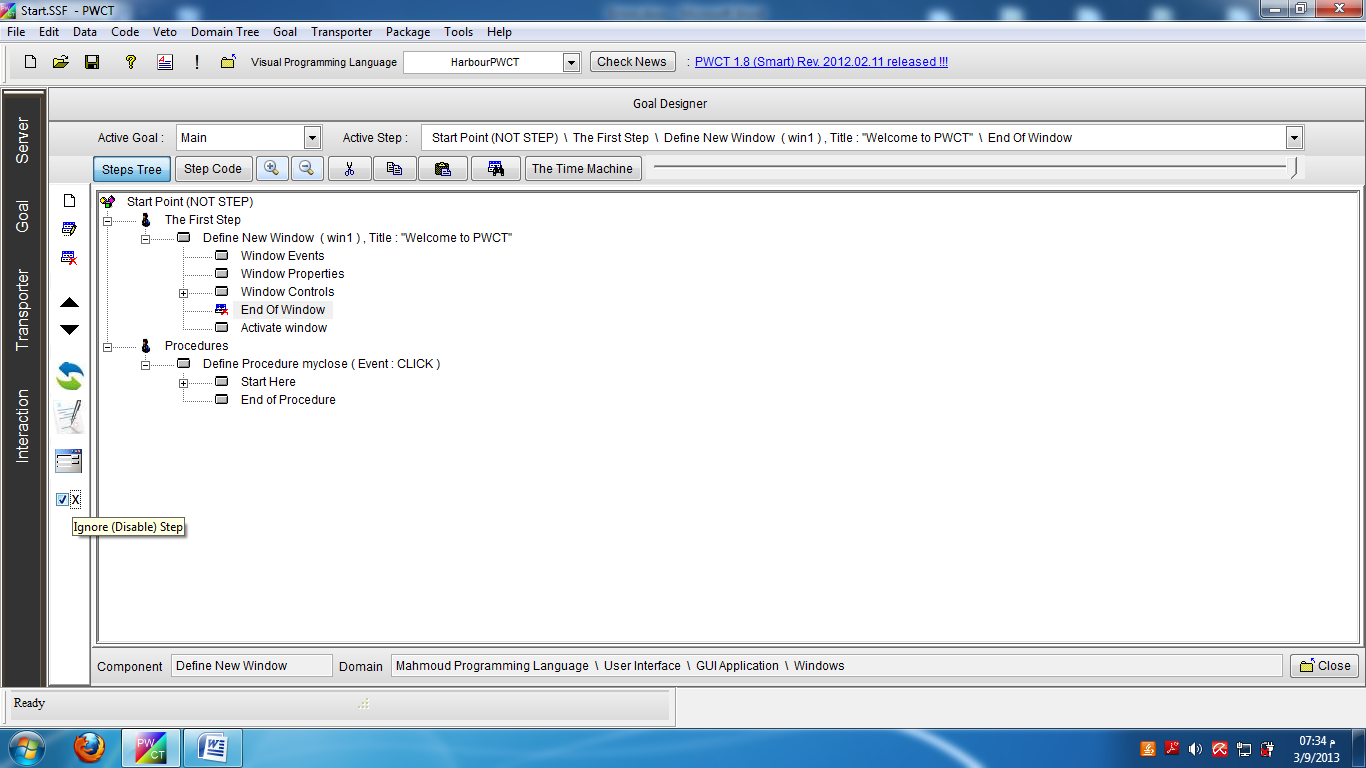
1. **Change the order of the generated step (Move UP – Move Down)**

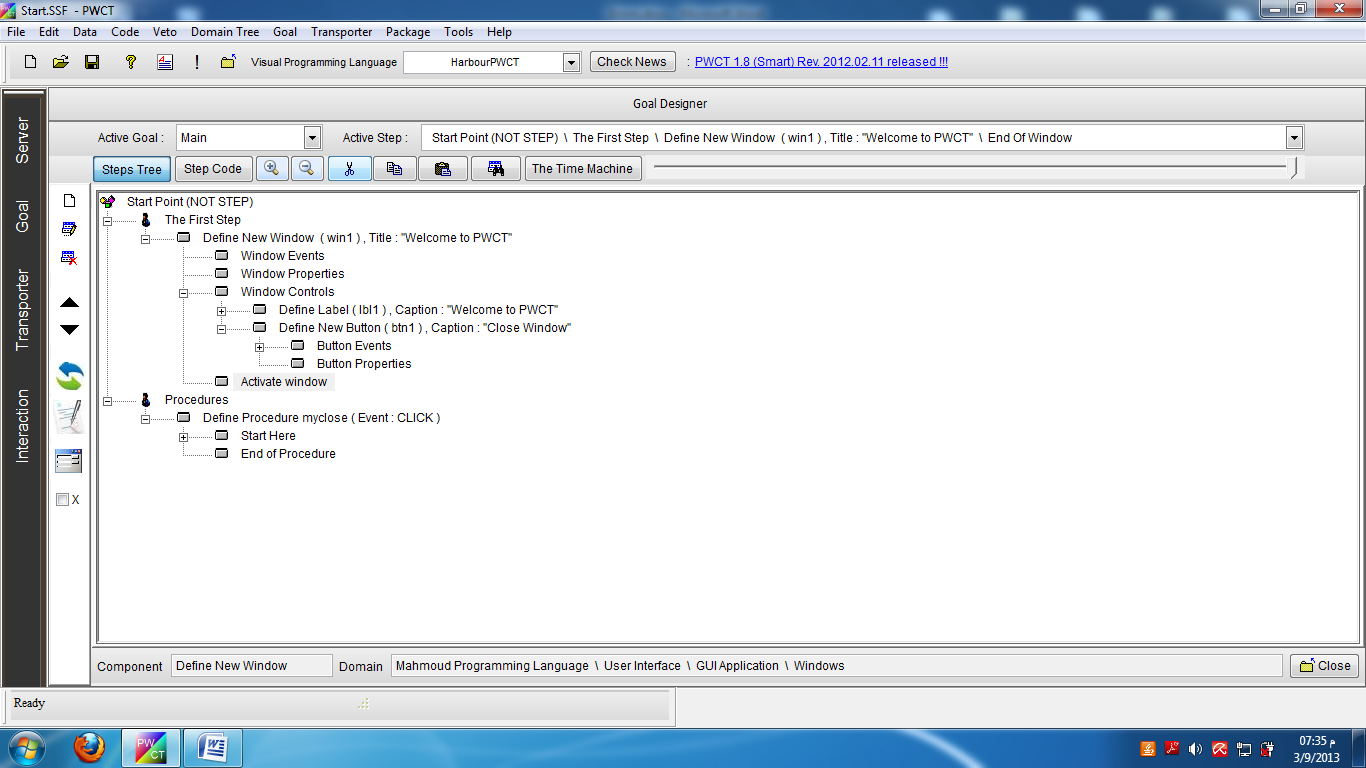




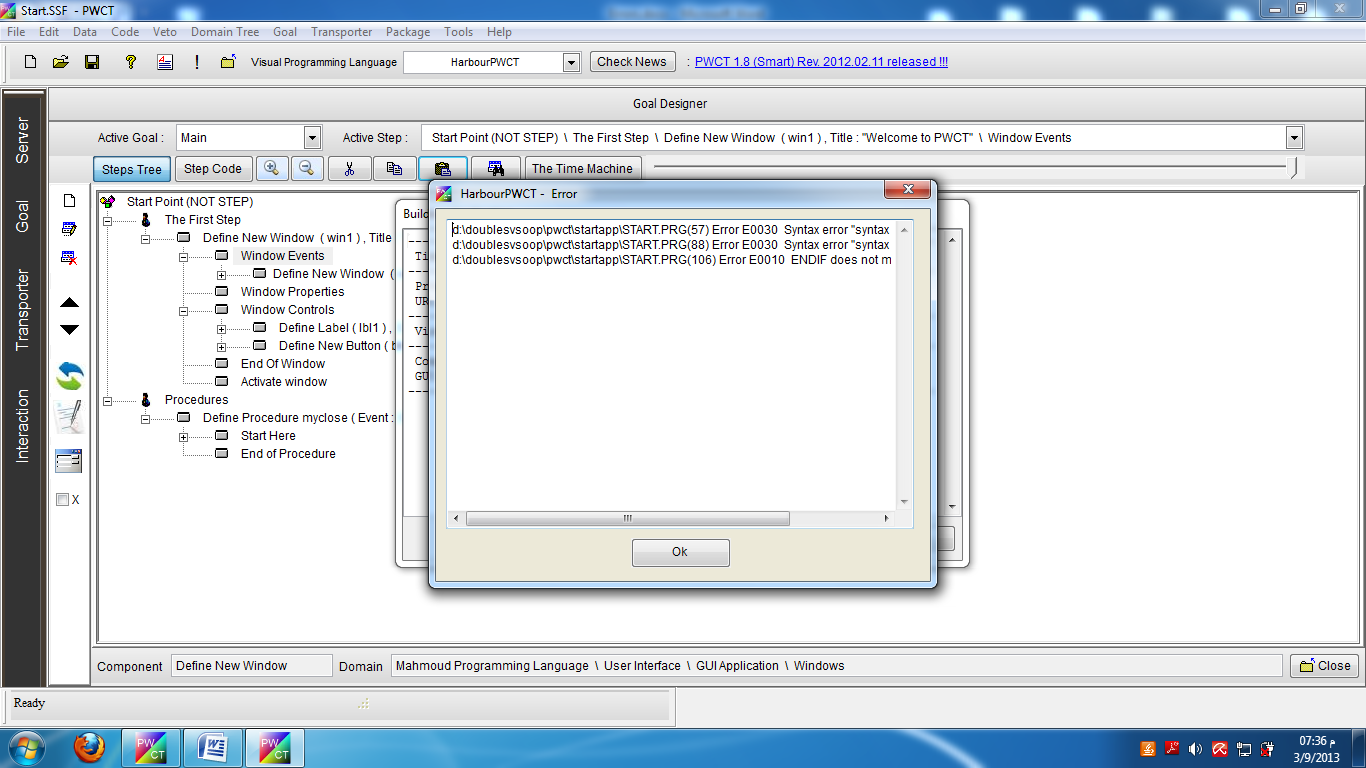


1. **Ignore/Cut/Delete generated step**



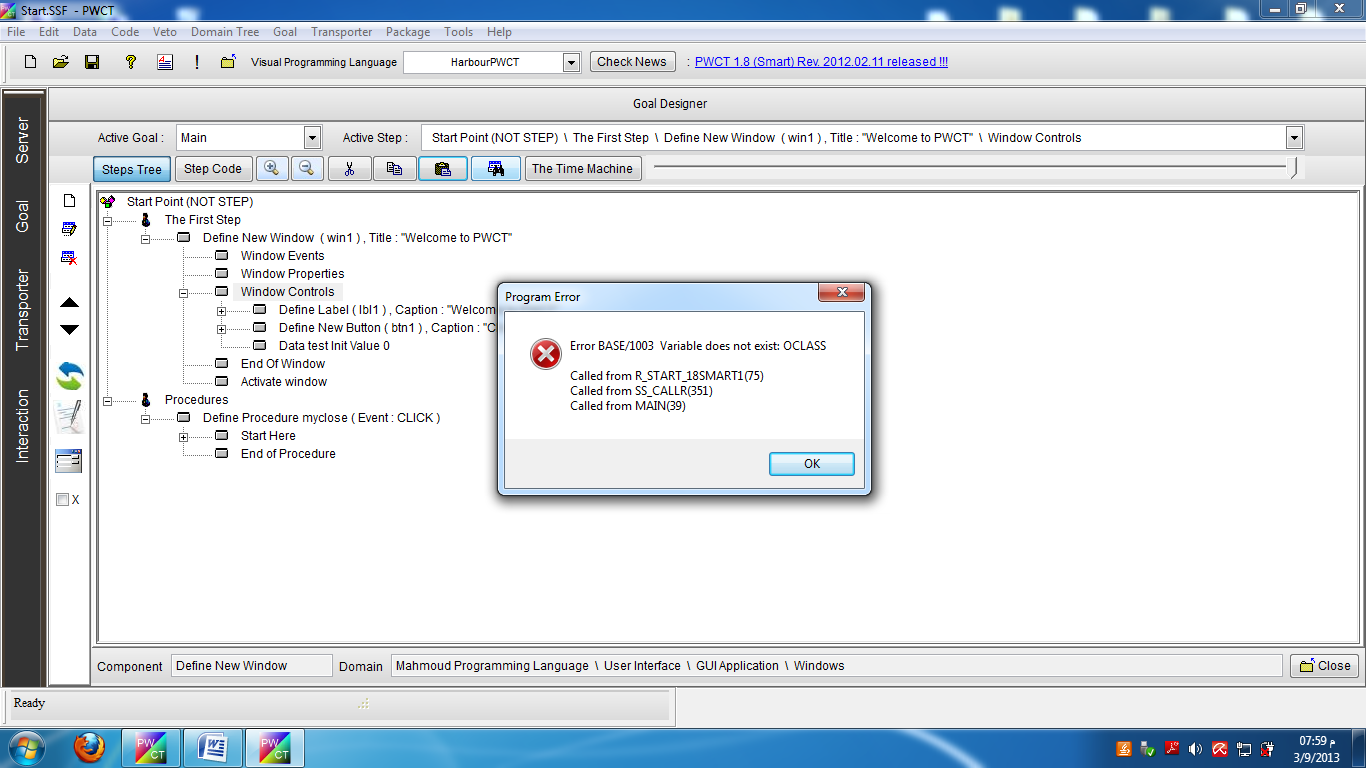


1. **Copy and Paste generated step in bad place**



1. **Generate step in a bad position (using interaction)**

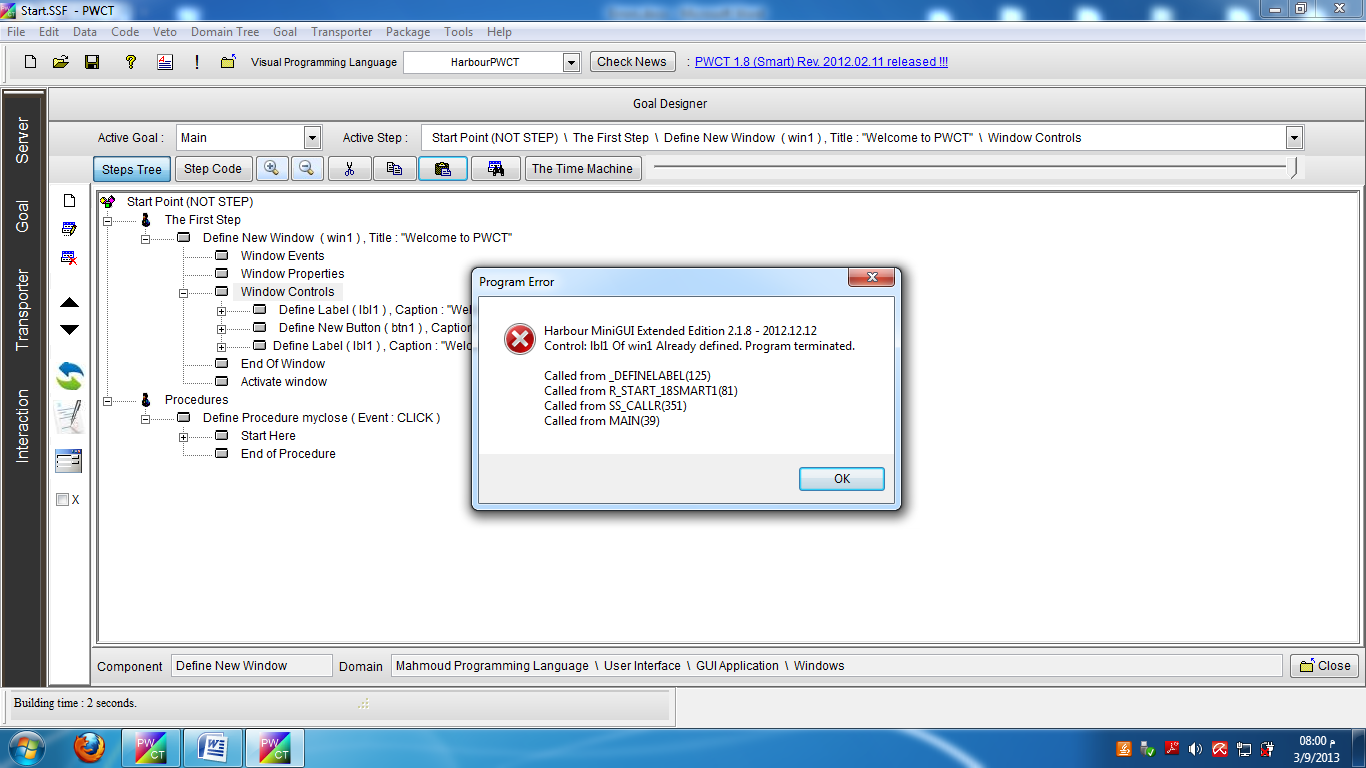
This example uses the component (Generate class data) to generate a step in a bad position



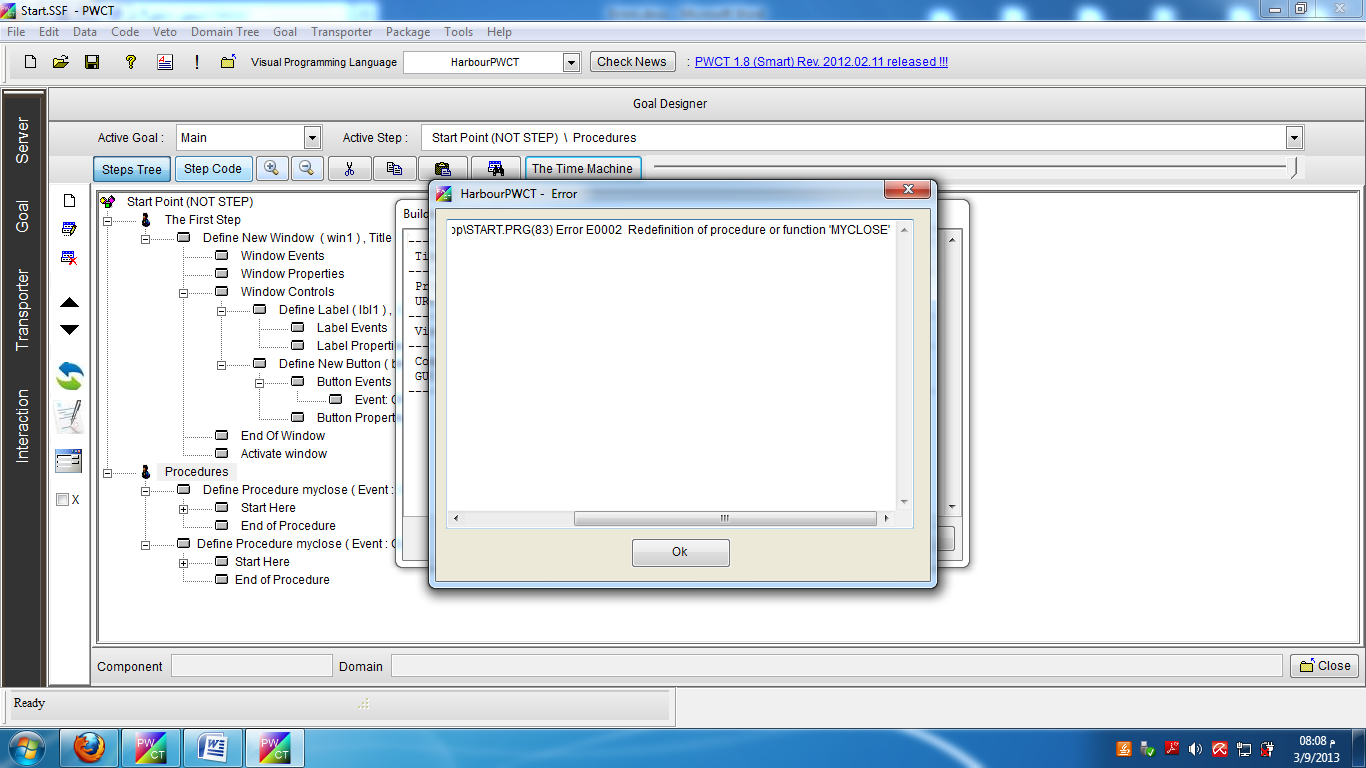
Take care when the parent step is a step created by the user, because here we need to check the parent step that are generated (not created)

1. **Copy and paste step without changing the name**

In this example LBL1 is copied and pasted as LBL1 without changing the control name



Copy and paste steps create a procedure



Here we have to

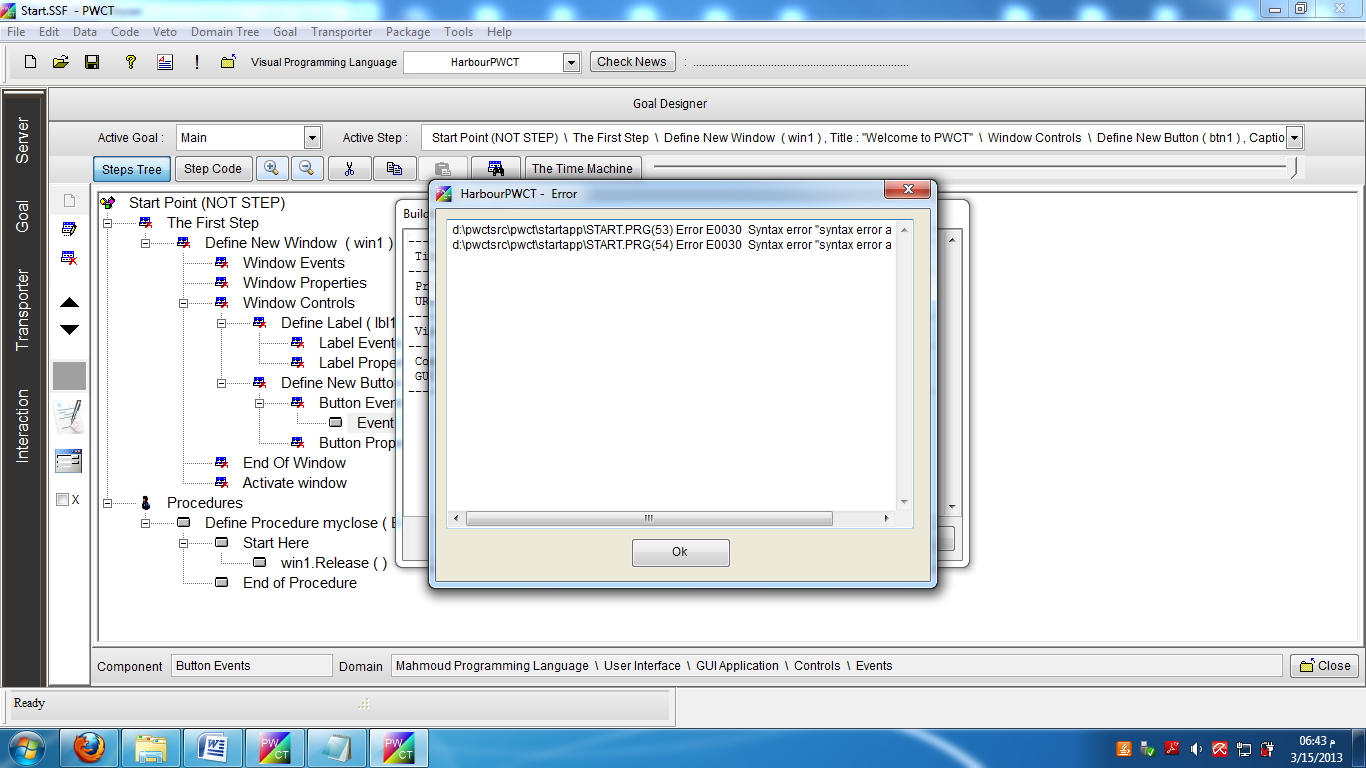
1 – Prevent name duplication

2 – Provide auto naming

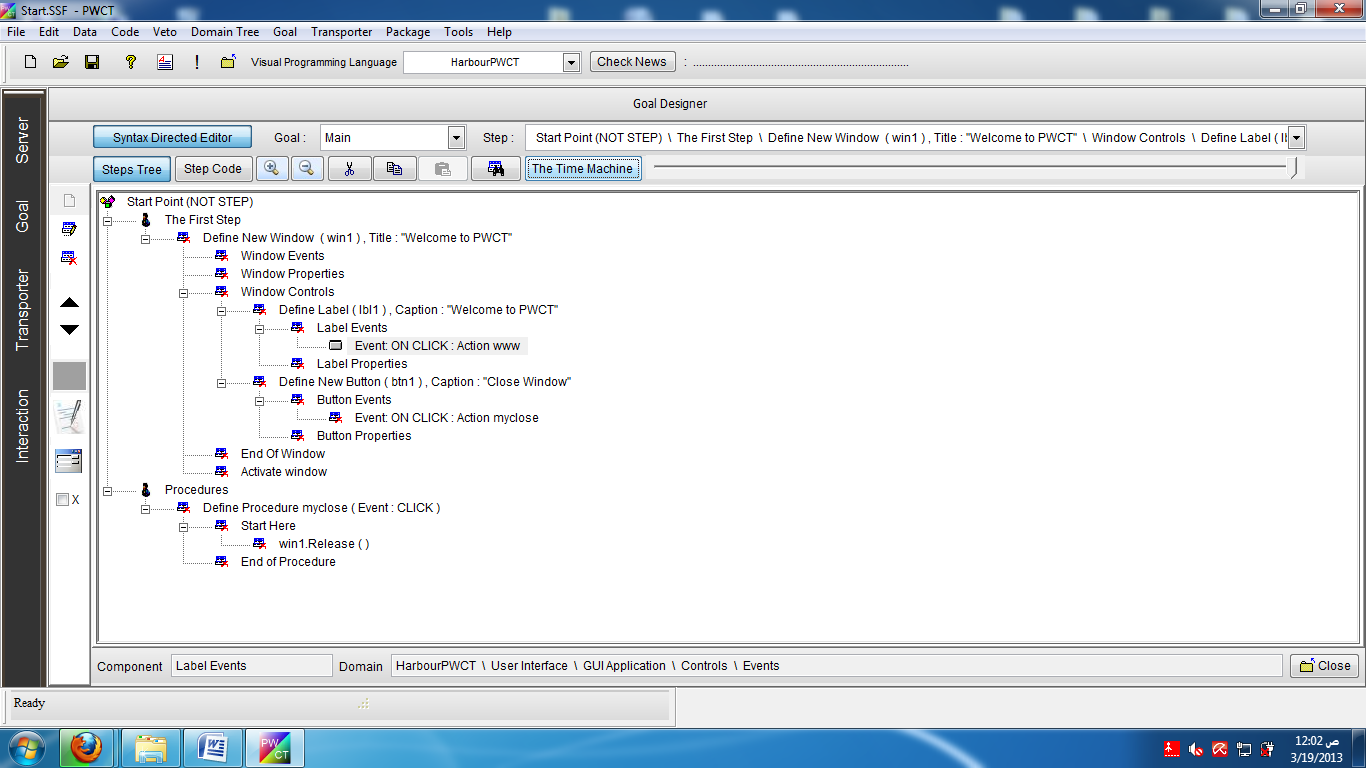
1. – Take care with the scope because method name duplication are only inside the same class
2. **Bad data entered through interaction pages**

* Variable not defined/used (error in variable name)
* Procedure not defined/used (error in procedure name)
* Method not defined/used
* Attribute not defined/used
* Control not defined/used
* Object not defined/used
* Class not defined/used
* Error in database table name
* Error in field name
* Error in expression (operators) ex: procedure name with () while it’s not required

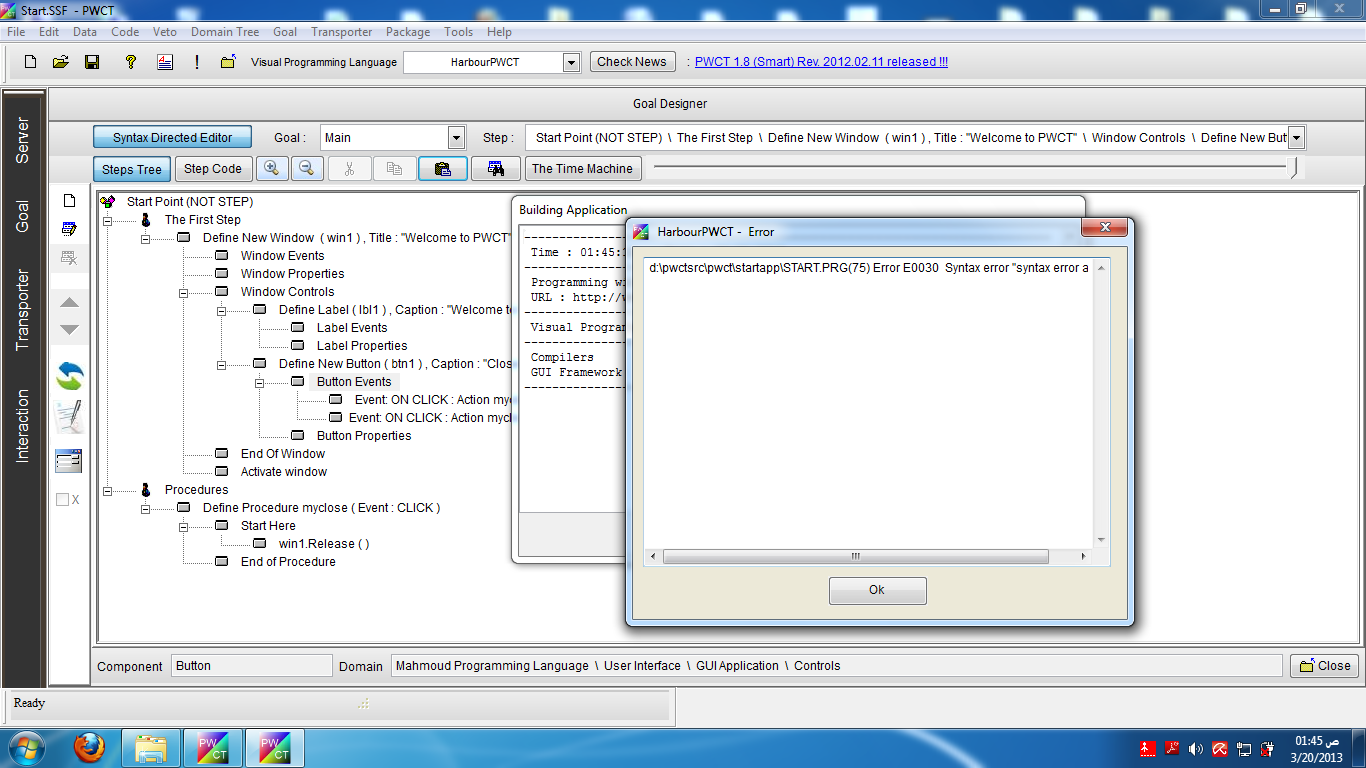
1. **Ignore parent step (ex: Define Class) without ignoring Childs (ex: declare class data)**
2. **Enabling Steps Without Enabling The correct Parent steps**



1. **Generate step inside disabled component**



1. **Event Duplication**

****

The Step (Event On Click) repeated two times

**Errors Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Error | Solution | Status |
| 1 | Change the order of the generated step  (Move UP – Move Down) | 1 | Solved |
| 2 | Ignore/Cut/Delete generated step | 1 | Solved |
| 3 | Copy and Paste generated step in bad place | 3 | Solved |
| 4 | Generate step in a bad position (using interaction) | 3 | Solved |
| 5 | Copy and paste step without changing the name | 6 |  |
| 6 | Bad data entered through interaction pages | 4 + 5 |  |
| 7 | Ignore parent step (ex: Define Class) without ignoring Childs (ex: declare class data) | 3 +2 | Solved |
| 8 | Enabling Steps Without Enabling The correct Parent steps | 3 | Solved |
| 9 | Generate Steps inside disabled component | 7 | Solved |
| 10 | Event Duplicaiton |  |  |

**Solutions**

|  |  |  |
| --- | --- | --- |
| Index | Description | Status |
| 1 | Don’t allow manipulation for a sub generated step (Step interaction Num != 1) | Done |
| 2 | Operation done for all generated step related to the same interaction | Done |
| 3 | Rules for composition | Done |
| 4 | Expression/Source File Revision Scripts |  |
| 5 | Dynamic lists (Variables/Classes/Procedures/Attributes) |  |
| 6 | Forcing Autonaming Rules |  |
| 7 | Don’t allow interaction/paste from/into a disabled component | Done |

**Actions**

|  |  |  |
| --- | --- | --- |
| Index | Action | Solutions |
| 1 | Move Up | 1 |
| 2 | Move Down | 1 |
| 3 | Cut | 1 |
| 4 | Copy | 1 |
| 5 | Paste | 3 + 6 + 7 |
| 6 | Ignore | 1 + 2 + 3 |
| 7 | Delete | 1 + 2 |
| 8 | Interact | 3 + 5 + 6 + 7 |

**How to implement the solutions**

1. **Don’t allow manipulation for a sub generated step (Step interaction ID != 1)**

Create a class called (AvoidErrors)

Contains a method (CheckGeneratedStep)

Parameters : Goal Designer Window

Check : Steps Tree, Active Step ID

IF Step Interaction ID != 1

MoveUpBtn.enabled = false

MoveDownBtn.enabled = false

CutBtn.Enabled = False

CopyBtn.Enabled = False

Else

MoveUpBtn.enabled = True

MoveDownBtn.enabled = True

CutBtn.Enabled = True

CopyBtn.Enabled = True

Endif