**FILE: ParentForm.h**  
 ***// forward declaration is needed so that the parent form knows that there is a type called ChildForm.*class ChildForm;  
  
public ref class ParentForm : public System::Windows::Forms::Form  
{  
public: ParentForm(ChildForm^ form) { m\_child = form; }  
  
public: void CallChildMethod();  
public: void MyMethod() { ... }  
  
private:  
ChildForm^ m\_child;  
};**  
  
  
  
  
**FILE: ChildForm.h**  
  
 ***// forward declaration is needed so that the child form knows that there is a type called ParentForm.*class ParentForm;  
  
public ref class ChildForm : public System::Windows::Forms::Form  
{  
public: ChildForm(ParentForm^ form) { m\_parent = form; }  
  
public: void CallParentMethod();  
public: void MyMethod() { ... }  
  
private:  
ParentForm^ m\_parent;  
};**  
  
Now, in your the ParentForm.cpp or ChildForm.cpp you can call a method of child in the parent or vice versa by just doing:  
  
**FILE ParentForm.cpp**  
**#include "ParentForm.h"  
#include "ChildForm.h" *// this include is needed so that the compiler knows about the ChildForm's definition*  
  
void ParentForm::CallChildMethod()  
{  
m\_child->MyMethod();  
}**  
or  
  
**FILE ChildForm.cpp  
  
#include "ChildForm.h"  
#include "ParentForm.h" *// this include is needed so that the compiler knows about the ChildForm's definition*  
  
void ChildForm::CallParentMethod()  
{  
m\_parent->MyMethod();  
}**