Creating Multilanguage Applications Translated by Users.

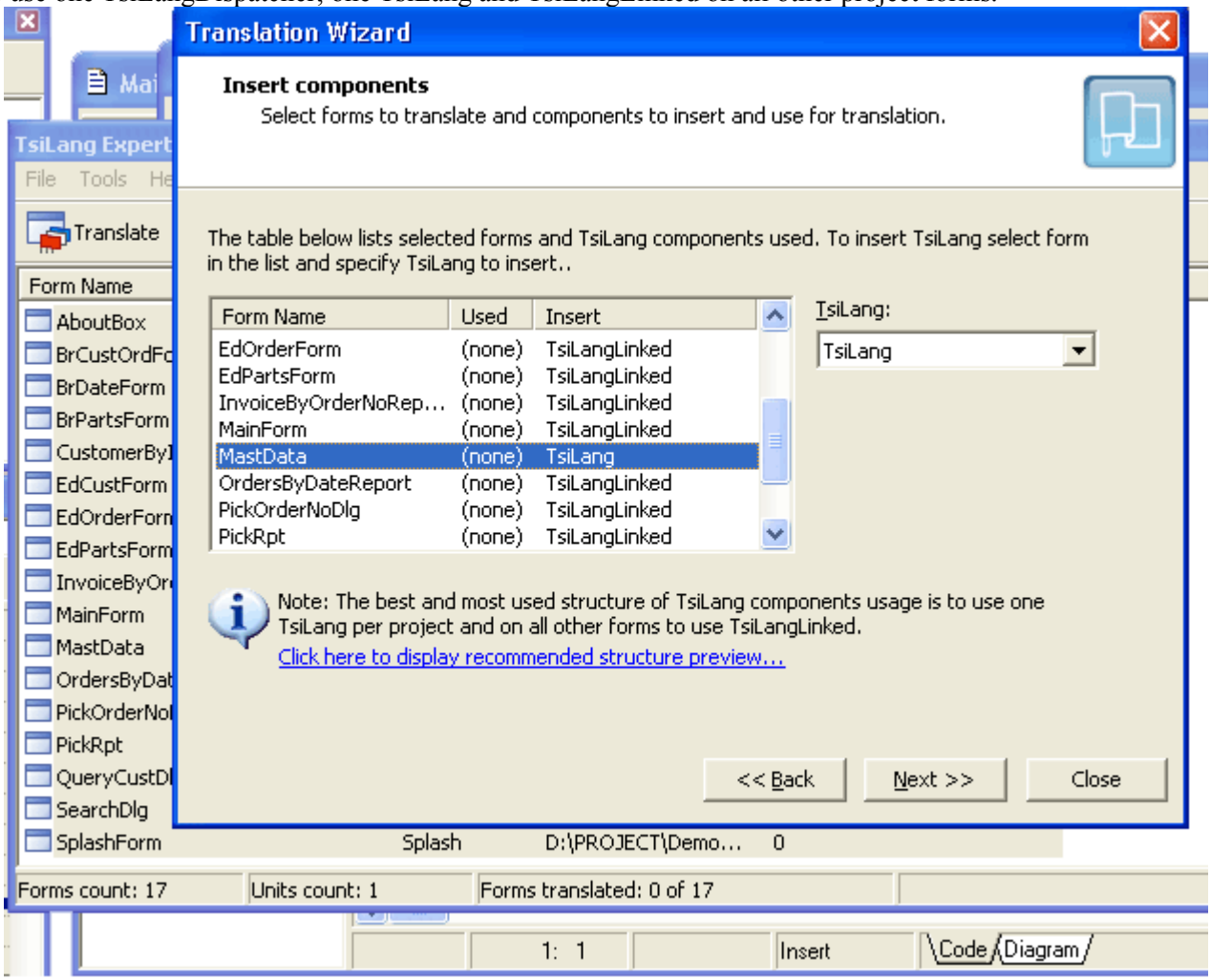
The problem of creating multilanguage applications that allow users to enter translations and even add new language without recompiling is easier than it might look at first glance. Of course, in case of TsiLang Components Suite usage to create multilingual software.

Let's describe the solution for this problem using sample project MastApp from Demos Delphi sub-folder.

1. Translating the project.

We don't describe the translation process detailed assuming you're already familiar with this part of TsiLang Components Suite. We just mention the components used for translation.

We use one TsiLangDispatcher; one TsiLang and TsiLangLinked on all other project forms.



Picture 1: Translation Wizard

Place TsiLang and TsiLangDispatcher onto MastData unit since it is auto-created and used by all project forms.

2. Creating translations file.

We can create external translations file either after providing the translations or even without translations but having just English terms. We use ***File | Save/Load Translations|Save Project*** from TsiLang Expert menu (TsiLang Expert is available under Tools Delphi menu). It is recommended to use SIB files to store translations as

SIB files are much faster than SIL files. But you may also use SIL files as they store data in the simple ASCII format, which could be edited by any text editor.

3. Using external translations file and dynamic update of available languages.

There are two ways:

1. To define the file name in FileName property of TsiLangDispatcher
2. To check the existence of the translation file into specified location in code and to load it manually.

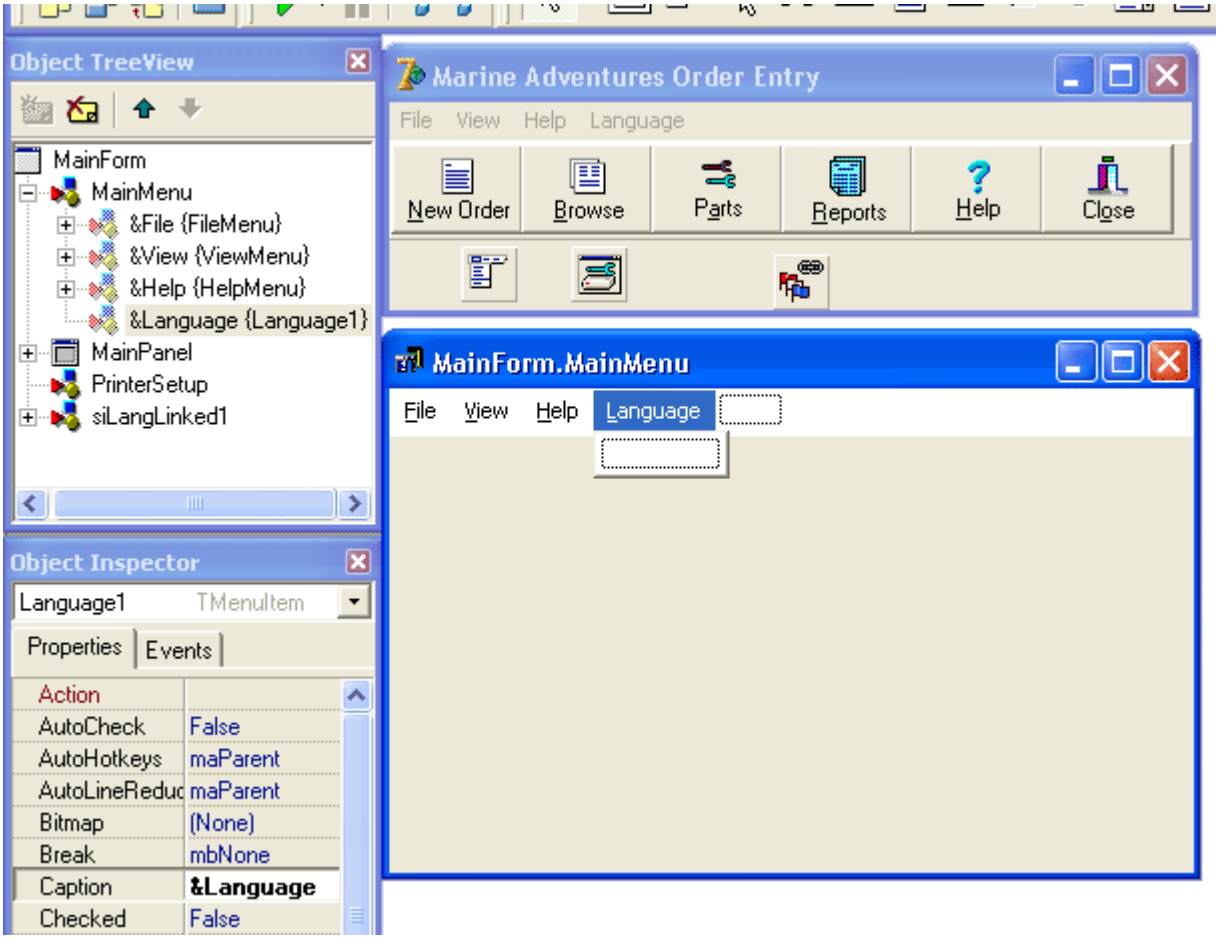
In the first case the dispatcher will automatically check the availability of the file defined in FileName property and load it into all project forms. To implement the second way you can use the following sample code:

```
procedure TMastData.DataModuleCreate(Sender: TObject);
var
  sOurSibFile: string;
begin
 // determine the file name
 sOurSibFile := ExtractFilePath(Application.ExeName) + 'TheNameOfYourFile.sib';
 // checking the existence of file
 if FileExists(sOurSibFile) then
 begin
 // set the property value
 // this will automatically load forms created later
  siLangDispatcher1.FileName := sOurSibFile;
 // load translations into already created forms
  siLangDispatcher1.LoadAllFromFile(sOurSibFile);
 end;
end;
```

Select any way above depending on your own preferences.

To dynamically display the available languages:

1. Add new top level menu item to the application's main menu.



Picture 2: Creating the languages menu

2. Add the code that will dynamically adjust and create menu items according to the available languages. This code could be added to the main form OnShow event:

```
// changing active language upon menu item click
procedure TMainForm.LanguageMenuItemClick(Sender: TObject);
begin
  MastData.siLangDispatcher1.ActiveLanguage := TMenuItem(Sender).Tag;
  TMenuItem(Sender).Checked := True;
end;

procedure TMainForm.FormShow(Sender: TObject);
var
   MenuItem: TMenuItem;
   I: Integer;
begin
  for I := 1 to MastData.siLangDispatcher1.NumOfLanguages do
  begin
    MenuItem := TMenuItem.Create(Self);
    // set Caption
    MenuItem.Caption := MastData.siLangDispatcher1.LangNames[I - 1];
    // set Tag property for easier language switching
    MenuItem.Tag := I;
    MenuItem.RadioItem := True;
    // display current language
    MenuItem.Checked := I = MastData.siLangDispatcher1.ActiveLanguage;
    // set event handler
    MenuItem.OnClick := LanguageMenuItemClick;
    // add menu item
    Language1.Add(MenuItem);
  end;
end;
```

Please note! The code of language updating and displaying must be executed after loading of the translations file.

4. Conclusion.

Run the project and “that's all”! The task looked complicated at first glance was solved in few minutes using TsiLang Components Suite. Users will be able to translate your applications without recompiling and even without your assistance. You can offer your users to use SIL Editor, which could be downloaded from http://www.sicomponents.com/download.html

TsiLang Components Suite home page: http://www.tsilang.com

Latest version can be downloaded from: http://www.tsilang.com/download.html