Для реализации произвольной логики работы инсталлятора в Inno Setup предусмотрена секция **[Code]**. В пределах этой секции размещается код реализующих логику функций на языке Pascal. Содежимое этой секции мы вынесем в отдельный файл **dotnet.pas** и включим в основной скрипт дерективой #include;------------------------------------------------------------------------------  
*;   Секция кода включенная из отдельного файла*  
*;------------------------------------------------------------------------------*  
[Code]  
**#include "dotnet.pas"**

Хотя можно набить код и непосредственно в секции **[Code]**. Надо помнить, что внутри этой секции используется синтаксис Pascal, и комментарии предваряются последовательностью "//" вместо используемой в основной части скрипта точки с запятой.

Напишем функцию, определяющую наличие в системе нужной версии .NET  
  
//-----------------------------------------------------------------------------  
*//  Проверка наличия нужного фреймворка*  
*//-----------------------------------------------------------------------------*  
**function** **IsDotNetDetected**(version: **string**; release: cardinal): boolean;  
  
**var**   
    reg\_key: **string**; *// Просматриваемый подраздел системного реестра*  
    success: boolean; *// Флаг наличия запрашиваемой версии .NET*  
    release45: cardinal; *// Номер релиза для версии 4.5.x*  
    key\_value: cardinal; *// Прочитанное из реестра значение ключа*  
    sub\_key: **string**;  
  
**begin**  
  
    success := false;  
    reg\_key := 'SOFTWARE\Microsoft\NET Framework Setup\NDP\';

*// Вресия 3.0*  
    **if** Pos('v3.0', version) = 1 **then**  
      **begin**  
          sub\_key := 'v3.0';  
          reg\_key := reg\_key + sub\_key;  
          success := RegQueryDWordValue(HKLM, reg\_key, 'InstallSuccess', key\_value);  
          success := success **and** (key\_value = 1);  
      **end**;  
  
    *// Вресия 3.5*  
    **if** Pos('v3.5', version) = 1 **then**  
      **begin**  
          sub\_key := 'v3.5';  
          reg\_key := reg\_key + sub\_key;  
          success := RegQueryDWordValue(HKLM, reg\_key, 'Install', key\_value);  
          success := success **and** (key\_value = 1);

**end**;   
  
     *// Вресия 4.0 клиентский профиль*   
     **if** Pos('v4.0 Client Profile', version) = 1 **then**   
      **begin**   
          sub\_key := 'v4\Client';   
          reg\_key := reg\_key + sub\_key;   
          success := RegQueryDWordValue(HKLM, reg\_key, 'Install', key\_value);   
          success := success **and** (key\_value = 1);   
      **end**;   
  
     *// Вресия 4.0 расширенный профиль*   
     **if** Pos('v4.0 Full Profile', version) = 1 **then**   
      **begin**   
          sub\_key := 'v4\Full';   
          reg\_key := reg\_key + sub\_key;   
          success := RegQueryDWordValue(HKLM, reg\_key, 'Install', key\_value);   
          success := success **and** (key\_value = 1);   
      **end**;   
  
     *// Вресия 4.5*   
     **if** Pos('v4.5', version) = 1 **then**   
      **begin**   
          sub\_key := 'v4\Full';   
          reg\_key := reg\_key + sub\_key;   
          success := RegQueryDWordValue(HKLM, reg\_key, 'Release', release45);   
          success := success **and** (release45 >= release);   
      **end**;   
           
    result := success;   
  
end;

Не смотря на обилие кода, логика его работы достаточно проста — в зависимости от значения параметра version с помощью функции RegQueryDWordValue(...) читается значение соответствующего ключа реестра и сравнивается с требуемым значением (смотрим таблицу 1). Для версии 4.5 дополнительно передаем номер релиза в параметре release.  
  
Для того приложения, которое мы будем развертывать, нужна вполне определенная версия .NET, поэтому напишем функцию-обертку для её определения в целевой системе  
  
//-----------------------------------------------------------------------------  
*//  Функция-обертка для детектирования конкретной нужной нам версии*  
*//-----------------------------------------------------------------------------*  
**function** **IsRequiredDotNetDetected**(): boolean;  
**begin**  
    **result** := IsDotNetDetected('v4.0 Full Profile', 0);  
**end**;

Для того, чтобы перед началом установки проверить наличие фреймворка и сообщить пользователю о предпринимаемых действиях используем Callback-функцию InitializeSetup()  
  
//-----------------------------------------------------------------------------  
*//    Callback-функция, вызываемая при инициализации установки*  
*//-----------------------------------------------------------------------------*  
**function** **InitializeSetup**(): boolean;  
**begin**  
  
  *// Если нет тербуемой версии .NET выводим сообщение о том, что инсталлятор*  
  *// попытается установить её на данный компьютер*  
  **if** **not** IsDotNetDetected('v4.0 Full Profile', 0) **then**  
    **begin**  
      MsgBox('{#Name} requires Microsoft .NET Framework 4.0 Full Profile.'#13#13  
             'The installer will attempt to install it', mbInformation, MB\_OK);  
    **end**;     
  
  **result** := **true**;  
**end**;

Теперь в секцию **[Files]** добавим запись о том, где компилятор должен взять дистрибутив .NET, куда инсталлятор он должен её распаковать, и при каких условиях следует выполнять распаковку  
  
; .NET Framework 4.0  
Source: "E:\install\dotNetFx40\_Full\_x86\_x64.exe"; DestDir: "{tmp}"; Flags: deleteafterinstall; Check: not IsRequiredDotNetDetected  
  
Во флагах скажем, что надо удалить дистрибутив .NET после установки (deleteafterinstall). Условие, при котором требуется распаковка задаем опцией **Check**, где вызываем функцию IsRequiredDotNetDetected(), выполняя распаковку ели она вернет false.

Сам запуск инсталляции фрейворка можно выполнить после установки основной программы, поэтому включаем в скрипт секцию **[Run]**, в которой указывается, что необходимо запускать по окончании установки;------------------------------------------------------------------------------  
*;   Секция кода включенная из отдельного файла*  
*;------------------------------------------------------------------------------*  
[Code]  
**#include "dotnet.pas"**  
  
[**Run**]  
*;------------------------------------------------------------------------------*  
*;   Секция запуска после инсталляции*  
*;------------------------------------------------------------------------------*  
Filename: {tmp}\dotNetFx40\_Full\_x86\_x64.exe*; Parameters: "/q:a /c:""install /l /q"""; Check: not IsRequiredDotNetDetected; StatusMsg: Microsoft Framework 4.0 is installed. Please wait...*

Снова компилируем наш проект. Теперь, при запуске на системе без Framework инсталлятор выдаст сообщение, в котором требуется .NET Framework 4.0 и для нас его установят.

При распаковке мы заметим, что дополнительно распаковывает дистрибутив .NET во временную папку. Затем процесс переключается на установку .NET  
Майкрософт просит нас принять лицензию на у*становку .NET*