**среда, 18 февраля 2009 г.**

**Debug Visual C++ Redistributable Package своими руками**

Довольно частое явление когда у клиента девственно чистый компьютер - в смысле без Visual Studio, а вам необходимо запустить debug версию вашего приложения на C++ для отладки например. Если ваша сборка использует динамическую линковку системных DLL, то почти наверняка ваше приложение не загрузится. Скорее всего причина кроется в том, что у клиента не установлен Visual C++ Redistributable Package. На сайте Microsoft лежат такие package, но только для release, для debug вы их там не найдете:

* [Microsoft Visual C++ 2005 Redistributable Package (x86)](http://www.microsoft.com/downloads/details.aspx?FamilyId=32BC1BEE-A3F9-4C13-9C99-220B62A191EE&displaylang=en)
* [Microsoft Visual C++ 2005 Redistributable Package (x64)](http://www.microsoft.com/downloads/details.aspx?familyid=90548130-4468-4BBC-9673-D6ACABD5D13B&displaylang=en)
* [Microsoft Visual C++ 2005 SP1 Redistributable Package (x86)](http://www.microsoft.com/downloads/details.aspx?familyid=200B2FD9-AE1A-4A14-984D-389C36F85647&displaylang=en)
* [Microsoft Visual C++ 2005 SP1 Redistributable Package (x64)](http://www.microsoft.com/downloads/details.aspx?familyid=EB4EBE2D-33C0-4A47-9DD4-B9A6D7BD44DA&displaylang=en)
* [Microsoft Visual C++ 2008 Redistributable Package (x86)](http://www.microsoft.com/downloads/details.aspx?familyid=9B2DA534-3E03-4391-8A4D-074B9F2BC1BF&displaylang=en)
* [Microsoft Visual C++ 2008 Redistributable Package (x64)](http://www.microsoft.com/downloads/details.aspx?familyid=BD2A6171-E2D6-4230-B809-9A8D7548C1B6&displaylang=en)
* [Microsoft Visual C++ 2008 SP1 Redistributable Package (x86)](http://www.microsoft.com/downloads/details.aspx?familyid=A5C84275-3B97-4AB7-A40D-3802B2AF5FC2&displaylang=en)
* [Microsoft Visual C++ 2008 SP1 Redistributable Package (x64)](http://www.microsoft.com/downloads/details.aspx?familyid=BA9257CA-337F-4B40-8C14-157CFDFFEE4E&displaylang=en)

Так как же быть? Создадим такой package сами. Открываем Visual Studio, создаем новый solution. В диалоге выбора типа проекта выбираем Other Project Typoes|Setup Project. Потом добавляем необходимые merge module в проект и компилируем. Учтите, что для каждой платформы будь то x86 или x64 необходимо создать свой project в solution. Не забываем добавлять соответствующие policy. В х64 вариант обьязательно добавляем merge module для x86. Ниже приведен список необходимых merge module для:

Visual Studion 2005 (DebugCRT x86)

* Microsoft\_VC80\_DebugCRT\_x86.msm
* policy\_8\_0\_Microsoft\_VC80\_DebugCRT\_x86.msm

Visual Studion 2005 (DebugCRT x64)

* Microsoft\_VC80\_DebugCRT\_x86.msm
* Microsoft\_VC80\_DebugCRT\_x86\_x64.msm
* policy\_8\_0\_Microsoft\_VC80\_DebugCRT\_x86.msm
* policy\_8\_0\_Microsoft\_VC80\_DebugCRT\_x86\_x64.msm

Visual Studion 2008 (DebugCRT x86)

* Microsoft\_VC90\_DebugCRT\_x86.msm
* policy\_9\_0\_Microsoft\_VC90\_DebugCRT\_x86.msm

Visual Studion 2008 (DebugCRT x64)

* Microsoft\_VC90\_DebugCRT\_x86.msm
* Microsoft\_VC90\_DebugCRT\_x86\_x64.msm
* policy\_9\_0\_Microsoft\_VC90\_DebugCRT\_x86.msm
* policy\_9\_0\_Microsoft\_VC90\_DebugCRT\_x86\_x64.msm

Если вы используете MFC, то дополнительно добовляете MFC merge module в соответствующий проект по аналогии - имена должны содержать DebugMFC.

Остается только скомпилировать ваш solution и в результате вы получите msi для установки на компьютере клиента. Рекомендую также правильно и солидно назвать исталляцию и сменить имя msi файла, чтобы на стороне клиента это все не выглядело поделкой на коленке.

P.S. Стиль когда нужные сисетмные DLL кладутся в проект рядом с вашим приложением на мой взгляд является анахронизмом, хоть и работает. С другой стороны мне несложно придумать конфигурацию, когда такой фокус уже не прокатит. Самый простой пример: ваше приложение для x86 и для x64 лежат в одном каталоге.

P.P.S Все доступные merge module хрянятся в "%CommonProgramFiles%\Merge Modules".