**Инструкция для переноса проекта из среды Borland C++ Builder 6**

**в среду Builder XE8.**

(дополняемая)

1. Открытие проекта:

* открыть файл проекта (.bpr) в среде Builder XE8;
* при возникновении ошибки «Класс не зарегистрирован ClassID {№ ID F8FEDD39-E3CE-4B8D-A657-9CA24686881F}» выйти и запустить файл «Builder 2007 CLSID F8FEDD39-E3CE-4B8D-A657-9CA24686881F.reg»;
* повторно открыть файл проекта (.bpr) в среде Builder XE8;
* при возникновении ошибки «Класс не зарегистрирован ClassID {№ ID 1D7A3F34-1338-420D-932F-D16FBA1858A2}» выйти и запустить файл «Builder 2007 CLSID 1D7A3F34-1338-420D-932F-D16FBA1858A2.reg»;
* повторно открыть файл проекта (.bpr) в среде Builder XE8;
* в случае повторных сообщениях об ошибках следует произвести поиск по ID соответствующих файлов на других машинах в реестре или в интернете;
* при корректном открытии проекта создаётся новый файл проекта (.cbproj), которым впоследствии и необходимо пользоваться.

1. Редактирование проекта:

* Параметр «ProjectOptions…» ─> «C++(Shared options)» ─> «\_TCHAR maps to» должен быть «wchar\_t»;
* В «ProjectOptions…» ─> «Forms» проверить правильность порядка форм с указанием главной формы;
* Следует «переподключить» все библиотеки проекта посредством: «Project» ─> «Remove from project…» (исключить), «Project» ─> «Add from project…» (добавить);
* В случае, если при компиляции среда будет выдавать ошибки поиска необходимых объектных файлов (.obj), проект следует перенести в папку с путём исключающим кириллицу;
* В файле с именем проекта (.cpp) найти строчку «WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)» и заменить ее на

«WINAPI wWinMain(HINSTANCE, HINSTANCE, LPWSTR, int)»;

* Заменить «fopen()» на «\_wfopen()», «sprintf()» на «wsprintf()», «strcat()» на «wcscat()», «strcpy()» на «wcscpy()» и непосредственно перед кавычками в функциях необходимо ставить «L»;
* При вызове функции «IntToStr()», убедиться, что аргумент типа «int» (не «long);
* При использовании функции «swap» использовать «std::swap»; ✓
* Вместо <algorith.h> использовать <utility>; ✓
* Вместо <chart.hpp> использовать <VCLTee.chart.hpp>; ✓
* Вместо <series.hpp> использовать <VCLTee.series.hpp>; ✓
* Вместо <teengine.hpp> использовать <VCLTee.teengine.hpp>; ✓
* Вместо <teeprocs.hpp> использовать <VCLTee. teeprocs.hpp>; ✓
* Вместо <vcl\registry.hpp> использовать <registry.hpp>; ✓
* При создании файла, использовать для имени файла массив типа «wchar\_t» вместо «char», а так же убедиться в том, что если имя файла формируется с использованием строки, убедиться что строка типа «UnicodeString», а не «AnsiString»;
* При вызове «MessageBox», если вызов прямой (без переменных), не забывать перед кавычками ставить «L» («MessageBox(L"Текст",L"Текст",MB\_OK)»), если вызов непрямой (с использованием переменных), использовать строки типа «UnicodeString» и метод «w\_str()» («MessageBox(message.w\_str(), title,w\_str(), MB\_OK», где «message» и «title» строки типа «UnicodeString», а если вызывается «MessageBoxA», заменить на «MessageBox»;
* При использовании графического модуля «ddutil.cpp» изменить тип переменной «szBitmap» с «LPCSTR» на «LPCWSTR», так же поменять тип в заголовочном файле «ddutil.h». В «ddutil.cpp» не забыть поставить «L» перед кавычками при выводе сообщений типа «OutputDebugString(L"handle is null\n")» в телах процедур и привеcти «szBitmap» к типу «LPCSTR» в «\_lopen((LPCSTR)szBitmap…»
* Убедиться что везде в файлах использующих функции графического модуля вместо «char szBitmap[] = "ALL"» стоит «wchar\_t szBitmap[] = L"ALL"»;
* При несоответствии типов параметров можно использовать приведение типов: «(int)…», «(wchat\_t\*)…».
* Убрать код сжатия «.png» файлов в файле «DD.cpp»

Возможные ссылки для решения возникающих проблем:

<https://habrahabr.ru/post/164193/>

<http://community.embarcadero.com/blogs/entry/migrating-legacy-c-builder-apps-to-c-builder-10-seattle>

<http://www.delphimaster.ru/books/978527200384/fragment.html>

<http://stackoverflow.com/questions/9907280/c-builder-xe-cant-succesfully-convert-project-from-bcb6>

<https://youtu.be/H-KFq5M0z0A?t=2400>