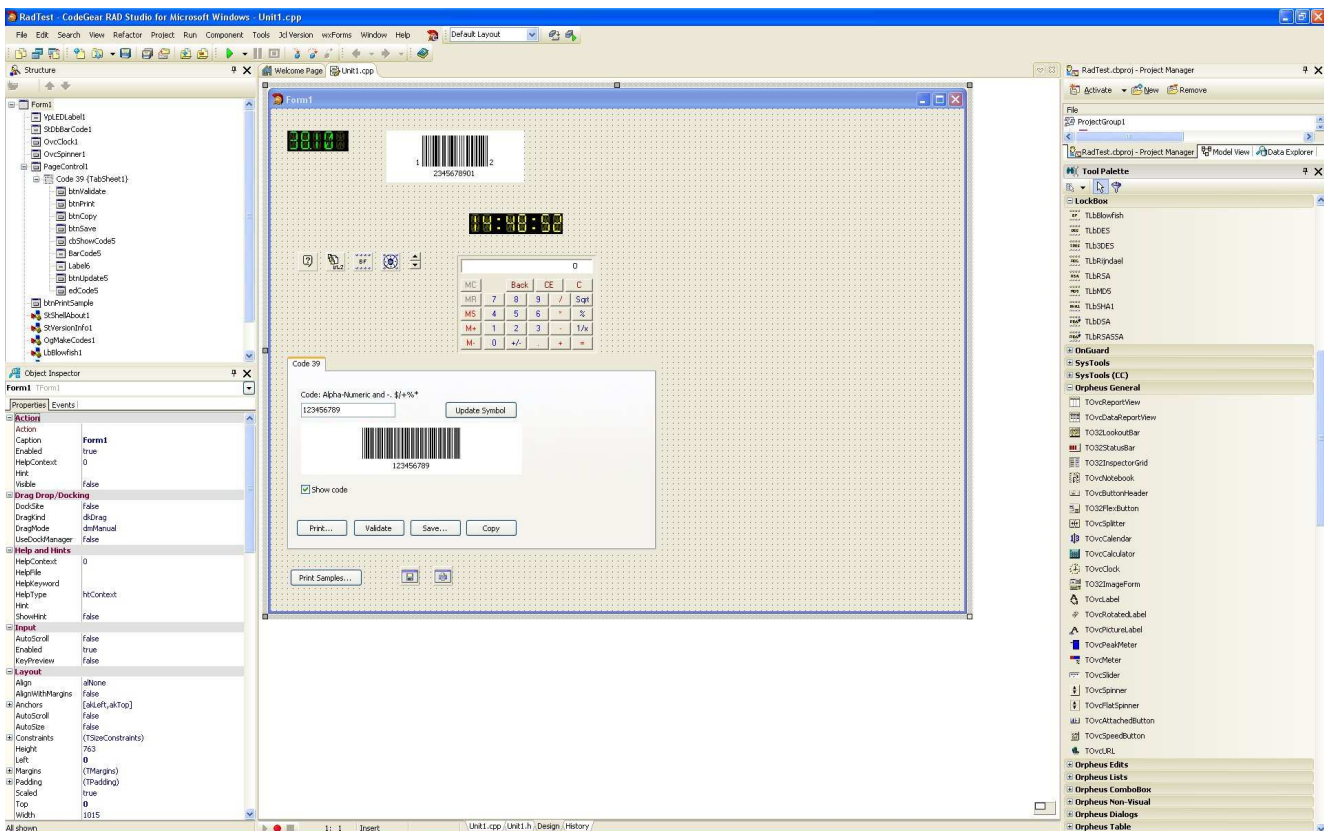


TurboPower VCL for CodeGear RAD 2007 C++ (and 2009)

If you are a Borland C++Builder 6 developer and you have been using the TurboPower Software VCL components like LockBox, OnGuard, Orpheus, Planit ShellShock, SysTools and many others in your projects and you have been hesitating to upgrade to any new version of Borland/CodeGear C++Builder then this article is for you.

I have been using Borland development tools since the late 80' and Borland C++ from the early 90'. For me Borland has always had the little something extra to get the job done fast. When Borland C++Builder came along I was not too convinced about dropping all my trusty OWL code and taking the plunge into a mixed language environment called VCL, with Delphi being the driving force behind the extremely fast, unmatched RAD development environment for Windows. I tried it and it was indeed the fastest way to create Windows applications. I was hooked. Borland C++Builder 6 was released in 2002 and was the only C++ version from Borland/CodeGear for many years. Despite being released at the time Windows XP was released, the actual VCL components did not support themes nor did they support the XP native control messages. This has led to a lot of pressure from my customers to have software versions that support native Windows XP and Windows Vista styles.

CodeGear RAD 2007 comes to the rescue.



My greatest fear of upgrading was that my many VCL packages which installed and worked fine in Borland C++Builder 6 would simply not install in the new version. The fear was unfounded.

RAD 2007 does a great job of upgrading the VCL package file to the latest version and generates the needed Delphi and C++ files, before installing the package successfully (you must always use the latest DELPHI package project file if you wish to install in all personalities). At this point it is important to mention that when you are installing VCL components you always need to start the **complete CodeGear RAD Studio** with all personalities loaded. Do not just start the C++, Delphi or Delphi .Net personalities.

Let's start off by using the TurboPower LockBox encryption VCL components as an example.

If you have downloaded the latest LockBox source files from SourceForge.net then you should be able to follow this example. In any case you can download the complete RAD 2007 ready sources from <http://www.nordenlogic.com> in the download section as well as all the other projects I have converted already.

Some info regarding RAD 2009 and a new version of LockBox can be found here: <https://forums.codegear.com/thread.jspa?messageID=69548>

I normally copy the project package files into the source directory of the TurboPower VCL component. I recommend you do the same. Before we open the most recent Delphi project files provided by TurboPower we need to add some compiler pre-compile directives to the xyz.INC file (each TurboPower VCL component has this file in the source directory). Open the file in a text editor and add the following to it:

```
{ $IFDEF VER170 }                { Delphi 2005 }
  { $DEFINE VERSION6 }           { Version 6 or greater }
  { $DEFINE VERSION7 }
  { $IFDEF BCB }
    { $DEFINE CBuilder }
    { $ObjExportAll On }
  { $ENDIF }
{ $ENDIF }

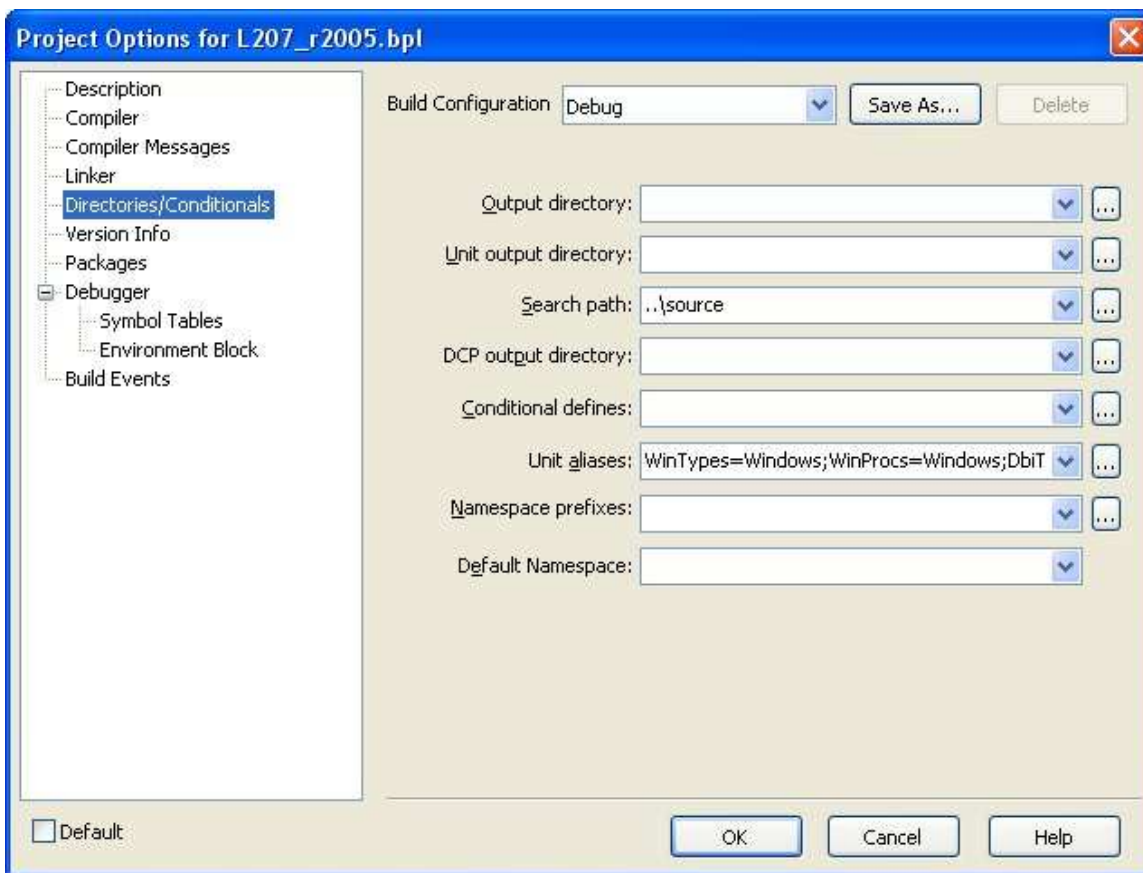
{ $IFDEF VER180 }                { Delphi 2006 }
  { $DEFINE VERSION6 }           { Version 6 or greater }
  { $DEFINE VERSION7 }
  { $IFDEF BCB }
    { $DEFINE CBuilder }
    { $ObjExportAll On }
  { $ENDIF }
{ $ENDIF }

{ $IFDEF VER185 }                { Delphi 2007 }
  { $DEFINE VERSION6 }           { Version 6 or greater }
  { $DEFINE VERSION7 }
  { $IFDEF BCB }
    { $DEFINE CBuilder }
    { $ObjExportAll On }
  { $ENDIF }
{ $ENDIF }
```

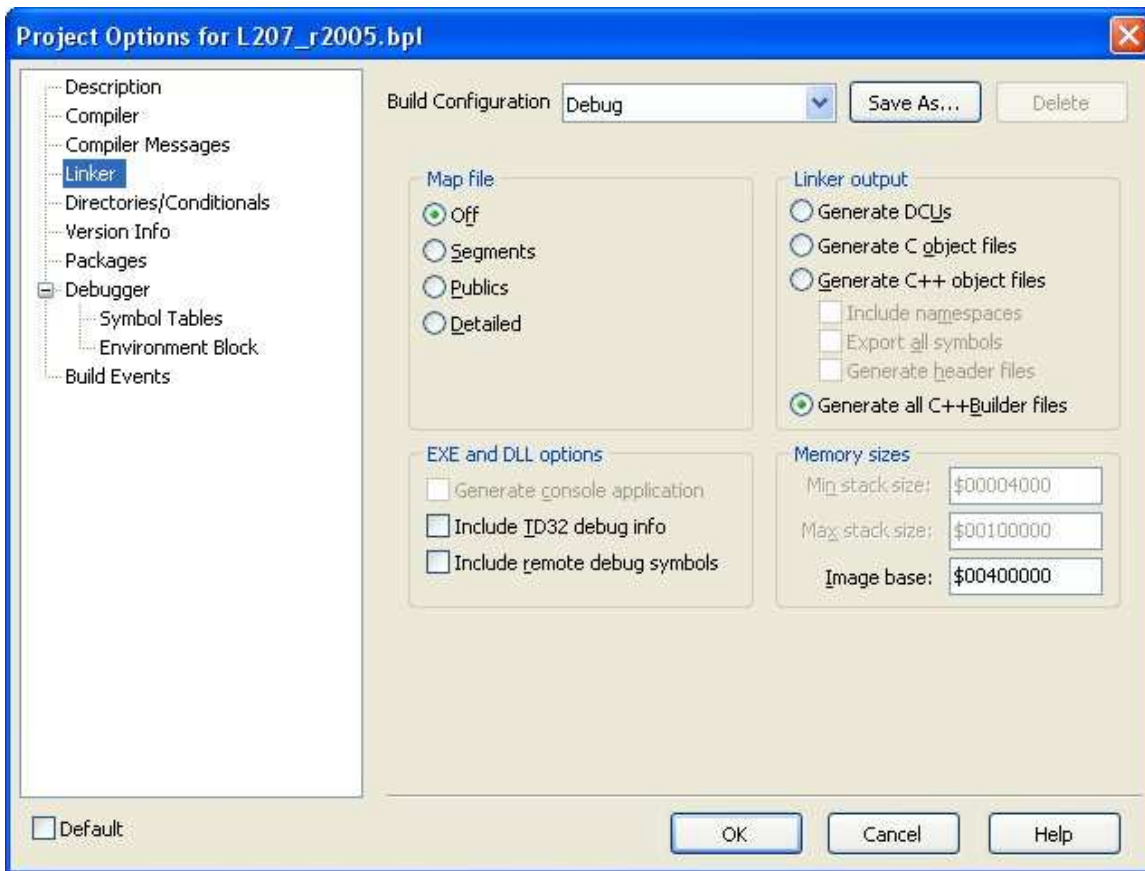
Save the file and exit your text editor.

The LockBox VCL components have run-time components and design-time components. We will first build the run-time component since most of the time design-time components have dependencies with those files. Open the file **L207_r70.dpk**. You will notice that my projects are using the file **L207_r2005.dproj**. Feel free to use those files if you prefer. They are already converted. Using the Delphi 7 project files is not any more difficult.

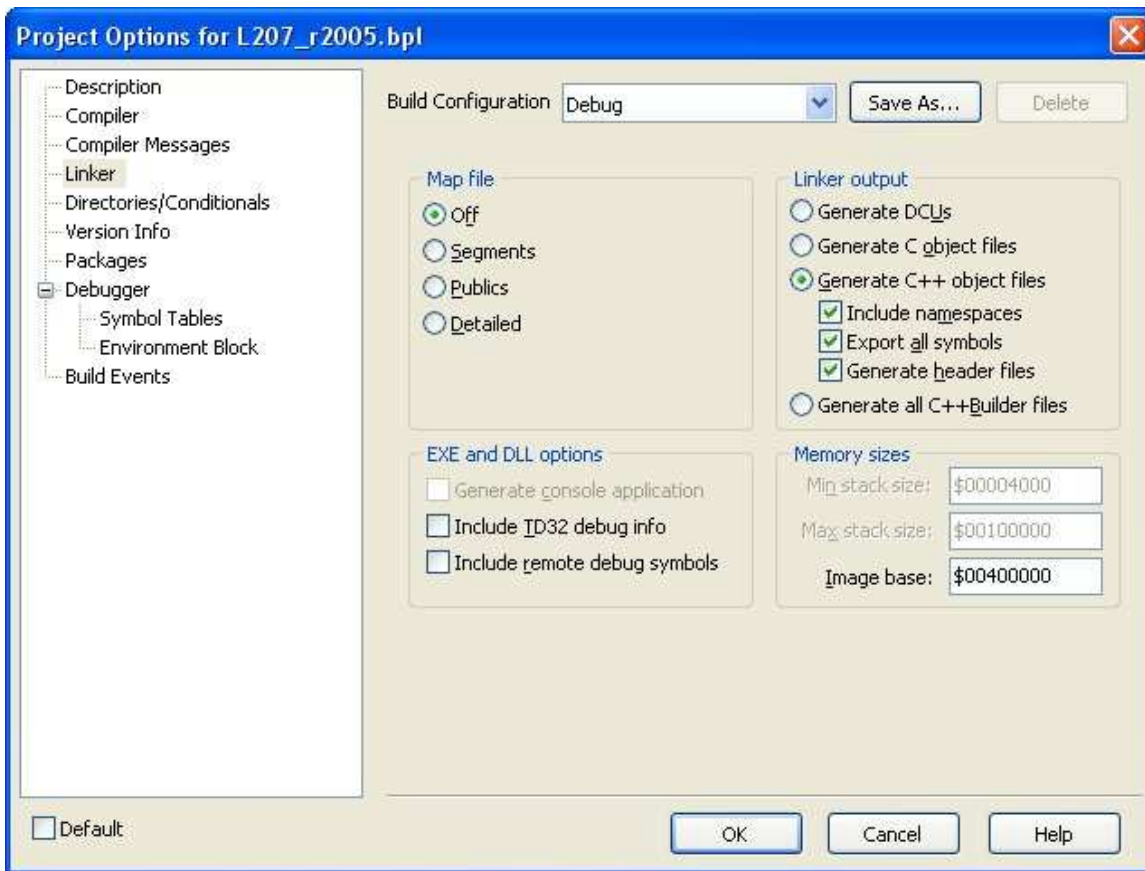
Once the file is loaded you need to set some important options! First of all make sure that the output directories are empty, so that the final files are placed to the environment default directories.



Next you need to set some linker options so that ALL Delphi and C++ files are generated. Note however that you will need to do the following several times. First you build with the linker option **Generate all C++Builder files**.



After that you need to set the options **Generate C++ object files** and tick on **Include namespaces**, **Export all symbols** and **Generate header files**.

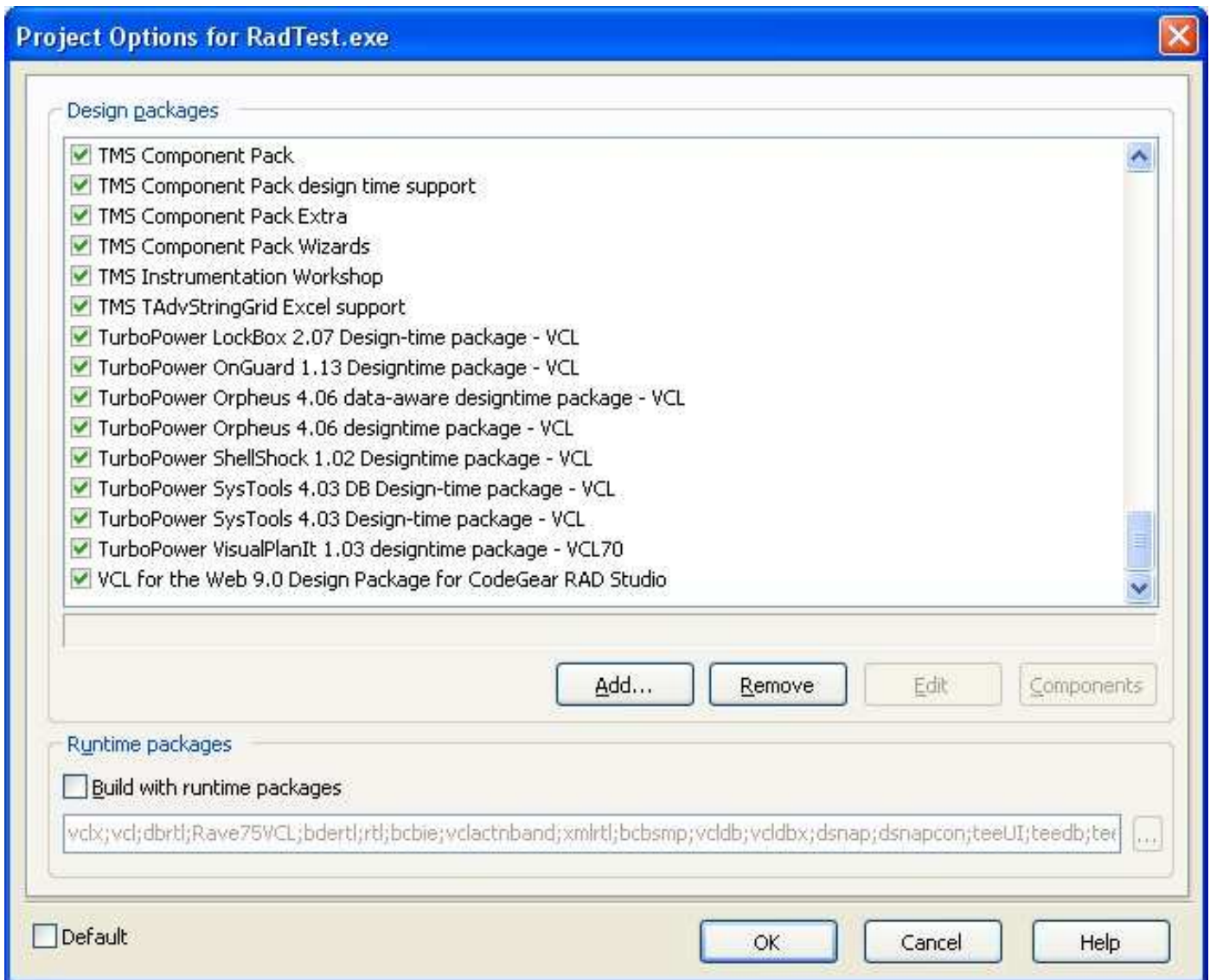


After that build once more with the linker option **Generate all C++Builder files**.

The run-time packages are now ready and you can move on to build the design-time packages. Before you do however, copy the *.bpi and *.lib files from the compile directory to the shared RAD 2007 directory called:

C:\Documents and Settings\All Users\Documents\RAD Studio\5.0\dcg

The design-time packages follow the exact same process, however, once built you right-click on the Project-file and select Install. This will install the component(s) into the RAD 2007 IDE for Delphi and C++.



This is all that is needed to get TurboPower VCL components or any older components to work with Delphi and most importantly C++ in RAD 2007.

A couple of words on porting your old projects to the new RAD 2007 environment. In most cases the direct import of the Borland C++Builder file into the RAD 2007 will not work. Not because RAD 2007 does a bad job in converting, but rather because your old project file is full of old library names which may not be available with the same name on RAD 2007.

A better way is to create a new project with RAD 2007 with the same name as the original including the same name for the main form and generate this project file in a new directory. Then add one by one your source files from you original project to this project (best method is to copy all the files you want to add to the project to the new directory). Note that you will need to add the following line to your file containing the WinMain function:

```
SetApplicationMainFormOnTaskBar(Application, true);
```

This function controls how your Windows are stacked in Windows Vista. I tend to use this function with the last value being FALSE in order to have the same behavior as in previous Windows versions.

One last note on creating a new project. You need to manually edit the project and make sure that your main form name is associated correctly. By default the project manager will assign Form1 to your main form even though your main form has probably a different form/class name. You need to edit this in several places. Simply do a Find in Files Form1 and you will find the places in the code. The project file you need to edit outside of the IDE.

If you are in need for resources or help for your software project, please don't hesitate to contact me. I am always on the lookout for interesting work big or small (Windows, Mac & Linux).

Just drop me a mail if you need help with plain C, C++Builder (VCL), Borland C++ (OWL), Qt, MFC, wxWidgets or others.

If you have questions please don't hesitate to contact me at:

tim.hamel@nordenlogic.com

The sources and this document can be found in the download section of Norden Logic Oy <http://www.nordenlogic.com>