Dave's Development Blog
Software Development using Borland / Codegear / Embarcadero RAD Studio

An Alternative Idea for Managing OTA Projects for Different Compilers

By David | March 2, 2018 0 Comment

Overview

For all my Open Tools API projects to date I've managed separate `.dpr` and `.dproj` files for each version. I currently have RAD Studio 2006 to Tokyo installed (yes, all on the same machine as I have no VM capability with my current machine) so I have an `.dpr` and `.dproj` files for each version of RAD Studio and I wondered whether there was a way to maintain just a single `.dpr` and `.dproj` pair of files and still manage to create all the DLLs (or packages) for for each version of RAD Studio.

One thing I should mention is that once I've finished with a release of an OTA Project, I compile all the different version builds using the command-line compilers and a Take Command batch file. This means that I need to maintain a `.cfg` file for each version which I managed, to date, by writing out a new `.cfg` file in the batch file. RAD Studio does not maintain your project options in both the `.dproj` and `.cfg` files as the `.cfg` files can only understand a single set of build options unlike the IDE which can understand multiple build options but since OTA projects are Win32, you only have to consider Debug or Release builds. I've chosen to produce Release builds from the command line and Debug builds from within the IDE.

$LIBSUFFIX

Initially the obvious choice for this was the `--lib-suffix` command-line parameter however this has only been available (according to the command-line help for each compiler) with Tokyo but the `{$LIBSUFFIX}` compiler directive has been available much much longer (at least back to RAD Studio 2006).

So for all my OTA plug-ins I use a `.inc` file to define compiler definitions for the different version as below (excerpt):

```
// Borland Delphi 1, Version 1
{$IFDEF VER80}
 {$DEFINE D0001}
```

```
{$ENDIF}
...
// Embarcadero Delphi 2010 (RAD Studio 7.0), Version 14, Package version 140
{$IFDEF VER210}
 {$DEFINE D0001}
 {$DEFINE D0002}
 {$DEFINE D0003}
 {$DEFINE D0004}
 {$DEFINE D0005}
 {$DEFINE D0006}
 {$DEFINE D0007}
 {$DEFINE D0008}
 {$DEFINE D2005}
 {$DEFINE D2006}
 {$DEFINE D2007}
 {$DEFINE D2009}
 {$DEFINE D2010}
{$ENDIF}
...
// Embarcadero Delphi XE10.2 Tokyo (Studio 19.0), Version 25, Package version 250
{$IFDEF VER320}
 {$DEFINE D0001}
 {$DEFINE D0002}
 {$DEFINE D0003}
 {$DEFINE D0004}
 {$DEFINE D0005}
 {$DEFINE D0006}
 {$DEFINE D0007}
 {$DEFINE D0008}
 {$DEFINE D2005}
 {$DEFINE D2006}
 {$DEFINE D2007}
 {$DEFINE D2009}
 {$DEFINE D2010}
 {$DEFINE DXE00}
 {$DEFINE DXE20}
 {$DEFINE DXE30}
 {$DEFINE DXE40}
 {$DEFINE DXE50}
 {$DEFINE DXE60}
 {$DEFINE DXE70}
 {$DEFINE DXE80}
 {$DEFINE DXE100}
 {$DEFINE DXE101}
 {$DEFINE DXE102}
```

```
{$ENDIF}

{$IFNDEF D0001}
 {$MESSAGE ERROR 'The Condition Definitions need to be updated!!!!!'}
{$ENDIF}
```

So using the above compiler definitions I created the following `.inc` file for the library suffixes:

```
{$IFDEF DXE102}
  {$LIBSUFFIX 'XE102'}
{$ELSE}
  {$IFDEF DXE101}
    {$LIBSUFFIX 'XE101'}
  {$ELSE}
    {$IFDEF DXE100}
      {$LIBSUFFIX 'XE10'}
    {$ELSE}
      {$IFDEF DXE80}
        {$LIBSUFFIX 'XE8'}
      {$ELSE}
        {$IFDEF DXE70}
          {$LIBSUFFIX 'XE7'}
        {$ELSE}
          {$IFDEF DXE60}
            {$LIBSUFFIX 'XE6'}
          {$ELSE}
            {$IFDEF DXE50}
              {$LIBSUFFIX 'XE5'}
            {$ELSE}
              {$IFDEF DXE40}
                {$LIBSUFFIX 'XE4'}
              {$ELSE}
                {$IFDEF DXE30}
                  {$LIBSUFFIX 'XE3'}
                {$ELSE}
                  {$IFDEF DXE20}
                    {$LIBSUFFIX 'XE2'}
                  {$ELSE}
                    {$IFDEF DXE00}
                      {$LIBSUFFIX 'XE'}
                    {$ELSE}
                      {$IFDEF D2010}
                        {$LIBSUFFIX '2010'}
                      {$ELSE}
                        {$IFDEF D2009}
```

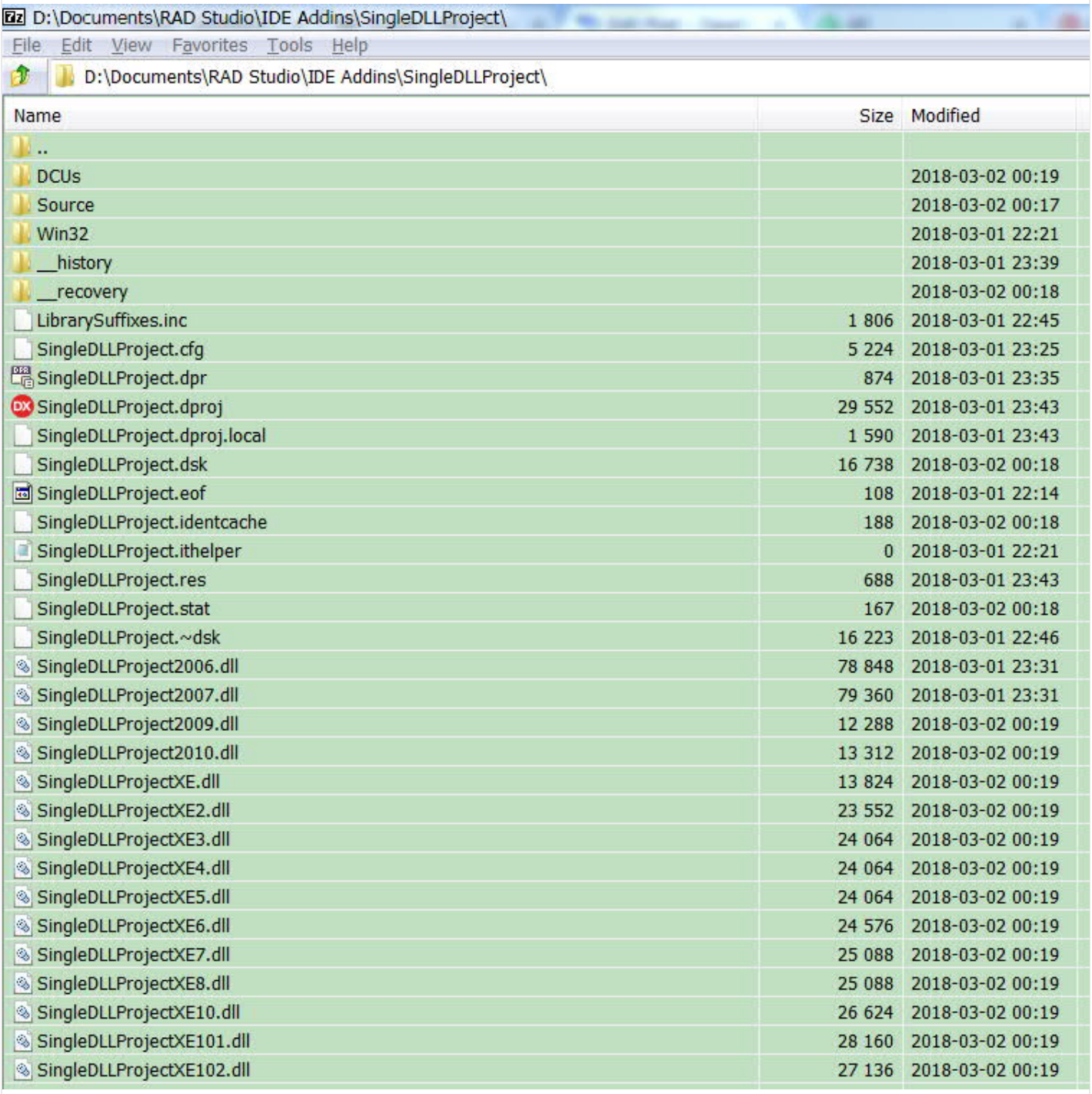
```
                          {$LIBSUFFIX '2009'}
                        {$ELSE}
                          {$IFDEF D2007}
                            {$LIBSUFFIX '2007'}
                          {$ELSE}
                            {$IFDEF D2006}
                              {$LIBSUFFIX '2006'}
                            {$ELSE}
                              {$MESSAGE ERROR 'The Condition Definitions need to be
updated!!!!!'}
                            {$ENDIF}
                          {$ENDIF}
                        {$ENDIF}
                      {$ENDIF}
                    {$ENDIF}
                  {$ENDIF}
                {$ENDIF}
              {$ENDIF}
            {$ENDIF}
          {$ENDIF}
        {$ENDIF}
      {$ENDIF}
    {$ENDIF}
  {$ENDIF}
{$ENDIF}
```

Then in my test plug-in, I added the two `.inc` files as follows:

```
Library SingleDLLProject;

Uses
  SysUtils,
  Classes,
  SingleDLLProject.Wizard in 'Source\SingleDLLProject.Wizard.pas';

{$INCLUDE Source\CompilerDefinitions.inc}
{$INCLUDE LibrarySuffixes.inc}

{$R *.res}

Begin
End.
```

I had to update my Take Command batch file however once that was done the batch file produced DLLs for each version of RAD Studio back to 2009 (it didn't go back further due to a completely different issue with the 2007

installation not wanting to build OTA projects from the command-line).



Name	Size	Modified
..		
DCUs		2018-03-02 00:19
Source		2018-03-02 00:17
Win32		2018-03-01 22:21
__history		2018-03-01 23:39
__recovery		2018-03-02 00:18
LibrarySuffixes.inc	1 806	2018-03-01 22:45
SingleDLLProject.cfg	5 224	2018-03-01 23:25
SingleDLLProject.dpr	874	2018-03-01 23:35
SingleDLLProject.dproj	29 552	2018-03-01 23:43
SingleDLLProject.dproj.local	1 590	2018-03-01 23:43
SingleDLLProject.dsk	16 738	2018-03-02 00:18
SingleDLLProject.eof	108	2018-03-01 22:14
SingleDLLProject.identcache	188	2018-03-02 00:18
SingleDLLProject.ithelper	0	2018-03-01 22:21
SingleDLLProject.res	688	2018-03-01 23:43
SingleDLLProject.stat	167	2018-03-02 00:18
SingleDLLProject.~dsk	16 223	2018-03-01 22:46
SingleDLLProject2006.dll	78 848	2018-03-01 23:31
SingleDLLProject2007.dll	79 360	2018-03-01 23:31
SingleDLLProject2009.dll	12 288	2018-03-02 00:19
SingleDLLProject2010.dll	13 312	2018-03-02 00:19
SingleDLLProjectXE.dll	13 824	2018-03-02 00:19
SingleDLLProjectXE2.dll	23 552	2018-03-02 00:19
SingleDLLProjectXE3.dll	24 064	2018-03-02 00:19
SingleDLLProjectXE4.dll	24 064	2018-03-02 00:19
SingleDLLProjectXE5.dll	24 064	2018-03-02 00:19
SingleDLLProjectXE6.dll	24 576	2018-03-02 00:19
SingleDLLProjectXE7.dll	25 088	2018-03-02 00:19
SingleDLLProjectXE8.dll	25 088	2018-03-02 00:19
SingleDLLProjectXE10.dll	26 624	2018-03-02 00:19
SingleDLLProjectXE101.dll	28 160	2018-03-02 00:19
SingleDLLProjectXE102.dll	27 136	2018-03-02 00:19

Hopefully this provides you with ideas to simplify your builds.

D.

Related posts:

1. Conditional Compilation of Open Tools API experts (9.9)
2. Stupid things not to do when change compilers (7.7)
3. UniSynEdit for RAD Studio 2010 (6)
4. Compiling Component Packages for C++ (5.5)

Category: Delphi Open Tools API RAD Studio