DCU(Delphi Compiled Unit) Format

卓能文(Zhuo Nengwen)

2024-02-14

Contents

1 Types

1.1 Id

Offset	Name	Type	Notes
0	len	u8	Length.
1	name	utf8 chars	Name.

1.2 FileDate

time: hour(5bits) minute(6bits) second(5bits >> 1)

date: year(7bits + 1980) month(4bits) day(5bits)

1.3 Packed Signed Int(PI)

1. LSB: 0: 7 bit signed int
2. LSB: 01: 14 bit signed int
3. LSB: 011: 21 bit signed int
4. LSB: 0111: 28 bit signed int
5. LSB: 101_1111: 32 bit signed int
6. LSB: 1111_1111: 64 bit signed int

1.4 Packed Unsigned Int(PU)

1. LSB: 0: 7 bit unsigned int
2. LSB: 01: 14 bit unsigned int
3. LSB: 011: 21 bit unsigned int
4. LSB: 0111: 28 bit unsigned int
5. LSB: 101_1111: 32 bit unsigned int
6. LSB: 1111_1111: 64 bit unsigned int

2 Header

Offset	Name	Type	Notes
0	?	u8	Unknown.
1	platform	u8	As following.
2	?	u8	Always 0
3	compilerVersion	u8	As following.
4	fileSize	u32	File size, including this header.
8	compiledDate	FileDate	Compiled date time.
c	crc32	u32	Or 0.

2.1 version

CompilerVersion Constant: https://delphi.fandom.com/wiki/CompilerVersion_Constant

Compiler	CompilerVersion	Defined Symbol
Delphi 12.0 Athens	36	VER360
Delphi 11.0 Alexandria	35	VER350
Delphi 10.4 Sydney	34	VER340
Delphi 10.3 Rio	33	VER330
Delphi 10.2 Tokyo	32	VER320
Delphi 10.1 Berlin	31	VER310
Delphi 10 Seattle	30	VER300
Delphi XE8	29	VER290
Delphi XE7	28	VER280
Delphi XE6	27	VER270
AppMethod 1	26.5	VER265

Delphi XE5	26	VER260
Delphi XE4	25	VER250
Delphi XE3	24	VER240
Delphi XE2	23	VER230
Delphi XE	22	VER220
Delphi 2010	21	VER210
Delphi 2009	20	VER200
Delphi 2007 .NET	19	VER190
Delphi 2007	18.5	VER185 (also VER180)
Delphi 2006	18	VER180
Delphi 2005	17	VER170
Delphi 8 .NET	16	VER160
Delphi 7	15	VER150
Delphi 6	14	VER140
Delphi 5	13(*)	VER130
Delphi 4	12(*)	VER120
Delphi 3	10(*)	VER100
Delphi 2	9(*)	VER90
Delphi 1	8(*)	VER80

(*) These versions did not have a CompilerVersion constant, it was introduced with Delphi 6.

More details: Borland Compiler Conditional Defines: https://delphi.fandom.com/wiki/Borland_Compiler_Conditional_Defines

Product Name	Version	Conditional Define	CompilerVersion
Embarcadero RAD Studio 12.0 Athens	29.0	VER360	36
Embarcadero RAD Studio 11.0 Alexandria	28.0	VER350	35
Embarcadero RAD Studio 10.4 Sydney	27.0	VER340	34
Embarcadero RAD Studio 10.3 Rio	26.0	VER330	33
Embarcadero RAD Studio 10.2 Tokyo	25.0	VER320	32
Embarcadero RAD Studio 10.1 Berlin	24.0	VER310	31
Embarcadero RAD Studio 10 Seattle	23.0	VER300	30
Embarcadero RAD Studio XE8	22.0	VER290	29
Embarcadero RAD Studio XE7	21.0	VER280	28
Embarcadero RAD Studio XE6	20.0	VER270	27
Embarcadero RAD Studio XE5	19.0	VER260	26
Embarcadero RAD Studio XE4	18.0	VER250	25
Embarcadero RAD Studio XE3	17.0	VER240	24
Embarcadero RAD Studio XE2	16.0	VER230	23
Embarcadero RAD Studio XE	15.0	VER220	22
Embarcadero RAD Studio 2010	14.0	VER210	21
CodeGear C++ Builder 2009	12.0	VER200	20
CodeGear Delphi 2007 for .NET	11.0	VER190	19
CodeGear Delphi 2007 for Win32	11.0	VER180 and VER185	18.5
Borland Developer Studio 2006	10.0	VER180	18
Borland Delphi 2005	9.0	VER170	17
Borland Delphi 8 for .NET	8.0	VER160 *	16
C++BuilderX	?	?	
Borland C#Builder	1.0	VER160 *	
Borland Delphi 7	7.0	VER150	15
Borland Kylix 3	3.0	VER140 **	
Borland C++Builder 6	?	VER140 **(‼)	
Borland Kylix 2	2.0	VER140 **	
Borland Delphi 6	6.0	VER140 **	14
Borland Kylix	1.0	VER140 **	
Borland C++Builder 5	?	VER130 ***	
Borland Delphi 5	5.0	VER130 ***	
Borland C++Builder 4	?	VER125	

Borland Delphi 4	4.0	VER120
Borland C++Builder 3	?	VER110 ****
Borland Delphi 3	3.0	VER100
Borland C++ 5	?	?
Borland C++Builder 1	?	VER93
Borland Delphi 2	2.0	VER90
Borland C++ 4.5	?	?
Borland Delphi	1.0	VER80
Borland C++ 4	?	?
Borland Pascal 7	7.0	VER70
Borland C++ 3.1	?	?
Turbo Pascal for Windows 1.5	1.5	VER15
Turbo C++ for DOS 3	?	?
Borland C++ 3	?	?
Turbo C++ for Windows 3 (Win16)	?	?
Turbo Pascal for Windows 1.0	1.0	VER10
Borland C++ 2	?	?
Turbo Pascal 6	6.0	VER60
Turbo C++ for DOS	?	?
Turbo C for DOS 2	?	?
Turbo Pascal 5.5	5.5	VER55
Turbo C for DOS 1.5	?	?
Turbo Pascal 5	5.0	VER50
Turbo Pascal 4	4.0	VER40
Turbo C for DOS	?	?
Turbo Pascal 3	3.0	N/A
Turbo Pascal 2	2.0	N/A
Turbo Pascal 1	1.0	N/A

* This conditional define is shared by the Delphi compilers used to build C#Builder 1 and Delphi 8, which do not natively support Delphi for Win32. This define is used in the "IDE Integration Packs" that were released to Borland partners in order to allow IDE plugins to be compiled.

** This conditional define is shared between C++Builder 6, Delphi 6, Kylix 1, 2, and 3 (Checking for the conditional define "LINUX" helps to determine whether the compiler is Kylix or Delphi and "BCB" can be used to determine if C++Builder is being used).

*** This conditional define is shared with C++Builder 5

**** C++Builder 3.0 used VER110 (it had its own version of the Delphi compiler included).

**** CompilerVersion (Delphi 6 or later) can be used with conditional directives like

```
{$IF CompilerVersion >= 20}  {$DEFINE CanUnicode}  {$IFEND}
```

or using code:

```
if System.CompilerVersion >= 22 then  <do something>;
```

From unofficial sources, look at 4th byte of the .dcu

```
0E = Delphi 6
0F = Delphi 7
11 = Delphi 2005
12 = Delphi 2006 or 2007(BDS2006)
14 = Delphi 2009
15 = Delphi 2010
16 = Delphi XE
17 = Delphi XE2
18 = Delphi XE3
```

```
19 = Delphi XE4
1A = Delphi XE5
1B = Delphi XE6
1C = Delphi XE7
1D = Delphi XE8
1E = Delphi 10 Seattle
1F = Delphi 10.1 Berlin
20 = Delphi 10.2
21 = Delphi 10.3
22 = Delphi 10.4
23 = Delphi 11
24 = Delphi 12
```

There was no change in .dcu format going from Delphi 2006 to Delphi 2007. Therefore they use the same.

Edit Jul 2, 2016 Added XE8, 10 and 10.1 to the list.

On request, also the target platform, which is found in the second byte of the .dcu. Values are of course valid only for versions that have these targets.

```
00 = Win32
03 = Win32
23 = Win64
04 = OSX32
14 = iOSSimulator
67 = Android32
76 = iOSDevice32
77 = Android32
87 = Android64
94 = iOSDevice64
```

3 Tags

3.1 00 Start flag

3.2 02 Unit Addtional Info(>= Delphi7)

```
02 06 55 45 6D 70 74 79 | FE | 27 FE EF 03 (Win64)
      UEmpty
02 15 49 53 42 61 73 65 36 34 41 6E 64 45 6E 63 72 79 70 74 69 6F 6E | FE 27 FE E7 03
(Win32)
02 15 49 53 42 61 73 65 36 34 41 6E 64 45 6E 63 72 79 70 74 69 6F 6E | FE 13 FF F7
02 15 49 53 42 61 73 65 36 34 41 6E 64 45 6E 63 72 79 70 74 69 6F 6E | FE 89 3F (>=
Delphi2009)
02 15 49 53 42 61 73 65 36 34 41 6E 64 45 6E 63 72 79 70 74 69 6F 6E (Delphi2006,
Delphi2007)
02 (Delphi7)
```

Offset	Name	Type	Notes
0	id	Id	Unit Name
?	?	PU	
?	?	PU	

3.3 0A | 10 Segment?

3.4 14

3.5 20 Variable Information

- Delph6

```
20 02 2E 31 | 66 0E 00
      .1
```

3.6 25 Const Defination

- Delphi12

```
25 0E 4E 75 6C 6C 4F 62 6A 65 63 74 46 6C 61 67 | 0A 00 00 30 00 00 FB FF 07

25 11 4F 62 6A 52 65 67 4D 65 74 61 46 69 6C 65 4F 62 6A | 0A 00 00 32 00 00 63 5B 01

25 0D 4F 62 6A 52 65 67 49 63 6F 6E 4F 62 6A | 0A 00 00 32 00 00 6B 5B 01
```

3.7 28 Function

- Delph6

```
28 07 41 6E 73 69 43 68 72 | 80 D6 81 0A C4 00 02 04 04 | 21 03 56 61 6C | 16 02 00 |
20 06 52 65 73 75 6C 74 | 16 04 00 | 63

28 17 47 65 74 45 6E 76 69 72 6F 6E 6D 65 6E 74 56 61 72 69 61 62 6C 65 41 | 80 0C 8D
FD 18 00 24 09 40 0A | 21 04 4E 61 6D 65 12 0A 06 | 23 06 52 65 73 75 6C 74 16 0A 08
| 63
```

- Delphi12

```
28 0C 46 69 6E 61 6C 69 7A 61 74 69 6F 6E | 80 00 00 E9 60 70 C5 00 1E 04 21 90 04 00
| 63
```

End with 63

3.7.1 20 Result

- Delph6

```
20 06 52 65 73 75 6C 74 | 16 04 00
```

3.7.2 21 Parameter

- Delph6

```
21 03 56 61 6C | 16 02 00
21 04 4E 61 6D 65 | 12 0A 06
```

3.7.3 23 Result

- Delph6

```
23 06 52 65 73 75 6C 74 | 16 0A 08
```

3.8 35 String Const Defination

- Delphi12

```
35 06 55 45 6D 70 74 79 | 84 00 00 5F B8 8E CF 02 | 63
      UEmpty
35 06 53 79 73 74 65 6D | 00 00 00 04 | 63
      System
35 08 53 79 73 55 74 69 6C 73 | 80 00 00 00 00 00 00 56 | 63
      SysUtils
35 07 43 6C 61 73 73 65 73 | 80 00 00 00 00 00 00 4C | 63 | 63
      Classes
```

Offset	Name	Type	Notes
0	id	Id	Unit name
?	flag	PU	

Optional end with multiple 63 tags.

3.9 37 Variable Information(Same as 20)

- Delphi12

```
37 02 2E 31 | 66 00 00 02 00
      .1
37 02 2E 31 | 46 00 00 2E 00
```

3.10 61 All File End Flag

3.11 63 End of Any

3.12 64 Interface Use Unit | 65 Implementation Use Unit

- Delphi6

```
64 07 53 79 73 49 6E 69 74 | C8 | 43 D2 EF | 63
```

- Delphi12

```
64 07 53 79 73 49 6E 69 74 | 00 00 00 | 63
```

Offset	Name	Type	Notes
0	id	Id	Unit name
?	?	PU	
?	?	PU	
?	?	PU	

An unit have many const, procedures and types, end with 63 tag.

3.12.1 66 (Import Type)

- Delph6

```
66 04 42 79 74 65 | DD DE 52 6C
```

3.12.2 67 (Import Function)

- Delphi6

```
67 0E 40 48 61 6E 64 6C 65 46 69 6E 61 6C 6C 79 | 58 2C 54 64
```

- Delphi12

```
67 17 40 44 65 6C 70 68 69 45 78 63 65 70 74 69 6F 6E 48 61 6E 64 6C 65 72 | C8 7E 90
F4

67 0F 40 44 79 6E 41 72 72 61 79 4C 65 6E 67 74 68 | 98 D6 54 70 9F F0 58 BE C2

67 08 40 4C 53 74 72 4C 65 6E | B1 D9 99 A8 9F 19 5C 34 F0
```

3.13 70 | 76 Source File Name

```
70 0A 55 45 6D 70 74 79 2E 70 61 73 | 35 7F 91 57 | 00

70 3D 2E 2E 5C 2E 2E 5C 44 65 6C 70 68 69 20 33 5F 35 20 53 6F 75 72 63 65 20 43 6F
64 65 5C 4C 69 62 72 61 72 79 56 33 5C 49 53 44 65 6C 70 68 69 32 30 30 39 41 64 6A
75 73 74 2E 70 61 73 | 0C 75 77 3D | 00
```

Offset	Name	Type	Notes
0	id	Id	Source file name.
?	lastModified	TimeStamp	Last modified datetime.
+ 4	order	PU	Included order, count down to 0

3.14 96 Unit Flag

- Delphi6, Delphi7

```
96 00 3C
```

- >= Delphi2006

```
96 00 06 3C (Delphi2006)
96 00 00 3C (Delphi12)
```

Offset	Name	Type	Notes
0	?	PU	
?	?	PU	
?	?	PU	

4 Some useful sites

4.1 Delphi related

1. Internal Data Formats (Delphi): https://docwiki.embarcadero.com/RADStudio/Sydney/en/Internal_Data_Formats_(Delphi)
2. DCU32INT: http://hmelnov.icc.ru/DCU/index.eng.html

   source: https://gitlab.com/dcu32int/DCU32INT

   The utility DCU32INT parses *.dcu file and converts it into a close to Pascal form. See DCU32INT.txt for more details. The unit versions supported are Delphi 2.0-8.0, 2005-2006/Turbo Delphi (.net and WIN32), 2007-2010 (WIN32), XE (WIN32), XE2-XE3 (WIN32,WIN64,OSX32), XE4 (WIN32,WIN64,OSX32,iOS simulator, iOS device (no code)), XE5-XE7/AppMethod (WIN32,WIN64,OSX32,iOS simulator, iOS device (no code), Android (no code)), XE8, 10 Seattle, 10.1 Berlin (WIN32,WIN64,OSX32,iOS simulator, iOS device 32/64 (no code),Android (no code)), 10.2 Tokyo (WIN32,WIN64,OSX32,iOS simulator, iOS device 32/64 (no code),Android (no code),Linux (no code)), 10.3 Rio (WIN32,WIN64,OSX32,iOS simulator, iOS device 32/64 (no code),Android (no code),Linux (may be - not checked,no code)), Kylix 1.0-3.0.
3. Innova Solutions Object Database - Delphi DCUs: https://github.com/rogerinnova/ISObjectDbDCUs

   Delphi DCUs for Adding an Innova Solutions Object Db into your Delphi Application.

   > **Info**
   >
   > I list here as my test suites, because it has full delphi version dcus.

4. IDR (Interactive Delphi Reconstructor): https://github.com/crypto2011/IDR

   A decompiler of executable files (EXE) and dynamic libraries (DLL), written in Delphi and executed in Windows32 environment. Final project goal is development of the program capable to restore the most part of initial Delphi source codes from the compiled file but IDR, as well as others Delphi decompilers, cannot do it yet. Nevertheless, IDR is in a status considerably to

facilitate such process. In comparison with other well known Delphi decompilers the result of IDR analysis has the greatest completeness and reliability.

5. revendepro: http://www.ggoossen.net/revendepro/

   Revendepro finds almost all structures (classes, types, procedures, etc) in the program, and generates the pascal representation, procedures will be written in assembler. Due to some limitation in assembler the generated output can not be recompiled. The source to this decompiler is freely available. Unfortunately this is the only one decompiler I was not able to use - it prompts with an exception when you try to decompile some Delphi executable file.

6. EMS Source Rescuer: https://ems-source-rescuer.apponic.com/

   EMS Source Rescuer is an easy-to-use wizard application which can help you to restore your lost source code. If you lose your Delphi or C++Builder project sources, but have an executable file, then this tool can rescue part of lost sources. Rescuer produces all project forms and data modules with all assigned properties and events. Produced event procedures don't have a body (it is not a decompiler), but have an address of code in executable file. In most cases Rescuer saves 50-90% of your time to project restoration.

7. Dede: http://www.softpedia.com/get/Programming/Debuggers-Decompilers-Dissasemblers/DeDe.shtml

   source: https://github.com/Hanvdm/dedex

   DeDe is a very fast program that can analyze executables compiled with Delphi. After decompilation DeDe gives you the following:

   - All dfm files of the target. You will be able to open and edit them with Delphi.
   - All published methods in well commented ASM code with references to strings, imported function calls, classes methods calls, components in the unit, Try-Except and Try-Finally blocks. By default DeDe retrieves only the published methods sources, but you may also process another procedure in a executable if you know the RVA offset using the Tools| Disassemble Proc menu.
   - A lot of additional information.
   - You can create a Delphi project folder with all dfm, pas, dpr files. Note: pas files contains the mentioned above well commented ASM code. They can not be recompiled!

4.2 others

1. x86 Disassembly/Disassemblers and Decompilers: https://en.wikibooks.org/wiki/X86_Disassembly/Disassemblers_and_Decompilers

2. Software optimization resources: https://www.agner.org/optimize/

3. Okteta: https://apps.kde.org/okteta/

   Okteta is a simple editor for the raw data of files.

   Features:

   - Values and characters shown either in two columns (the traditional display in hex editors) or in rows with the value on top of the character
   - Editing and navigating similar to a text editor
   - Customizable data views, with loadable and storable profiles
   - Tools dockable on all sides or floating
   - Numerical encodings: Hexadecimal, Decimal, Octal, Binary

- Character encodings: All 8-bit encodings as supplied by Qt, EBCDIC
- Fast data rendering on screen
- Multiple open files
- Undo/redo support
- Structures tool for analyzing and editing based on user-creatable - - structure definitions
- And more…