

Migracija Delphi aplikacij iz Win32 v Win64



Primož Gabrijelčič

<http://primoz.gabrijelcic.org>

<http://www.thedelphigeek.com>

random ramblings on Delphi, programming, Delphi programming, and all the rest

Tuesday, April 21, 2026

OmniThreadLibrary-NG teaser

Normal suite (default — Stress excluded)

Platform	Found	Ignored	Passed	Failed	Time
Win32	322	0	322	0	40 s
Win64	322	0	322	0	42 s
Linux64	325	3	322	0	32 s
Android	325	3	322	0	38 s
Total			1288	0	~2 min 32 s

Stress only (--include:Stress)

Platform	Found	Ignored	Passed	Failed	Time
Win32	7	0	7	0	293 s
Win64	7	0	7	0	293 s
Linux64	7	0	7	0	275 s
Android	7	0	7	0	284 s
Total			28	0	~19 min 5 s



embarcadero
MVE

Pages

[Presentations](#)



Hands-On
Design Patterns
with Delphi

Od kod prihajamo in kam gremo

Platforme včasih in danes

- Windows 16-bit [Delphi 1, 1995]
- Windows 32-bit [Delphi 2, 1996]
- Windows 64-bit [Delphi XE2, 2011]
- Delphi 13, 2025: ~~Android32~~, Android64, iOS64, Linux64, Mac64, MacARM64
- Delphi 13.1: WindowsARM64EC
 - <https://www.embarcadero.com/products/rad-studio/whats-new-in-13-florence>

Danes: Migracija Delphi aplikacij iz Win32 v Win64

- Razlogi za migracijo
- Težave in rešitve
- Testiranje
- Uporaba AI pri migraciji

Razlogi za migracijo

Zakaj sploh migrirati iz Win32 v Win64?

- Omejitev 32-bitnega naslovnega prostora ($\approx 2-3$ GB)
 - `{ $\$$ SetPEFlags IMAGE_FILE_LARGE_ADDRESS_AWARE}`
- Boljša izraba sodobnih CPU-jev
- Prihodnost Windows je 64-bitna
 - Ne pričakujte 32-bitne podpore na novih arhitekturah
 - <https://docwiki.embarcadero.com/RADStudio/Florence/en/ARM64EC>
- (Zunanje knjižnice, ki niso več podprte v 32 bitih)

“Zamenjaj platformo in prevedi“?

- Koda se pogosto prevede brez napak
- Napake so skrite in odvisne od podatkov
- Težave se pogosto pokažejo šele v produkciji

Razlike med Win32 in Win64

Kaj se spremeni v Win64

- Kazalci: 32 → 64 bit
- Poravnava zapisov (record)
- Velikost nekaterih podatkovnih tipov (NativeInt, LongInt ...)
 - `cardinalVar := cardinal(pointerVar)`
- Zunanji DLL morajo biti 64-bitni
 - Težava, če imate samo 32-bitne verzije
- Zbirnik (assembler), prenos parametrov po skladu, obravnavanje prekinitev (exceptions)

Kazalci

- `cardinalVar := cardinal(pointerVar);`
 - Včasih deluje, včasih ne, odvisno od vrednosti v *pointerVar*
- `PostMessage(Handle, 123, 0, cardinal(pointer));`
 - => `NativeUInt(pointer)`
 - => `LPARAM(pointer)`

Zapisi in poravnava (record)

- SizeOf(record) se lahko spremeni
 - Zaradi spremenje velikosti osnovnih tipov
 - Zaradi spremenjenega 'Record field alignment'
 - V Windows32 je 'Double word' ali 'Quad word', odvisno od verzije
- Posebej nevarno pri file/network I/O
- Previdno, če uporabljate zunanje knjižnice (C, C++)
- Previdno pri klicih Windows API, če imate lastne deklaracije

Assembler in Win64

- Druga imena registrov
- Dodatni/spremenjeni ukazi
- Inline assembler v Win64 ne obstaja
 - Še vedno imamo ,pure asm' funkcije
 - Samo na Intel platformah
- Potrebna je predelava

Assembler: Strategije za migracijo

- Popoln prepis v Pascal (priporočeno)
 - Če hitrost ni problem
- Dvojna implementacija
 - V .pas kodi
 - Zunanja koda – objektna datoteka
 - {\$L 'helper.obj'}
 - `function zlibVersion: MarshaledAString; cdecl; external object 'zutil.o';`
 - Zunanja koda – knjižnica
 - WinARM64EC lahko naloži Intel DLL
 - `function zlibVersion: MarshaledAString; cdecl; external 'libzlib.dll';`
 - `function zlibVersion: MarshaledAString; cdecl; external 'libzlib.a';`

Komponente in zunanje knjižnice

- Nadgradnja na novejšo različico
- „Naredi sam“ – če imate izvorno kodo
- Menjava komponent/knjižnic
- Delna nadgradnja na 64-bitno kodo

Delna nadgradnja

- 32-bitni program, ki prenese del funkcionalnosti v 64-bitno kodo
- 64-bitni program, ki prenese del funkcionalnosti v 32-bitno kodo
- 32-bitni program ne more naložiti 64-bitnega DLL
 - In obratno
- 32-bitni program lahko požene 64-bitni program
 - In obratno

Komunikacijski kanali

- IPC (anonymous pipes, named pipes, memory-mapped files, mailslots, WM_COPYDATA)
- COM (out-of-process, surrogate / dllhost.exe)
- Rundll32 (odsvetovano)
- Sockets (UDP, TCP)
- Files

Primeri iz prakse

cardinal(pointer)

```
var headerEnd: cardinal;  
methodInfoHeader := ObjAuto.GetMethodInfo(  
    WorkerIntf.Implementor, methodName);  
headerEnd := cardinal(methodInfoHeader) + methodInfoHeader^.Len;
```

- Delovalo do Delphi 13.1!

```
if PInteger(@opsStage)^ <> NativeInt(nil) then begin  
    Assert(PInteger(@opsStageEx)^ = NativeInt(nil));
```

Koda iz zunanje knjižnice

```
function CompatibleTimeSetEvent(Delay, Resolution: UINT;  
    TimeProc: TFNTimeCallback; User: DWORD; Event: UINT):  
MMResult;
```

```
FEndOfStreamTimer := CompatibleTimeSetEvent(Delay,  
TIMEOUT_RESOLUTION, @EndOfStreamTimer, Cardinal(Self),  
TIME_ONESHOT);
```

Nizkonivojski heki

```
if ImageNTHaders.FileHeader.Machine  
    <> {$IFDEF CPUX64} $8664 {$ELSE} $14C {$ENDIF}  
then
```

- CPUX64 ali CPU64BITS
- [https://docwiki.embarcadero.com/RADStudio/en/Conditional_compilation_\(Delphi\)](https://docwiki.embarcadero.com/RADStudio/en/Conditional_compilation_(Delphi))

Windows API

```
SP_DEVICE_INTERFACE_DETAIL_DATA_A = packed record
    cbSize: DWORD;
    DevicePath: array [0..ANYSIZE_ARRAY - 1] of AnsiChar;
{$IFDEF WIN64}
    Padding: array [0..2] of AnsiChar;
{$ENDIF WIN64}
end;
```

Testiranje

Kazalci

- FastMM4
 - AlwaysAllocateTopDown
 - FastMM4Options.inc
 - Ni potrebno v Delphi 13
 - FullDebugMode
 - CheckHeapForCorruption (slow!)

Statična analiza kode - ročno

- cardinal(), integer(), longint(), longword(), pointer
- SetWindowLong, GetWindowLong, SendMessage, PostMessage
 - SetWindowLongPtr
- GWL_USERDATA, GWL_WNDPROC

- Case label outside of range of case expression **True**
- Combining signed and unsigned 64-bit types - treated as an unsigned type **True**
- Implicit cast of one integer type to another may result in data loss **True**
- Implicit conversion from wide integral type to narrow type, may result in data loss **True**
- Unsafe typecast **True**

Statična analiza kode - programsko

- FixInsight
- Peganza
- AI

Testiranje - ročno

- Testirajte z velikimi podatki
 - Velikost enega vira (datoteke) > 2GB
 - Število podatkov (zapisov) > MaxInt

Testiranje - avtomatsko

- Application Verifier
 - <https://learn.microsoft.com/en-us/windows-hardware/drivers/devtest/application-verifier>

AI

LLM AI kot pomočnik

- Iskanje problematičnih mest
- Razlaga zakaj je koda nevarna
- Postopna migracija enot

Primer AI prompta

- “Check unit xxx for 64-bit migration problems”
- Bolje: “What would be a good query to give Claude to check existing codebase for 64-bit migration problems?”

Pretvorba zbirnika

- „Can you convert this function to Win64 and also write a pure pascal version?“
- Obvezno: „Please verify that all three versions are functionally identical.“
- Obvezno²: Testiranje!

Zaključne misli

Zaključek

- Win64 migracija ni trivialen problem
- Problemi se lahko skrivajo dolgo časa
- AI je odličen pomočnik, ne nadomestilo