Writing FAST Delphi programs

EKON 24

Primož Gabrijelčič

- programmer, MVP, writer, blogger, consultant, speaker
- Blog *http://thedelphigeek.com*
- Twitter *@thedelphigeek*
- Skype *gabr42*
- LinkedIn *gabr42*
- GitHub *gabr42*
- SO *gabr*
- http://primoz.gabrijelcic.org

EKON 24

The Delphi Geek

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

Saturday, May 30, 2020

OmniThreadLibrary 3.07.8

New OmniThreadLibrary is out! Get it while it's hot!

Version 3.07.8 is mostly a bugfix release. It fixes few small bugs and enables support for Delphi 10.4.

You can get it now on git, download the ZIP archive, install it with Delphinus or with GetIt (in few days).

For more information, visit OmniThreadLibrary home page or write your question on the forum.
Read more »

Posted by gabr42 at 19:45 2 comments:
Labels: Delphi, multithreading, OmniThreadLibrary, open source

Wednesday, May 27, 2020

Top three Delphi 10.4 features

Delphi 10.4 has just been released (turn here for a great overview) and has some nice enhancements even for us, die-hard Windows developers. It is too early to give any deep analysis as I have just installed it and did not do any thorough testing, but I can already pick my top three new features. In no particular order, here they are:
Read more »

Posted by gabr42 at 09:24 3 comments:

Friday, February 14, 2020

Long live Delphi!

Something great has happened on this day, 25 years ago.

It was the sign of Aquarius. People were listening to Creep. And the AppBuilder was released.

You don't know AppBuilder? Sure you do! It was developed and — thanks to Novell — released under codename *Delphi*.

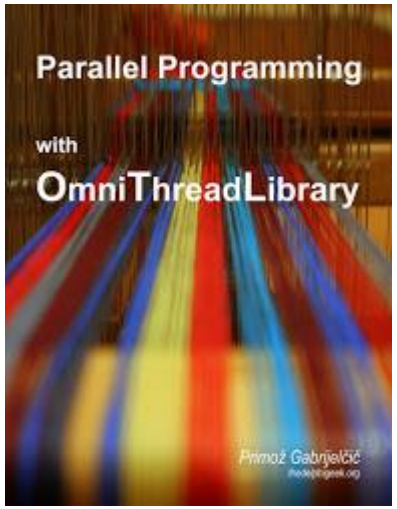
Indeed, our beloved Delphi is 25 years old today! A quarter of a century!

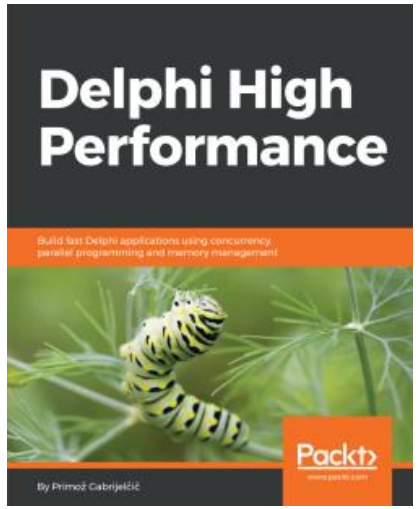
Pages

Presentations

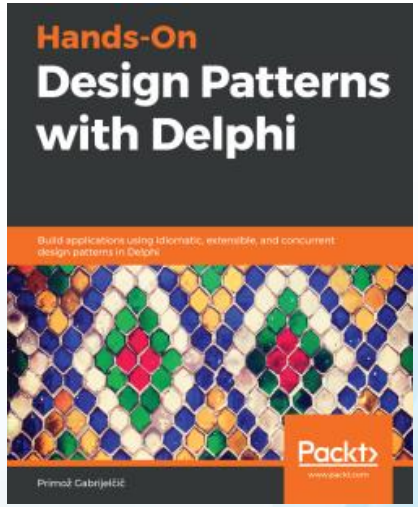
EKON 24

Books





- https://leanpub.com/omnithreadlibrary
- https://www.packtpub.com/product/delphi-high-performance/9781788625456
- https://www.packtpub.com/product/hands-on-design-patterns-with-delphi/9781789343243



EKON 24

Performance

EKON 24

Performance

- What is *performance*?
- How do we “add it to the program”?
  - There is no silver bullet!

What is performance?

- Running “fast enough”

- Raw speed
- Responsiveness
  - Non-blocking

Improving performance

EKON 24

Top-down approach

1. Find the problem
2. Fix the algorithm
3. Fine tune the code

Don’t skip steps!

9 steps to a faster program

- Find the problem
  1. (Educated) guess
  2. **Measure!**
- Fix the algorithm
  3. Run less code
  4. Find a better algorithm
- Fine tune the code
  5. Compiler settings
  6. Code tuning
  7. Memory manager
  8. Parallel code
  9. External libraries

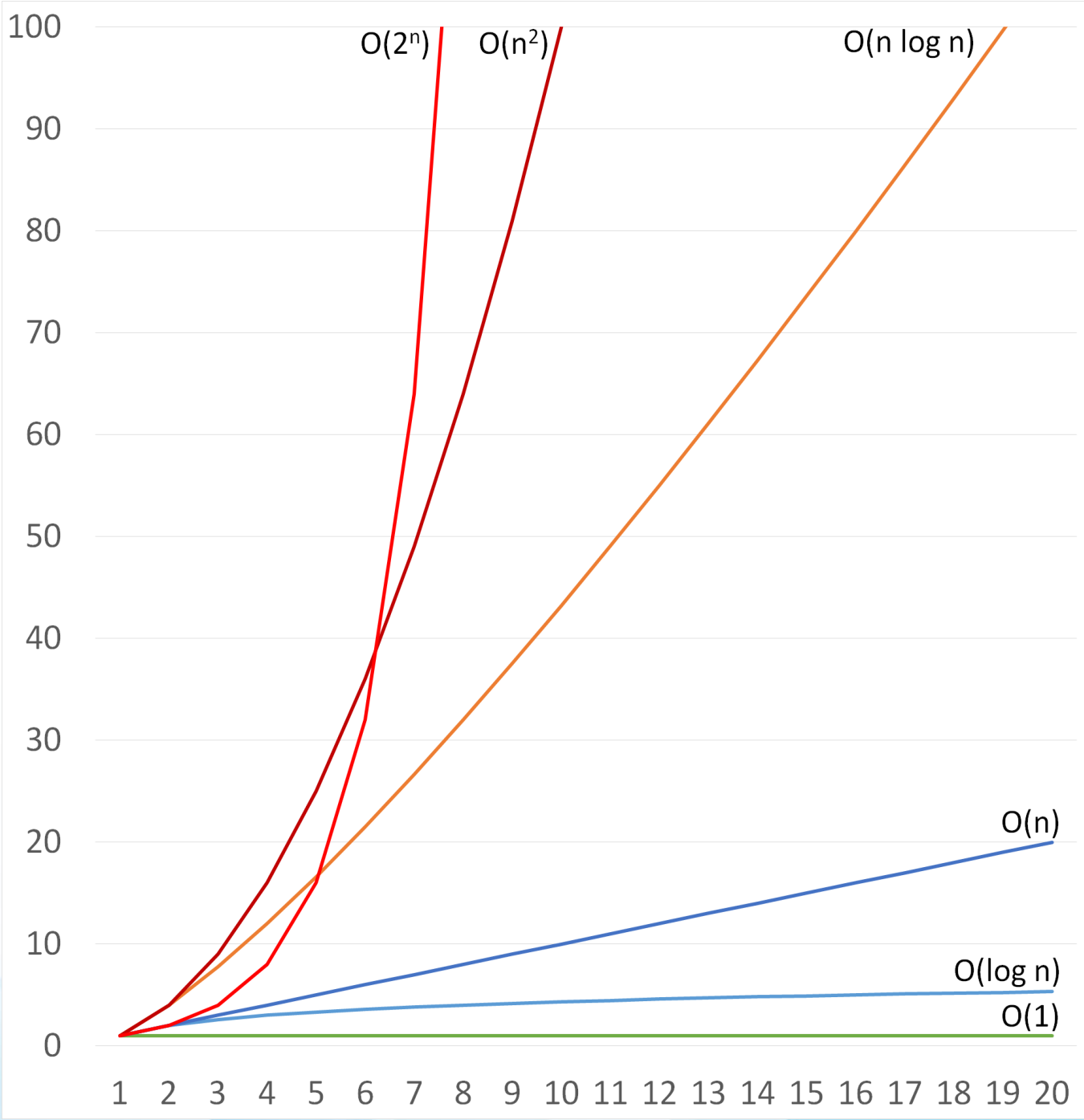
1. Performance estimation

EKON 24

Algorithm complexity

- Tells us how algorithm slows down if data size is increased by a factor of *n*
- O()
  - O(n), O(n^2), O(n log n) …
- Time **and** space complexity
- Constant factors are ignored …
  - … but can be important

- $O(1)$ accessing array elements
- $O(\log n)$ searching in ordered list
- $O(n)$ linear search
- $O(n \log n)$ quick sort (average)
- $O(n^2)$ quick sort (worst), naive sort (bubblesort, insertion, selection)
- $O(c^n)$ recursive Fibonacci, travelling salesman



Comparing complexities

Data size	O(1)	O(log n)	O(n)	O(n log n)	$O(n^2)$	$O(c^n)$
1	1	1	1	1	1	1
10	1	4	10	43	100	512
100	1	8	100	764	10.000	10^{29}
300	1	9	300	2.769	90.000	10^{90}

Delphi RTL

http://bigocheatsheet.com/

- Lists
  - access O(1)
  - Search O(n) / O(n log n)
  - Sort O(n log n)
- Dictionary
  - * (incl. search) O(1)
  - Search by value O(n)
- Trees
  - * O(log n)
  - Spring4D

2. Measuring performance

EKON 24

Measuring

- Manual
  - GetTickCount
  - QueryPerformanceCounter
  - Tstopwatch
  - GpTiming

Measuring

- Automated – Profilers
  - Sampling
  - Instrumenting
    - Binary
    - Source

Free profilers

- AsmProfiler
  - André Mussche
  - Instrumenting and sampling
  - 32-bit
  - https://github.com/andremussche/asmprofiler
- Sampling Profiler
  - Eric Grange
  - Sampling
  - 32-bit (64-bit in test)
  - https://www.delphitools.info/samplingprofiler/
  - https://www.delphitools.info/2020/09/23/samplingprofiler-64-test-version/

Commercial profilers

- AQTime
  - SmartBear
  - Instrumenting and sampling
  - 32- and 64-bit
  - https://smartbear.com/product/aqtime-pro/
- Nexus Quality Suite
  - NexusQA
  - Instrumenting
  - 32- and 64-bit
  - https://www.nexusdb.com
- ProDelphi
  - Helmuth J.H. Adolph
  - Instrumenting (source)
  - 32- and 64-bit
  - http://www.prodelphi.de/

3. Execute less code

EKON 24

Two simple ways to speed up a program

- If a part of program is slow, don’t execute it **so much**
- If a part of program is slow, don’t execute it **at all**

“Don’t execute it so much”

- Don’t update UI thousands of times per second
- Call BeginUpdate/EndUpdate
- Don’t send around millions of messages per second

“Don’t execute it at all”

- UI virtualization
  - Virtual listbox
  - Virtual TreeView
- Memoization
  - Caching
  - Dynamic programming
  - TGpCache<K,V>
    - O(1) all operations
    - GpLists.pas, https://github.com/gabr42/GpDelphiUnits/

4. Find a better algorithm

EKON 24

Find a better algorithm

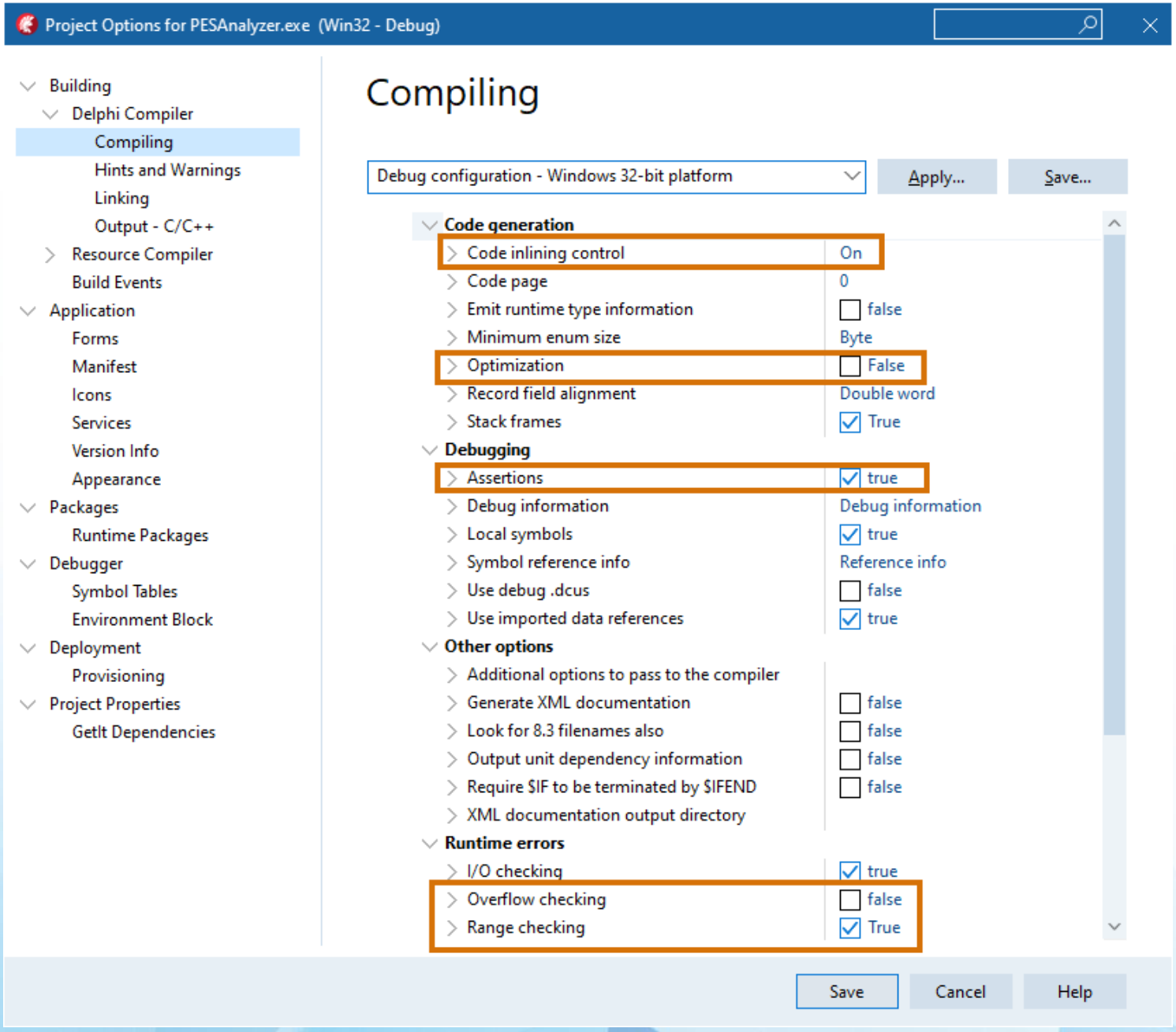
- Books
  - Donald E. Knuth, **The Art of Computer Programming**
    Niklaus Wirth, **Algorithms + Data Structures = Programs**
    Julian Bucknall, **Algorithms and Data Structures**
    Thomas H. Cormen, **Introduction to Algorithms**
    Robert Sedgewick & Kevin Wayne, **Algorithms**
- Web
  - Algorithms, 4th Edition, https://algs4.cs.princeton.edu/home/
  - Udacity, edX, Udemy, Coursera …
  - https://www.geeksforgeeks.org/data-structures/
- Forums
  - Delphi-PRAXIS, www.delphipraxis.net, en.delphipraxis.net

Find a better algorithm

- Books
  - Donald E. Knuth, **The Art of Computer Programming**
    Niklaus Wirth, **Algorithms + Data Structures = Programs**
    Julian Bucknall, **Algorithms and Data Structures**
    Thomas H. Cormen, **Introduction to Algorithms**
    Robert Sedgewick & Kevin Wayne, **Algorithms**
- Web
  - Algorithms, 4th Edition, https://algs4.cs.princeton.edu/home/
  - Udacity, edX, Udemy, Coursera …
  - www.geeksforgeeks.org/fundamentals-of-algorithms/
    www.geeksforgeeks.org/data-structures/
- Forums
  - Delphi-PRAXIS, www.delphipraxis.net, en.delphipraxis.net

5. Compiler settings

EKON 24


Project Options for PESAnalyzer.exe (Win32 - Debug)
Building
Delphi Compiler
Compiling
Hints and Warnings
Linking
Output - C/C++
Resource Compiler
Build Events
Application
Forms
Manifest
Icons
Services
Version Info
Appearance
Packages
Runtime Packages
Debugger
Symbol Tables
Environment Block
Deployment
Provisioning
Project Properties
GetIt Dependencies
Compiling
Debug configuration - Windows 32-bit platform
Apply...
Save...
Code generation
Code inlining control
On
Code page
0
Emit runtime type information
false
Minimum enum size
Byte
Optimization
False
Record field alignment
Double word
Stack frames
True
Debugging
Assertions
true
Debug information
Debug information
Local symbols
true
Symbol reference info
Reference info
Use debug .dcus
false
Use imported data references
true
Other options
Additional options to pass to the compiler
Generate XML documentation
false
Look for 8.3 filenames also
false
Output unit dependency information
false
Require $IF to be terminated by $IFEND
false
XML documentation output directory
Runtime errors
I/O checking
true
Overflow checking
false
Range checking
True
Save
Cancel
Help

EKON 24

6. Code tuning

EKON 24

Behind the scenes

- Strings
  - Reference counted, Copy on write
- Arrays
  - Static
  - Dynamic
    - Reference counted, Cloned
- Records
  - Initialized if managed
- Classes
  - Reference counted on ARC
- Interfaces
  - Reference counted

Calling methods

- Parameter passing
  - Dynamic arrays are strange
- Inlining
  - Single pass compiler!

7. Memory manager

EKON 24

“Do it less often”

- Less memory operations → Faster code
- Pre-allocate, over-allocate

Optimizing parallel (de)allocations

- FastMM4 from GitHub
  - Pierre le Riche
  - https://github.com/pleriche/FastMM4
- DEFINE LogLockContention
- DEFINE UseReleaseStack

- FastMM5
  - https://github.com/pleriche/FastMM5
  - Dual license!

8. Parallel code

EKON 24

When to parallelize?

- **When other means are exhausted**

- Pushing long operations into background
  - “Unblock” the user interface
- Supporting multiple clients in parallel
- Speeding up the algorithm
  - Hard!

Common problems

- Accessing UI from a background thread
- Reading/writing shared data

Synchronization

- Critical section
- Monitor
- Readers-writers / MREW / SWMR
  - TMREWSync / TMultiReadExclusiveWriteSynchronizer
    - Terribly slow
  - TLightweightMREW
    - Windows / Posix wrapper
    - Delphi 10.4.1

Common problems

- Accessing UI from a background thread
- Reading/writing shared data
  - Synchronization
    - Slowdown
    - Deadlocks

Interlocked operations

- “Microlocking”
- Faster
- Limited use

Common problems

- Accessing UI from a background thread
- Reading/writing shared data
  - Synchronization
    - Slowdown
    - Deadlocks
  - Interlocked
    - Complicated
    - Easy to break the code

Communication

- Windows messages
- TThread.Queue
  - TThread.Synchronize too, but …
- polling

Common problems

- Accessing UI from a background thread
- Reading/writing shared data
  - Synchronization
    - Slowdown
    - Deadlocks
  - Interlocked
    - Complicated
    - Easy to break the code
  - Communication
    - Requires algorithm redesign

9. External libraries

EKON 24

External libraries

- Find better implementation
- **Find / create linking unit!**
- Be prepared to find a different implementation
  - Libraries may become unsupported at any time

Q & A

EKON 24