




Embarcadero Conference



Modernizando: Unicode, 64bit e VCL Styles

[Kelter Merlotti]

[Kelver Merlotti]

- Embarcadero MVP
- Coordenador Editorial da Active Delphi
- Consultor e Instrutor independente
- Especialista em Mobile e Multicamadas
- Intitulado “Detentor de todas as respostas” #DeusTaVendo

@ kmerlotti@gmail.com

@kmerlotti

kmerlotti

Embarcadero conference



Unicode

[Unicode: muito prazer!]

- Unicode é um **padrão** que permite aos computadores representar e manipular, de forma consistente, texto de qualquer sistema de escrita existente.
 - Fonte: ~~wikipédia~~ The Unicode Standard. Version 5.0 5 ed. ISBN 0321480910
- Traduzindo: suportar mais caracteres!

[Unicode: e o Delphi com isso?]

- Desde o Delphi 2009, 100% Unicode
- Benefício ou Malefício? Nenhum... Fodefício!
- Veio pra resolver problemas que antes não tínhamos
- Mal necessário 🙌

[Unicode: o que mudou?]

- **String** agora é **UnicodeString**, não mais **AnsiString**
- **Char** agora é **WideChar** (2 bytes) - UTF-16
- **PChar** é **PWideChar**
- **AnsiString** representa o “antigo” tipo **String**

[Unicode: o que NÃO mudou?]

- AnsiString
- WideString
- AnsiChar, PAnsiChar
- ShortString continua contendo elementos AnsiChar
- Conversões implícitas

[Unicode: o que NÃO mudou?] (continuação)

- Concatenação de strings
- Funções para manipulação de strings, como:
 - Lenght;
 - Copy;
 - Pos; etc.
- Comparação de strings, como:
 - <string> <op. de comparação> <string>;
 - CompareStr();
 - CompareText(); etc.
- FillChar ()
- Windows API

[Unicode: com o que me preocupar?]

- Assumir que o `Sizeof(Char)` é 1.
- Assumir que o tamanho de uma string em bytes é igual ao número de caracteres na string.
- Manipular diretamente String ou PChars.
- **Gravar/ler Strings de/para arquivos.**
- Dica: fique atento aos Warnings!

[Unicode: onde consigo mais informações?]

- [Embarcadero Migration and Upgrade Center](#)
 - <https://www.embarcadero.com/rad-in-action/migration-upgrade-center>
- Artigos do Nick Hodges: **Delphi in a Unicode World**
 - Parte I - <http://edn.embarcadero.com/article/38437>
 - Parte II - <http://edn.embarcadero.com/article/38498>
 - Parte III - <http://edn.embarcadero.com/article/38693>



64bit

[64bit: pra quê?]

- Em suma: para lidar com volumes maiores de dados!
- Você sabia que em 2009, a Microsoft anunciou que a partir do Windows Server 2008 R2, seu SO só rodaria em processadores 64bit?
- Você sabia que com 32bit está limitado a 4GB de RAM?
 - $2^{32} = 4.294.967.296$ bytes
 - $2^{64} = 18.446.744.073.709.551.616$ bytes
 - Já passou por algum “out of memory” ? =)

[64bit: o que NÃO mudou?]

- Integer, Logint, Cardinal (continuam 32bit)
- Int64 e UInt64 (continuam 64bit)
- UnicodeString, AnsiString, WideString
- Exceptions
- Runtime Library (RTL)
- SysUtils, Classes, etc...

[64bit: o que mudou?]

- NativeInt, NativeUInt
- Pointer
- Arrays dinâmicos
- Floating point – Double
- Propriedade Tag passou a ser NativeInt

[64bit: com o que me preocupar?]

- `SizeOf(Pointer) <> SizeOf(Integer)`
- Todo o código envolvido no processo deve ser 64bit
 - DLLs **precisam** ser 64bit tbm
- Único *calling convention*
 - **register**, **pascal**, **cdecl** e **stdcall** são ignoradas
 - **safecall** ainda é “*especial*”

[64bit: onde consigo mais informações?]






- Embarcadero Migration and Upgrade Center
 - <https://www.embarcadero.com/rad-in-action/migration-upgrade-center>
- 64-bit Windows Development with Delphi
 - <https://www.embarcadero.com/products/delphi/64-bit>



VCL Styles

[VCL Styles: muito prazer!]

- Solução nativa (desde o XE2) para dar um “upgrade” no visual da sua aplicação
- Arquivos *.vsf em [Documentos Públicos\Embarcadero\Studio\18.0\Styles]

 AquaGraphite.vsf	28/08/2011 18:55	Arquivo VSF	83 KB
 AquaLightSlate.vsf	28/08/2011 18:55	Arquivo VSF	79 KB
 EmeraldLightSlate.vsf	28/08/2011 18:55	Arquivo VSF	79 KB
 GoldenGraphite.vsf	28/08/2011 18:55	Arquivo VSF	83 KB
 RubyGraphite.vsf	28/08/2011 18:55	Arquivo VSF	83 KB

- Registrados em *Designtime* ou *Runtime*
- Via código, manipuláveis através da classe TStyleManager
- Personalizáveis

[VCL Styles: demo!]

[VCL Styles: onde consigo mais informações?]

- [Working with VCL Styles](#)

- http://docwiki.embarcadero.com/RADStudio/Berlin/en/Working_with_VCL_Styles

- [Post do Marco Cantu: “VCL in XE7: Styles”](#)

- <http://blog.marcocantu.com/blog/2014-september-vcl-xe7-styles.html>

@ kmerlotti@gmail.com

@kmerlotti

f kmerlotti



[OBRIGADO]

Embarcadero conference





Embarcadero Conference