embarcadero®

Um evento com propósito

Embarcadero Conference 2020 Online

Integração do **Delphi** com aplicações externas via **Open Tools API**

André Luis Celestino

embarcadero®

Quem sou eu?

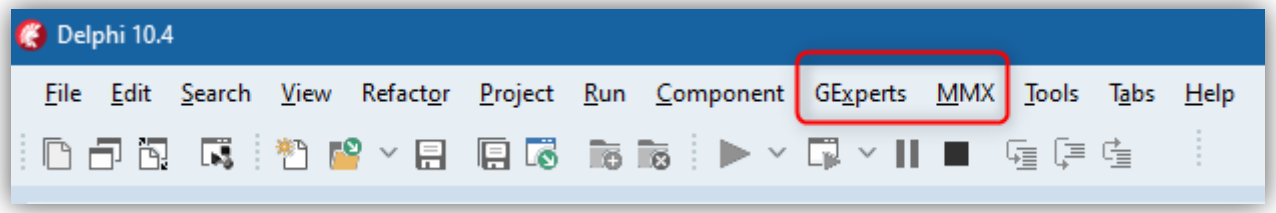
André Luis Celestino

- 9 anos de experiência em programação
- Desenvolvedor @ DB1 Group
- Embarcadero MVP desde 2017
- *Scrum Fundamentals Certified*
- *Scrum Foundation Professional Certificate*
- *SAFe Practitioner*
- *Certified Delphi Developer*
- Autor do blog www.andrecelestino.com


“Open Tools API”
Que isso?
embarcadero

Você já usou?

- CnPack
- GExperts
- MMX
- DDevExtensions
- IDE Fix Pack



Open Tools API (OTA, IOTA, OTAPI)

Trata-se de um conjunto de Interfaces que permitem o acesso à funcionalidades da IDE.

Por exemplo...

- Menu principal
- Editor de Códigos
- *Object Inspector*
- *Project Manager*
- *Toolbars*

Então você pode...

- Criar *wizards* (*add-ons*, *plugins*)
- Adicionar atalhos para aplicações externas
- Automatizar ações recorrentes

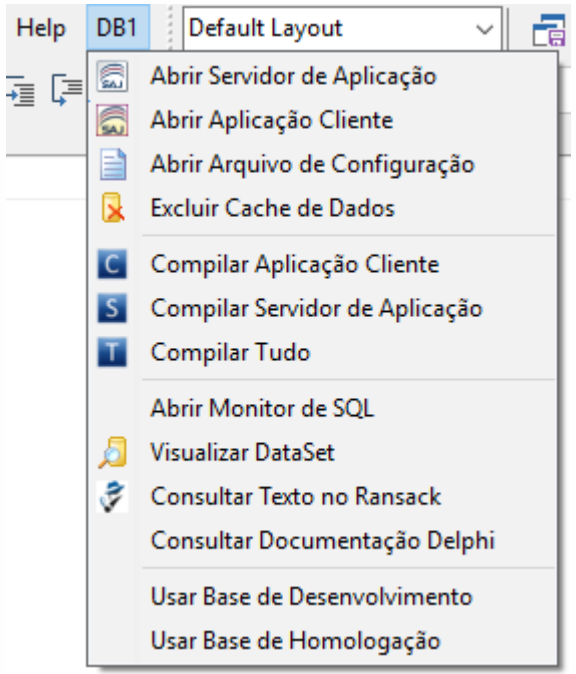
Introdução à algumas Interfaces

- *IOTAWizard* – desenvolvimento de plugins
- *IOTAMenuWizard* – criação de novos menus
- *IOTAEditor* – editor de códigos
- *IOTAFormEditor* – editor de formulários
- *IOTAEnviromentOptions* – opções da IDE
- *IOTAProjectGroup* – grupos de projetos
- *IOTAProject* – projeto ativo
- *IOTABreakpoint* – ações com o *breakpoints*
- *IOTAkeyboardBinding* – atalhos para funcionalidades
- *IOTAGetItServices* – ações com o GetIt Manager
- * *IOTAIDEThemingServices* – temas da IDE
- * *IOTAAsyncCodeInsightManager* – ações com o Code Insight

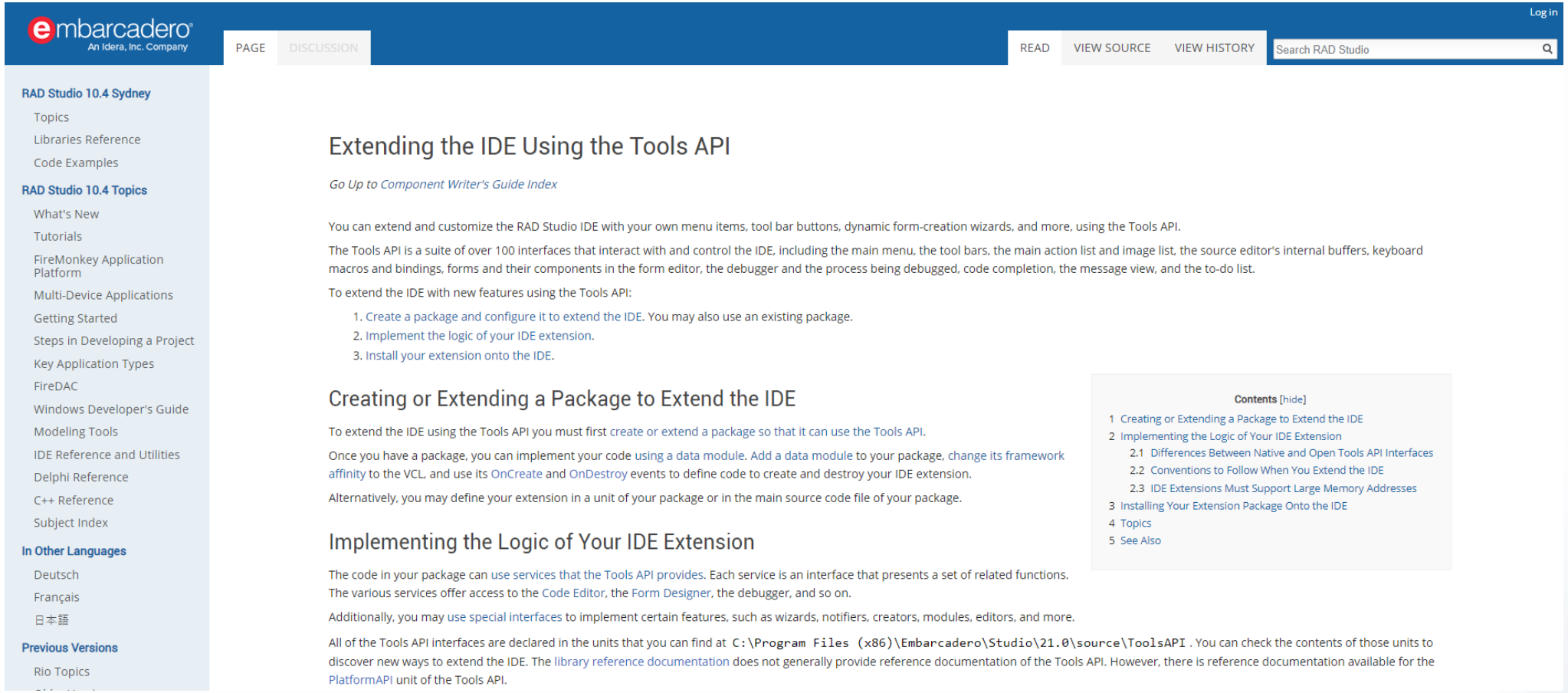
ToolsAPI.pas

Show me the code!

Plugin da DB1



Documentação



embarcadero
An Idera, Inc. Company

PAGE DISCUSSION READ VIEW SOURCE VIEW HISTORY Search RAD Studio Log in

RAD Studio 10.4 Sydney
- Topics
- Libraries Reference
- Code Examples

RAD Studio 10.4 Topics
- What's New
- Tutorials
- FireMonkey Application Platform
- Multi-Device Applications
- Getting Started
- Steps in Developing a Project
- Key Application Types
- FireDAC
- Windows Developer's Guide
- Modeling Tools
- IDE Reference and Utilities
- Delphi Reference
- C++ Reference
- Subject Index

In Other Languages
- Deutsch
- Français
- 日本語

Previous Versions
- Rio Topics

Extending the IDE Using the Tools API

Go Up to Component Writer's Guide Index

You can extend and customize the RAD Studio IDE with your own menu items, tool bar buttons, dynamic form-creation wizards, and more, using the Tools API.

The Tools API is a suite of over 100 interfaces that interact with and control the IDE, including the main menu, the tool bars, the main action list and image list, the source editor's internal buffers, keyboard macros and bindings, forms and their components in the form editor, the debugger and the process being debugged, code completion, the message view, and the to-do list.

To extend the IDE with new features using the Tools API:

1. Create a package and configure it to extend the IDE. You may also use an existing package.
2. Implement the logic of your IDE extension.
3. Install your extension onto the IDE.

Contents [hide]
1 Creating or Extending a Package to Extend the IDE
2 Implementing the Logic of Your IDE Extension
2.1 Differences Between Native and Open Tools API Interfaces
2.2 Conventions to Follow When You Extend the IDE
2.3 IDE Extensions Must Support Large Memory Addresses
3 Installing Your Extension Package Onto the IDE
4 Topics
5 See Also

Creating or Extending a Package to Extend the IDE

To extend the IDE using the Tools API you must first create or extend a package so that it can use the Tools API.

Once you have a package, you can implement your code using a data module. Add a data module to your package, change its framework affinity to the VCL, and use its OnCreate and OnDestroy events to define code to create and destroy your IDE extension.

Alternatively, you may define your extension in a unit of your package or in the main source code file of your package.

Implementing the Logic of Your IDE Extension

The code in your package can use services that the Tools API provides. Each service is an interface that presents a set of related functions. The various services offer access to the Code Editor, the Form Designer, the debugger, and so on.

Additionally, you may use special interfaces to implement certain features, such as wizards, notifiers, creators, modules, editors, and more.

All of the Tools API interfaces are declared in the units that you can find at `C:\Program Files (x86)\Embarcadero\Studio\21.0\source\ToolsAPI` . You can check the contents of those units to discover new ways to extend the IDE. The library reference documentation does not generally provide reference documentation of the Tools API. However, there is reference documentation available for the PlatformAPI unit of the Tools API.

andredelphi@gmail.com

www.andrecelestino.com

https://github.com/AndreLuisCelestino