



What's New in <RAD Studio 12.2>

Sponsored by



Embarcadero is part of **Idera, Inc.**

IDE & Tooling division:

- Embarcadero: Delphi, C++Builder, RAD Studio, InterBase
- UltraEdit: UltraEdit, UltraCompare, UltraFTP
- Whole Tomato: Visual Assist



whole tomato
Visual Assist

Introducing RAD Studio 12.2

- **NEW FEATURES**

- Modernized C++ Compiler and Toolchains
- Delphi 64-bit compilers and tools
- Smart CodeInsight
- Focus Mode and Scrollbar Annotations
- WebStencils

- **QUALITY**

- Improved Mobile Development
- Visual Assist for C++ Builder Enhancements
- Quality, Quality, Quality





New Features in 12.2

For C++Builder and Delphi



RAD




What's New in <RAD Studio 12.2>

bcc64x: the Modernized C++ Compiler

- *It's done!* 

And there are three exciting things...

- #1 Packages
- #2 CMake & ninja
- #3 Parallel building... and *speeeeeeed* 



RAD



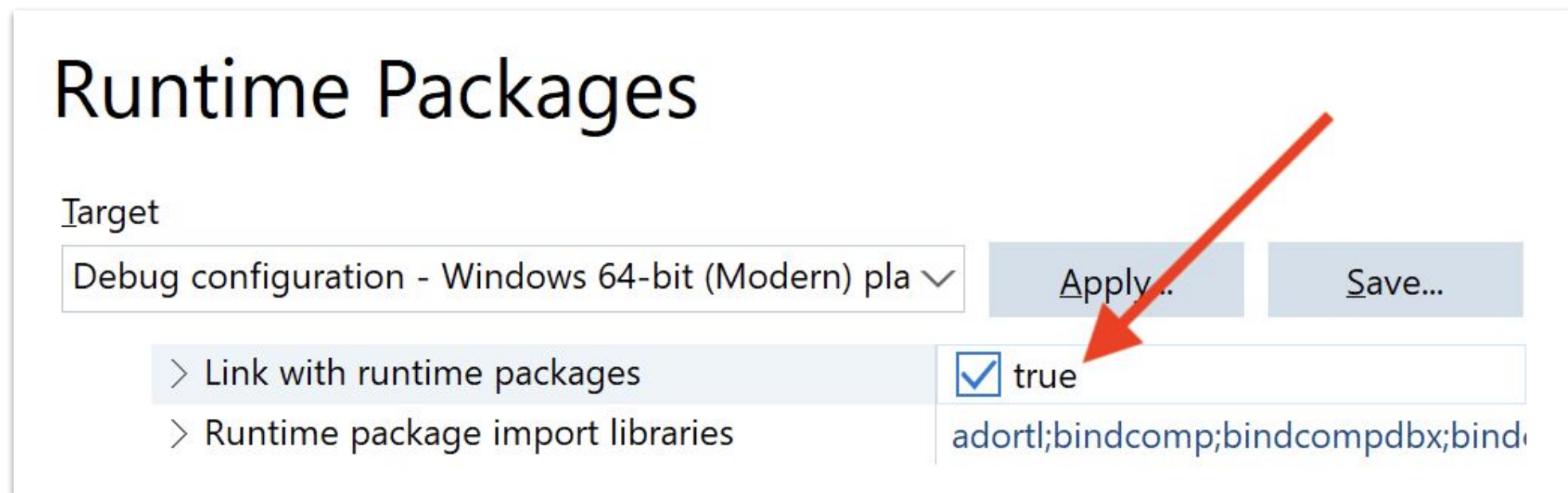
What's New in <RAD Studio 12.2>

bcc64x: the Modernized C++ Compiler

- We shipped 12.1 without consuming dynamic packages (.BPLs) or creating them; you had to link statically. *Now it's complete!*

- **#1 Dynamic Packages!**

- ⌘ Link *dynamically* against BPLs
 - ⌘ Plus *create* packages
(i.e. C++ source code -> a BPL)



- **Benefit:** full feature match vs old toolchain;
split dependencies when building and shipping
- Also lots of **C++ package documentation** going online



RAD



What's New in <RAD Studio 12.2>

bcc64x: the Modernized C++ Compiler

- #2 CMake
- *Coming soon to the GetIt Package Manager*
- Great for building third party libraries
Yes: supports building VCL & FMX apps too
- Benefit: fast compilation (but wait for the next slide)
Easy and compatible building of many, *many* C++ libraries.



RAD



What's New in <RAD Studio 12.2>

bcc64x: #3 Parallel building *speeeeeeeeee*

- 'CPU-saturating multicore builds with --jobs for Windows 64-bit Modern'
- Main 'pain point' for C++ is **compile time**.
It's a slow language to compile, but worth it for the fast apps, power, etc
- Previously:
 - TwineCompile (great tool)
 - CMake & ninja (prev slide; now back)
 - not IDE-integrated, manual setup
 - Batch compilation w '–jobs'
 - not ideal to set up and use



<https://xkcd.com/303/> following terms at <https://xkcd.com/about/>



RAD



What's New in <RAD Studio 12.2>

bcc64x: #3 Parallel building *speeeeee*

- 12.2 brings back '--jobs', but *completely reimplemented*
- And it's **fast**
- And it's **on by default**
- Install 12.2 →
 - immediate**
 - massive**
 - speedup***
 - vs 12.1 defaults



RAD

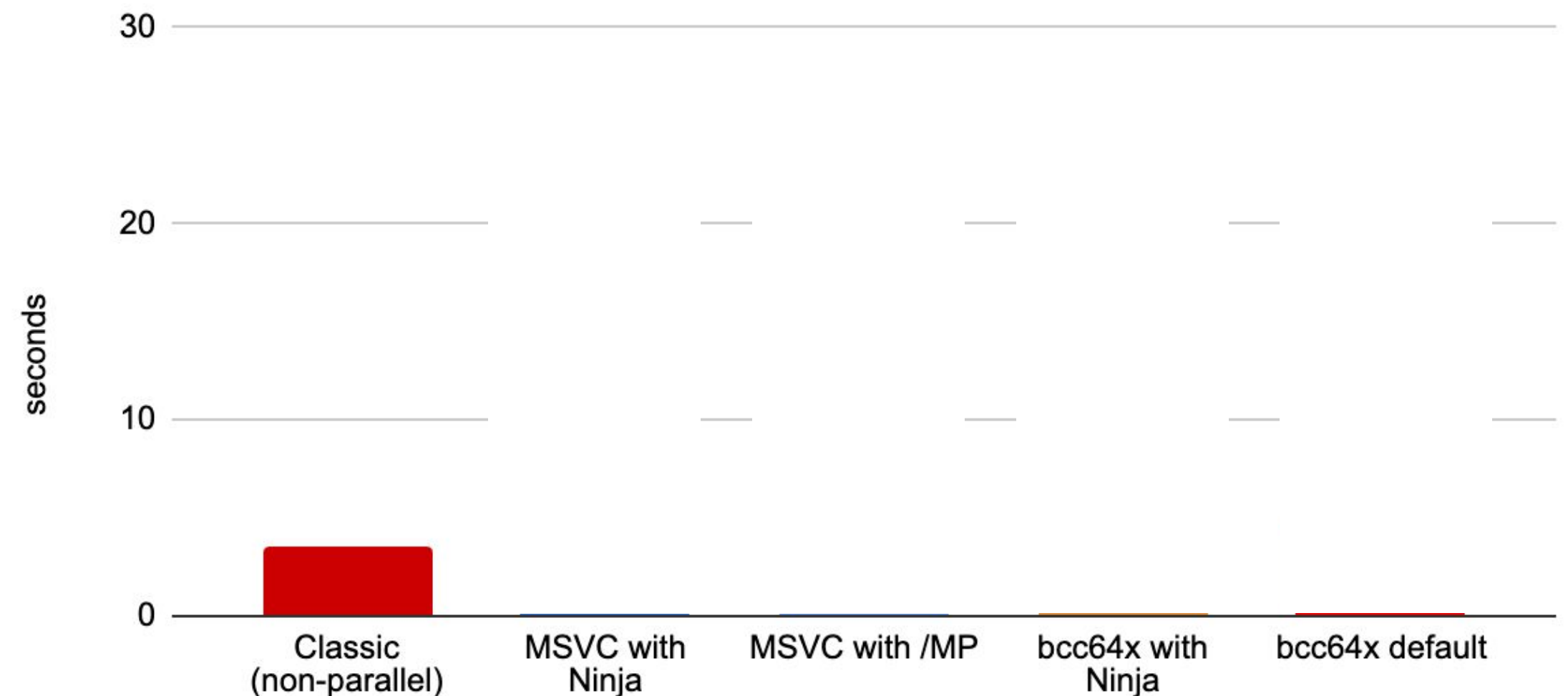


What's New in <RAD Studio 12.2>

bcc64x: #3 Parallel building *speeeeeeeeee*

- 12.2 brings back '--jobs', but *completely reimplemented*
- And it's **fast**
- And it's **on by default**
- Install 12.2 →
immediate
massive
speedup
vs 12.1 defaults

Parallel compilation across all toolchains
(5,000 files, 360,000 lines of code)



Parallel compilation across all toolchains (5K files, 360K LOC)

▪ Measured on a Dell XPS 17 9730, with 13th Gen Intel(R) Core(TM) i9-13900H, 2.60 GHz base speed, 14 cores, 20 logical processors, 64 GB RAM, Windows 11 Pro. Compilers were bcc64x from RAD Studio 12.2 build 29.0.53541.9777, msvc v14.29.30133 and ninja v1.12.1, on 29th of August of 2024. Code was synthetic, and real-world results may vary.



bcc64x: #3 Parallel building *speeeeeeeeee*

- That was a benchmark with *synthetic code*
- Code using the STL heavily compiles slower than code that doesn't:
May be to do with libc++ have many files, and transitive includes.
Or it may be the C++ version (14 vs 17 vs 20)
- We're *still benchmarking*: your code may not match that benchmark, but we aim to provide a guide on getting good compile performance. Expect a *blog* with more info
- Your mileage may vary
- ...but under the *right circumstances* we've seen stats of *noticeably faster than other compilers*



RAD

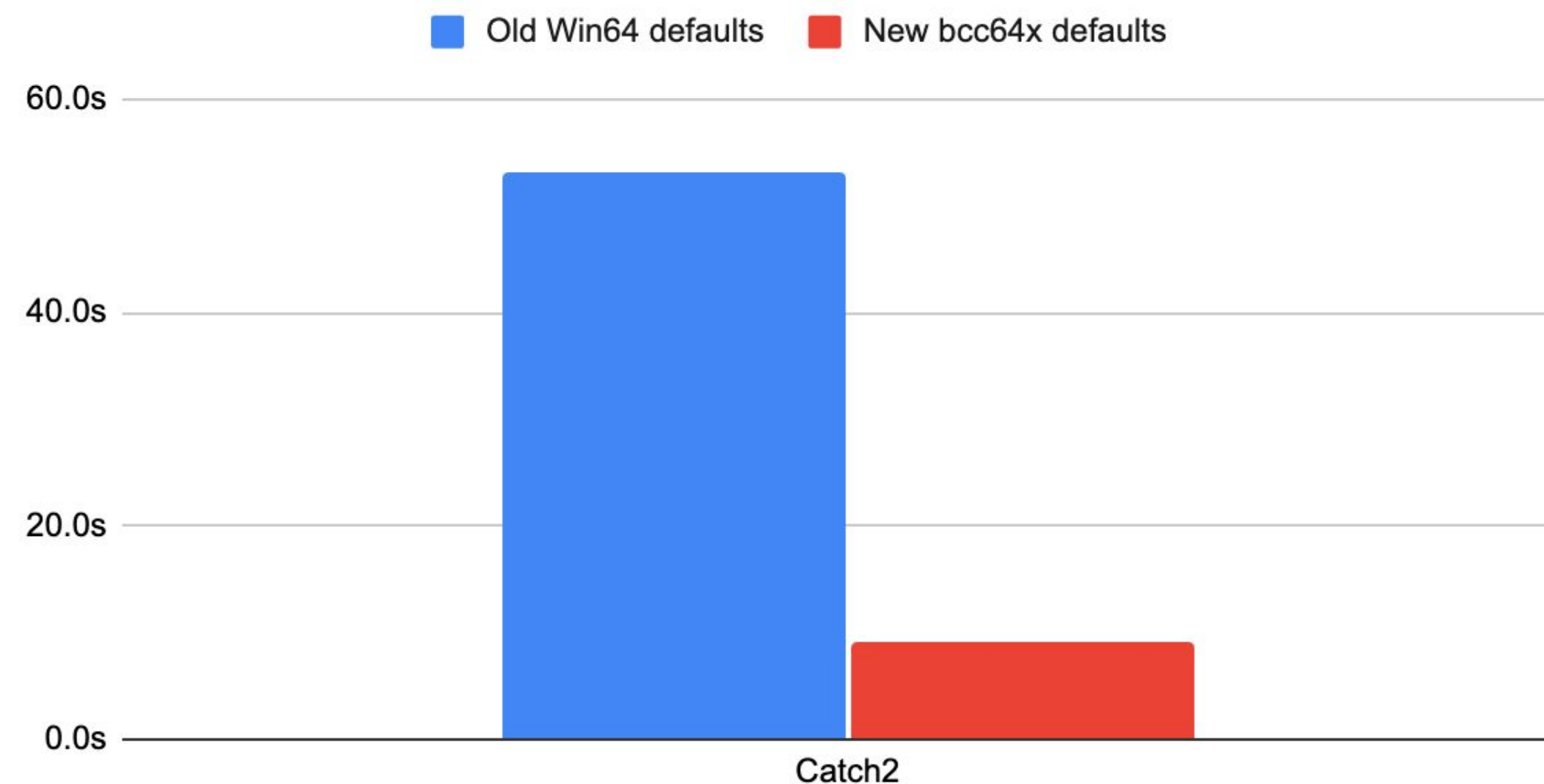


What's New in <RAD Studio 12.2>

bcc64x: #3 Parallel building *speeeeee*

- *Real-world code*: Catch2 library

Old Win64 (default setup) vs New bcc64x defaults on the Catch2 library



- Remember: this is the new *on by default* behaviour

▪ Measured on a Dell XPS 17 9730, with 13th Gen Intel(R) Core(TM) i9-13900H, 2.60 GHz base speed, 14 cores, 20 logical processors, 64 GB RAM, Windows 11 Pro. Compilers were bcc64 (old) and bcc64x from RAD Studio 12.2 build 29.0.53541.9777. Real-world results may vary.



RAD



What's New in <RAD Studio 12.2>

bcc64x: #3 Parallel building *speeeeee*

- 12.2 brings back '--jobs', but *completely reimplemented*
 - And it's **fast**
 - And it's **on by default**
 - Install 12.2 →
immediate
massive
speedup
vs 12.1 defaults
- Just use 12.2 and get it
 - No configuration, it's just there
 - *Upgrade from classic!*
 - It scales with cores and RAM:
any decent machine will see a speedup vs
12.1 defaults!



RAD



What's New in <RAD Studio 12.2>

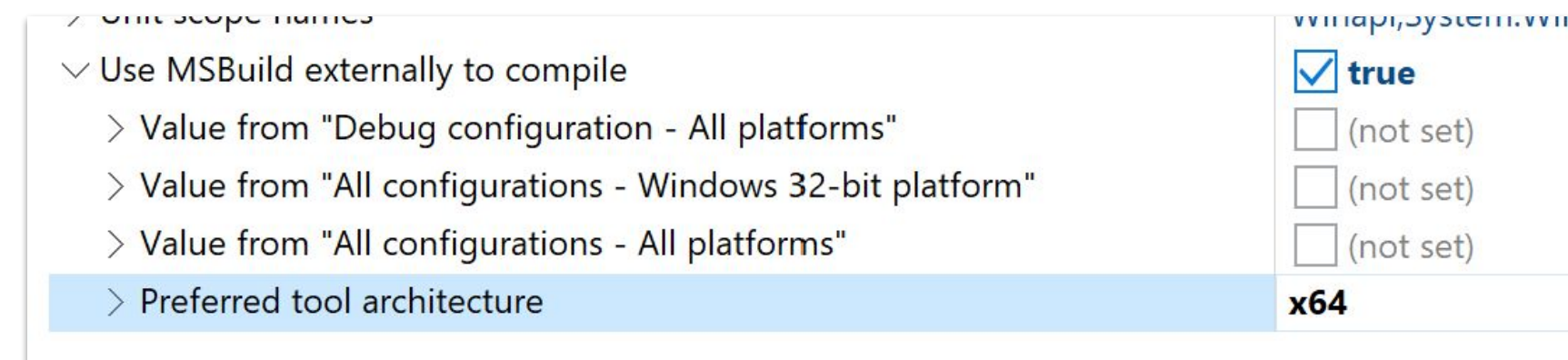
64-bit Version of Delphi Windows compilers

64-bit binaries (large memory address) for Delphi Windows compilers for both Win32 and Win64 **command line compilers**

- Can be enabled in the IDE, when using *external MSBuild*
- Allows building **very large** applications as a single executable
- Available only in **Enterprise** and **Architect**

64-bit binary (large memory address)
version of the **DelphiLSP** executable

- Can be activated in the IDE
- Also available only in Enterprise and Architect



Notice that the new C++ Clang compiler has been a 64-bit binary since release



RAD



What's New in <RAD Studio 12.2>

Smart CodeInsight: AI Powered Coding

Use **AI LLMs for coding** in the RAD Studio IDE

- An **open architecture** for AI plugins

Ready-to-use support for **three online solutions**

- OpenAI
- Gemini by Google
- Claude by Anthropic

and one **offline solution**

- Ollama

Ready to use docker image, works in WSL2





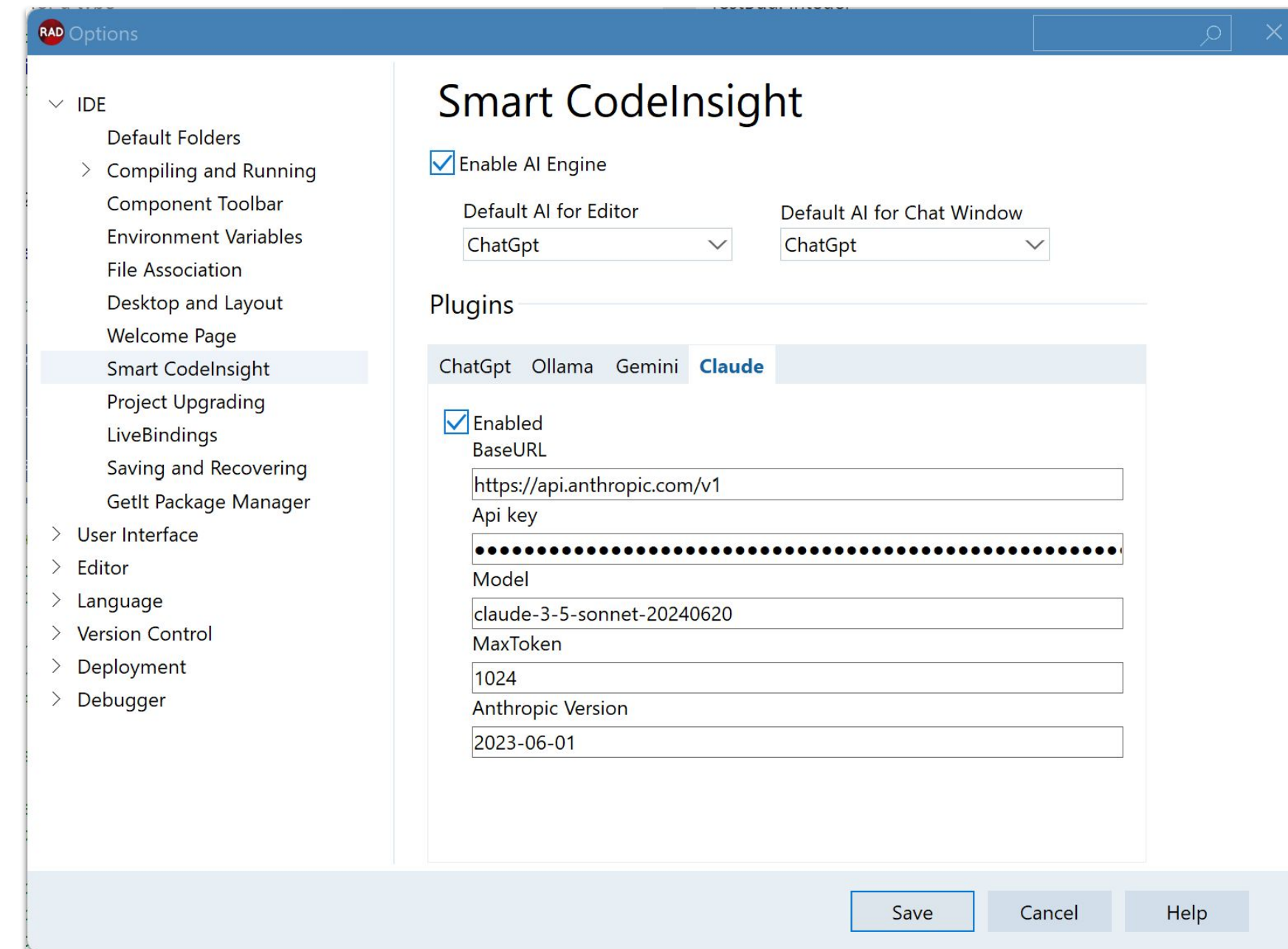
RAD



What's New in <RAD Studio 12.2>

Smart CodeInsight: AI Powered Coding

- Everything is **optional** and **disabled** by default
- **Multiple** providers, so you can **choose** which one or ones **you want** to enable
- A locally installed, **offline** solution for **maximum privacy**
- You decide **if** and **how** to use **AI**
... and you can even **create additional plugins** for your preferred solution





RAD



What's New in <RAD Studio 12.2>

Smart CodeInsight: AI Powered Coding

IDE: chat window and editor commands

RAD Project21 - RAD Studio 12 - Unit25

File Edit Search View Refactor Project Run Component

Smart CodeInsight

Answer>

In Delphi, "strict private" is a visibility specifier used in object-oriented programming to control access to class members (fields, methods, properties). It is one of the most restrictive visibility levels available in Delphi. Here's what you need to know about "strict private":

1. Scope: Members declared as "strict private" are only accessible within the class where they are declared.
2. Inheritance: Strict private members are not inherited by subclasses and cannot be accessed by descendant classes.
3. Unit scope: Unlike regular "private" members, strict private members are not accessible even within the same unit where the class is defined.
4. Usage: It's typically used for implementation details that should be completely hidden from outside the class and its descendants.

Claude>What's the meaning of strict private in Delphi?

Claude

Structure Smart CodeInsight

```
26 function TestBug: Integer;
  var
    P: Pointer;
  begin
30   GetMem(P, 100);
    Result := 100;
  end;
  {***** Find Bugs *****}
  1. Memory Leak: The allocated memory isn't freed after been used. After calling GetMem
  the programmer should call FreeMem(P) before the function ends to avoid leaking memory
  2. Uninitialized pointer: The allocated memory is not initialized. In order to avoid
  it is recommended to initialize the memory after allocating it. The allocated memory
  40 3. Unused pointer: The pointer 'P' was allocated memory but it's not used anywhere in
  4. No error handling: The function GetMem may fail if the system cannot provide the re
  This failure is not handled in the code.

  The improved version of the code could look like this:
  ``delphi
  function TestBug: Integer;
  var
50   P: Pointer;
  begin
    GetMem(P, 100);
    try
      // use 'P' for some operations here...
      // ...
      FillChar(P^, 100, 0);
      Result := 100;
    finally
      FreeMem(P);
60   end;
end;
  }
```

Smart CodeInsight

- Open File at Cursor Ctrl+Enter
- Topic Search F1
- Complete Class at Cursor Ctrl+Shift+C
- Add To-Do Item... Ctrl+Shift+T
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Toggle Bookmarks
- Goto Bookmarks
- Clear Bookmarks
- Debug
- Toggle Comment Ctrl+/
- Fold
- Unfold
- Refactor
- Find
- Find Symbols
- Sync Prototypes Ctrl+Alt+Shift+P
- Surround
- Add to Repository
- View Messages
- Read Only
- Editor Options

AI Chat

- Find Bugs
- Explain the Code
- Add Comment
- Complete the Code
- Optimize Code
- Add Unit Test
- Convert to Assembly
- Convert to Delphi
- Convert to C++



RAD




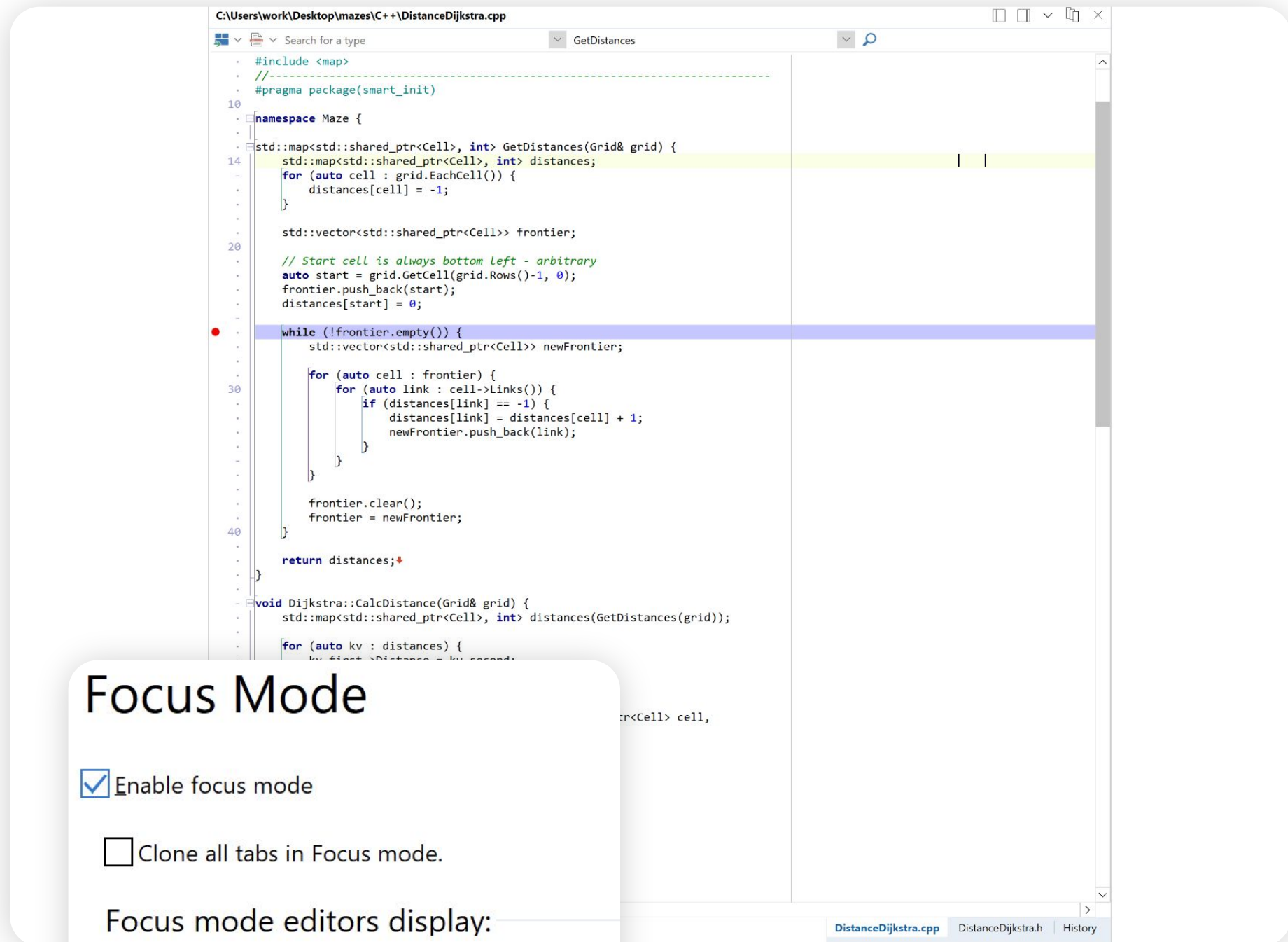
What's New in <RAD Studio 12.2>

Editor: Focus Mode

Progression from 12.1's split views, and known as 'zen mode' in other IDEs

Focus Mode leaves the IDE with **only your code editor**

- Removes other UI chrome
- Multiple side by side editors
- 'Code only'
- Distraction free... *focus!* 



Focus Mode

☒ Enable focus mode

☐ Clone all tabs in Focus mode.

Focus mode editors display:

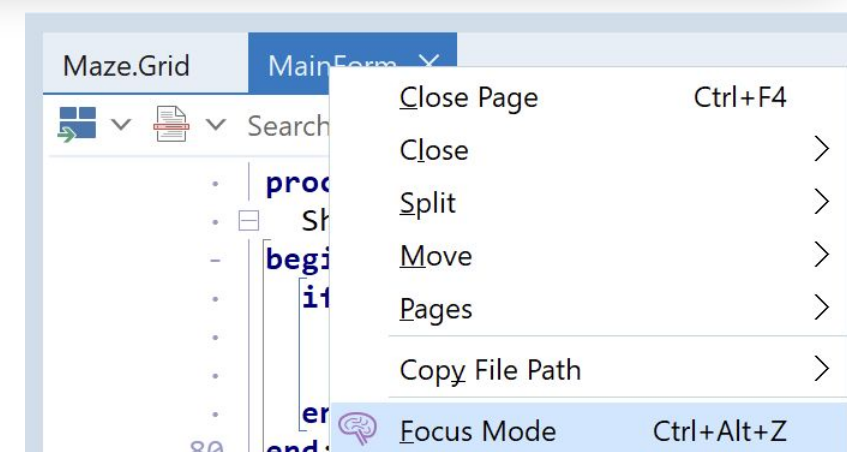
☒ Bookmarks

☐ Right margin

☐ Navigation toolbar

☐ Code folding

☒ Debug Info (Breakpoints, Compiled lines)





RAD



What's New in <RAD Studio 12.2>

Z:\projects\Embarcadero\mazes\mazes\Delphi\MainForm.pas

Search for a type

TForm1.RecreateGrid

```
procedure TForm1.edtCellSizeKeyUp(Sender: TObject; var Key: Word; var KeyChar: Char;
  Shift: TShiftState);
begin
  if Key = vkReturn then begin
    RecreateGrid;
    PaintGrid;
  end;
80 end;

procedure TForm1.FormShow(Sender: TObject);
begin
  PaintGrid;
end;

procedure TForm1.RecreateGrid;
var
  R, C : Integer;
90 begin
  C := Trunc(ImageViewer.Width / CellSize) - 1;
  R := Trunc(ImageViewer.Height / CellSize) - 1;

  // Limit the small size
  C := Max(C, 2);
  R := Max(R, 2);

98   FGrid := CreateGrid(R, C);
end;
100
function TForm1.CellSize : Integer;
begin
  Result := 50;
  TryStrToInt(edtCellSize.Text, Result);
end;

procedure TForm1.PaintGrid;
var
  CellDistPainting : TCellPainting;
110 begin
  CellDistPainting := TCellPainting(cmbDistance.ItemIndex);
  ImageViewer.Bitmap.Assign(FGrid.Repaint(CellSize, CellDistPainting));
  Invalidate;
end;

procedure TForm1.btnNewClick(Sender: TObject);
begin
  RecreateGrid;
  PaintGrid;
120 end;

function TForm1.GetMazeGenerationProc : TMazeGenerationProc;
begin
  Result := nil;
  case cmbAlgorithm.ItemIndex of
    0 : Result := BinaryTree;
    1 : Result := AldousBroder;
    2 : Result := RecursiveBacktracker;
  end;
130 end;

procedure TForm1.btnPlayClick(Sender: TObject);
var
```

Code History



RAD



What's New in <RAD Studio 12.2>

```
Z:\projects\Embarcadero\mazes\mazes\Delphi\MainForm.pas
TForm1.RecreateGrid

10
type
  TForm1 = class(TForm)
    ToolBar: TToolBar;
    btnNew: TSpeedButton;
    cmbDistance: TComboBox;
    btnPlayStepped: TSpeedButton;
    btnPlay: TSpeedButton;
    ImageViewer: TImageViewer;
    edtCellSize: TEdit;
    btnStop: TSpeedButton;
    cmbAlgorithm: TComboBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    procedure btnNewClick(Sender: TObject);
    procedure edtCellSizeKeyUp(Sender: TObject; var Key: Word; var KeyChar: Char;
      Shift: TShiftState);
    procedure FormShow(Sender: TObject);
    procedure btnPlaySteppedClick(Sender: TObject);
    procedure btnPlayClick(Sender: TObject);
    procedure btnStopClick(Sender: TObject);
    procedure cmbDistanceChange(Sender: TObject);
  private
    FGrid : IGrid;
    FStopPlaying : Boolean;
    procedure RecreateGrid;
    procedure PaintGrid;
    function CellSize: Integer;
    procedure DoDijkstra;
    function GetMazeGenerationProc: TMazeGenerationProc;
  public
    { Public declarations }
    constructor Create(AOwner : TComponent); override;
    destructor Destroy; override;
  end;

  var
    Form1: TForm1;

50 implementation

  {$R *.fmx}

  uses
    System.Math, Maze.DistanceDijkstra;

  { TForm1 }

  constructor TForm1.Create(AOwner: TComponent);
60 begin
    inherited;
    FStopPlaying := false;

    RecreateGrid;
  end;

  destructor TForm1.Destroy;
  begin
70 inherited;
```

```
Z:\projects\Embarcadero\mazes\mazes\Delphi\MainForm.pas

procedure TForm1.edtCellSizeKeyUp(Sender: TObject; var Key: Word; var KeyChar: Char;
  Shift: TShiftState);
begin
  if Key = vkReturn then begin
    RecreateGrid;
    PaintGrid;
  end;
80 end;

procedure TForm1.FormShow(Sender: TObject);
begin
  PaintGrid;
end;

procedure TForm1.RecreateGrid;
var
  R, C : Integer;
90 begin
  C := Trunc(ImageViewer.Width / CellSize) - 1;
  R := Trunc(ImageViewer.Height / CellSize) - 1;

  // Limit the small size
  C := Max(C, 2);
  R := Max(R, 2);

98 FGrid := CreateGrid(R, C);
end;

100 function TForm1.CellSize : Integer;
begin
  Result := 50;
  TryStrToInt(edtCellSize.Text, Result);
end;

procedure TForm1.PaintGrid;
var
  CellDistPainting : TCellPainting;
110 begin
  CellDistPainting := TCellPainting(cmbDistance.ItemIndex);
  ImageViewer.Bitmap.Assign(FGrid.Repaint(CellSize, CellDistPainting));
  Invalidate;
end;

procedure TForm1.btnNewClick(Sender: TObject);
begin
  RecreateGrid;
  PaintGrid;
120 end;

function TForm1.GetMazeGenerationProc : TMazeGenerationProc;
begin
  Result := nil;
  case cmbAlgorithm.ItemIndex of
    0 : Result := BinaryTree;
    1 : Result := AldousBroder;
    2 : Result := RecursiveBacktracker;
  end;
130 end;

procedure TForm1.btnPlayClick(Sender: TObject);
var
```



RAD



What's New in <RAD Studio 12.2>


Editor: Scrollbar Annotations

- Editor Scrollbar Annotations
 - See at a glance sections of code modified, with bookmarks, errors, warnings...
- Delphi Inline Assembly Highlighting
- Delphi LSP: code completion optionally in editor colors
- Split Editor Improvements around resizing, stability and IDE toolbars in high DPI quality

```
distances[start] = 0;

while (!frontier.empty()) {
    std::vector<std::shared_ptr<Cell>> newFrontier;

    for (auto cell : frontier) {
        for (auto link : cell->Links()) {
            if (distances[link] == -1) {
                distances[link] = distances[cell] + 1;
                newFrontier.push_back(link);
            }
        }
    }
}
```



```
procedure TForm1.FormCreate(Sender: TObject);
begin
    Self.|
end;
end.
```

procedure FormCreate(Sender: TObject);
procedure Arrangelcons;
procedure Cascade;
procedure Next;
procedure Previous;
procedure Tile;
property ActiveMDIChild: TForm;

✓ Tab Enter — ↑ ↓ ✗ Space " # \$ ' () * + , - . : ; < = > [] { } Esc →

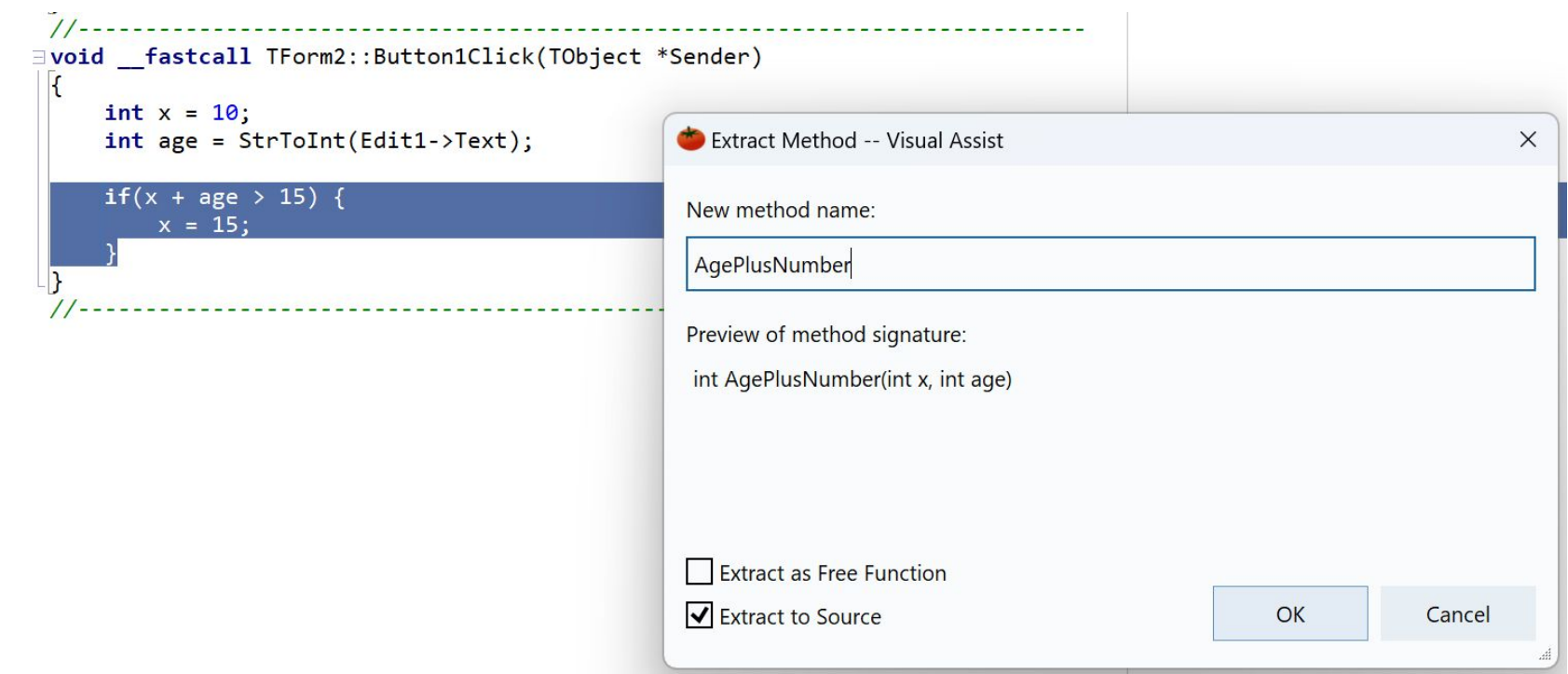
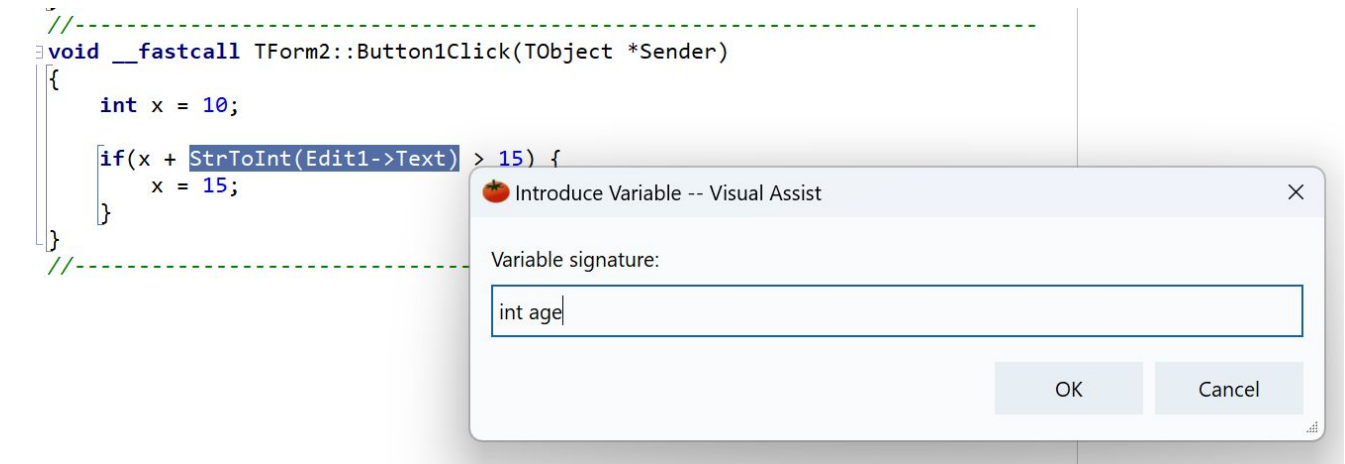


C++Builder Visual Assist Integration

The C++Builder **Visual Assist** integration has been expanded with three new refactorings

- Introduce Variable
- Create From Usage
- Extract Method

Plus **significant** quality and stability improvements



whole tomato

Visual Assist



RAD

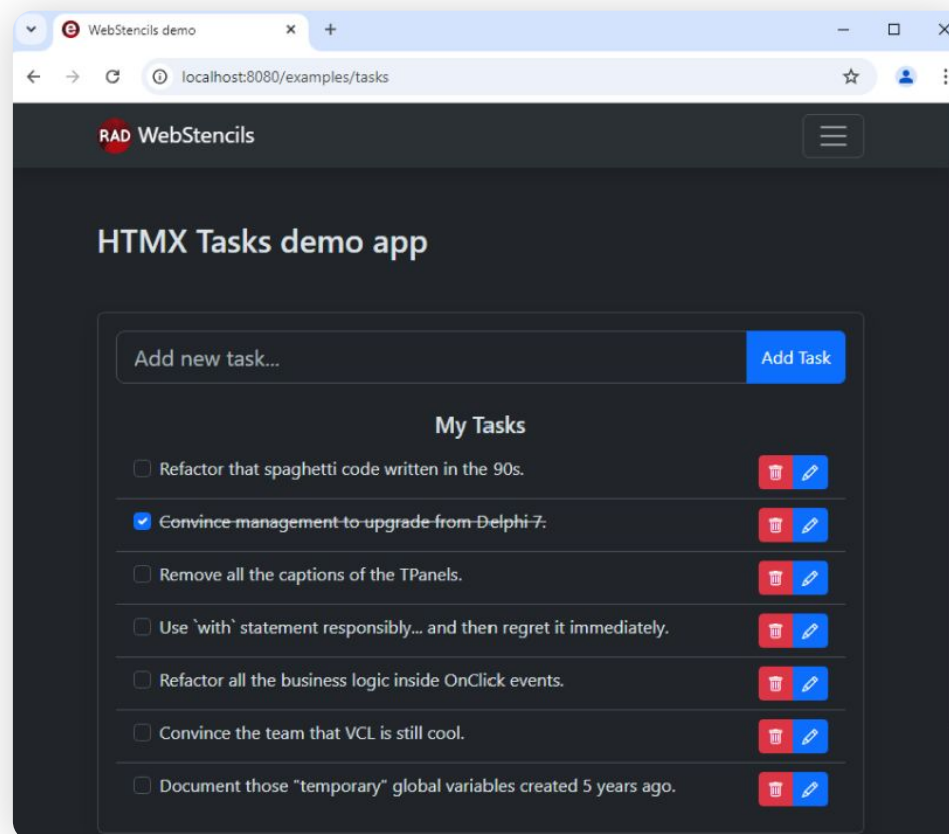


What's New in <RAD Studio 12.2>

WebStencils Template Library

Server-side script-based integration and processing of HTML files for WebBroker and RAD Server. **You** can develop modern websites

- Based on **any JavaScript library**, CSS, and web technology
- Powered by the data extracted and processed by a RAD Studio **server-side application**



- Foundation for **HTMX** as a web development solution.
- HTMX pages benefit from server-side code generation and hook into REST servers for content updates



RAD



What's New in <RAD Studio 12.2>

WebStencils Template Library

Core Web Scripting Syntax

- Uses **@** for special processing of symbols and **{ }** to define blocks
- **@object.value** to read data from server objects
- **@keywords** for special processing
- Supports simple expressions

```
ProcessorGeneric.AddVar( 'obj2',  
    TSimpleObject.Create( 'marc', 9 ) );
```

```
<h4>Obj2</h4>  
  
@if obj2.ValueBelowTen {  
    <p>@obj2.name <span> has a value of </span>  
    @obj2.value, which is under 10</p>  
}
```

Obj2

marc has a value of 9, which is below 10



RAD

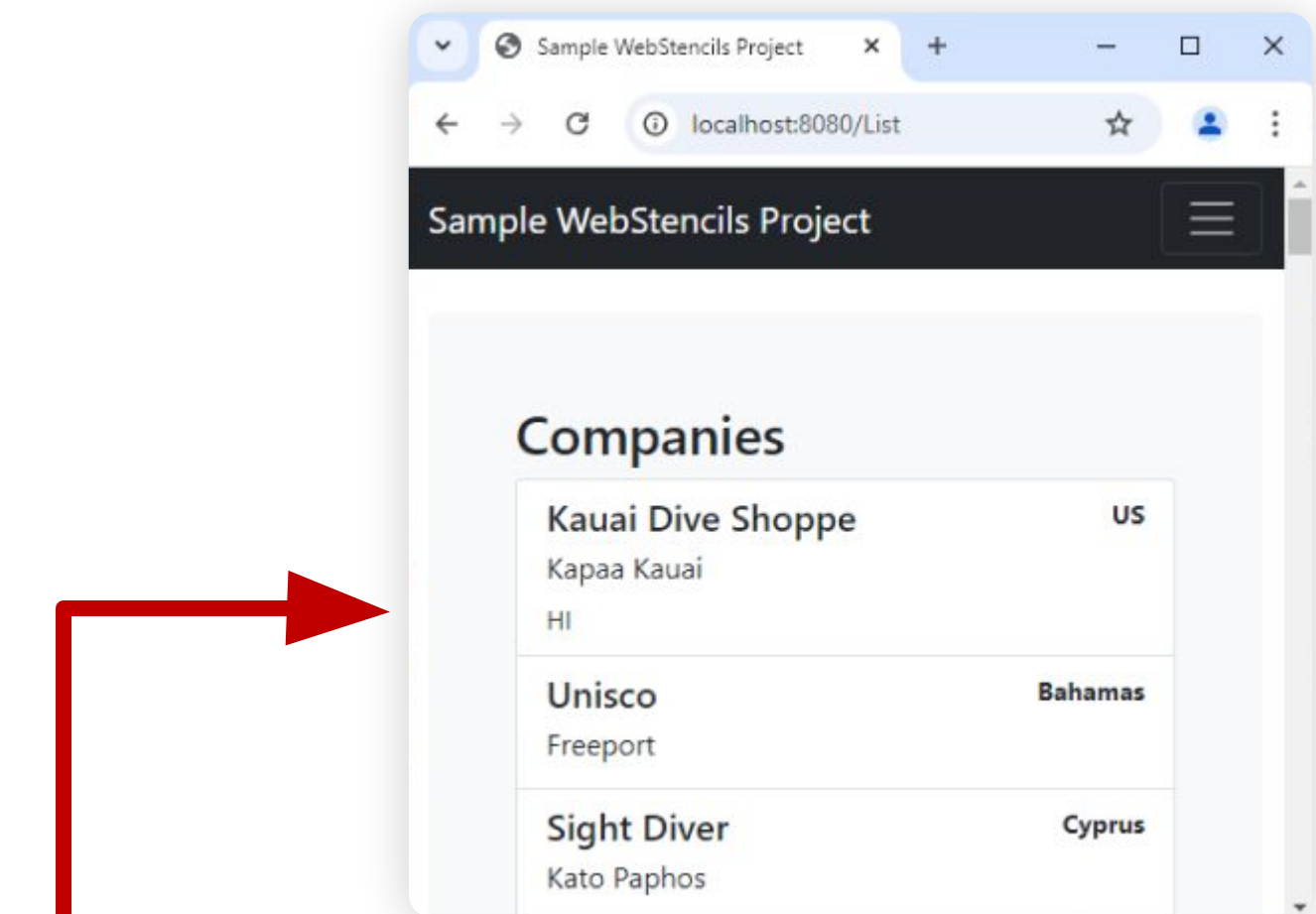


What's New in <RAD Studio 12.2>

WebStencils Template Library

Data access for the script

- RTTI based (but not only)
- `@object.value`
- `OnValue` event handler
- Use objects, lists, or datasets registered as **script variables**
- Define modules variables with attributes



```
<div class="list-group w-65">
  @foreach dataset {
    <a href="/company?id=@loop.custno" class="list-group-item
      list-group-item-action" aria-current="true">
      <div class="d-flex w-100 justify-content-between">
        <h5 class="mb-1">@loop.company</h5>
        <small><b>@loop.country</b></small>
      </div>
      <p class="mb-1">@loop.city</p>
      <small>@loop.state</small>
    </a>
  }
</div>
```




RAD

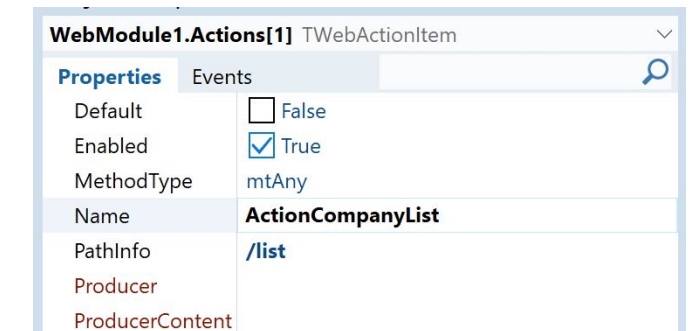


What's New in <RAD Studio 12.2>

From Simple WebBroker Scenario to Mapping

WebBroker Actions

- Connect action to WebStencilsProcessor
- Use OnAction handler to:
 - Initialize the processor script variables
 - Run the processor
 - Return the resulting HTML
- In the script, use the variables to generate the HTML



```
• procedure TWebModule1.WebModule1WebActionItem1Action( Sender : TObject;  
• Request : TWebRequest; Response : TWebResponse; var Handled : Boolean );  
• begin  
200 ClientDataSet1.Open;  
• ProcessorCompanyList.AddVar( 'dataset', ClientDataSet1, False ); // do not destroy  
• Response.Content := ProcessorCompanyList.Content;  
• ClientDataSet1.Close;  
• Handled := True;  
- end;
```

```
@foreach dataset {  
  
  <a href="/company?id=@loop.custno" class="list-group  
  <div class="d-flex w-100 justify-content-between">  
    <h5 class="mb-1">@loop.company</h5>  
    <small>@loop.country</small>  
  </div>  
  <p class="mb-1">@loop.city</p>  
  <small>@loop.state</small>  
</a>  
  
}
```

Both WebBroker and RAD Server offer a mapping layer to associate a URL with a file script



RAD



What's New in <RAD Studio 12.2>

WebStencils Keywords

@query - access HTTP parameters

@* .. *@ - comment

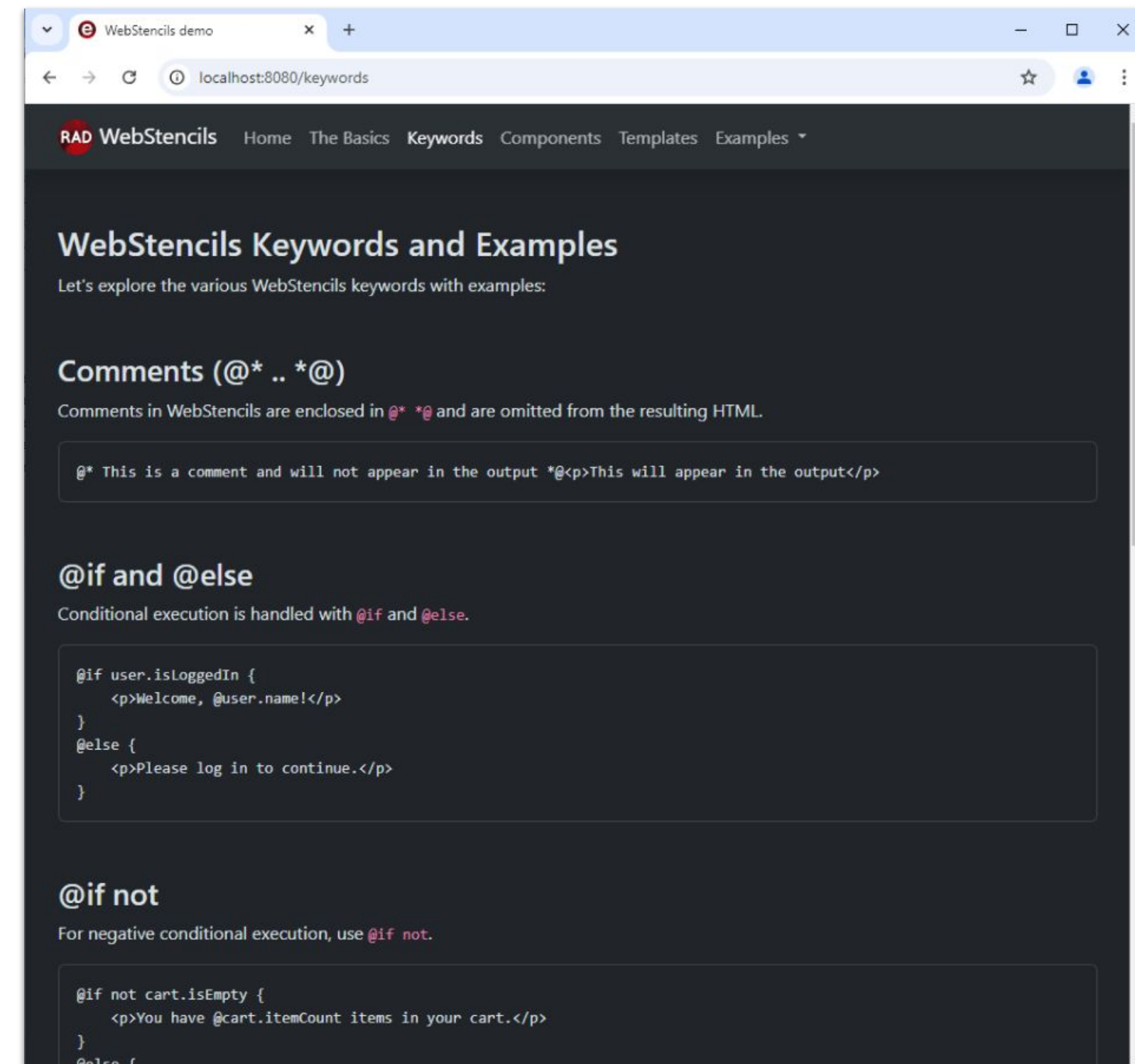
@if [not] object.value { ... } [@else { ... }]

@ForEach (var object in list) { ... }

@ForEach object { ... }

@Import

@Scaffolding





RAD



What's New in <RAD Studio 12.2>

WebStencils Template structure

```
<body data-bs-theme="dark">
  @Import layouts/navBar
  @RenderBody
  @Import layouts/footer
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@
    integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLES
    crossorigin="anonymous"></script>
</body>
```

```
<li class="list-group-item list-group-item-action d-flex" id="task-@Task.Id">
  <input class="form-check-input me-2"
    hx-put="/tasks/toggleCompleted?id=@Task.Id"
    hx-target="#task-@Task.Id"
    hx-swap="outerHTML"
    type="checkbox"
    id="Checkbox-@Task.Id"
    @if ( Task.completed ) { checked }>
  <span class="task-text @if( Task.completed ) { text-decoration-line-through }"
    hx-trigger="click"
    hx-put="/tasks/toggleCompleted?id=@Task.Id"
    hx-target="#task-@Task.Id"
    hx-swap="outerHTML">@Task.Description</span>
```

@LayoutPage

Indicates which template file to use

@RenderBody

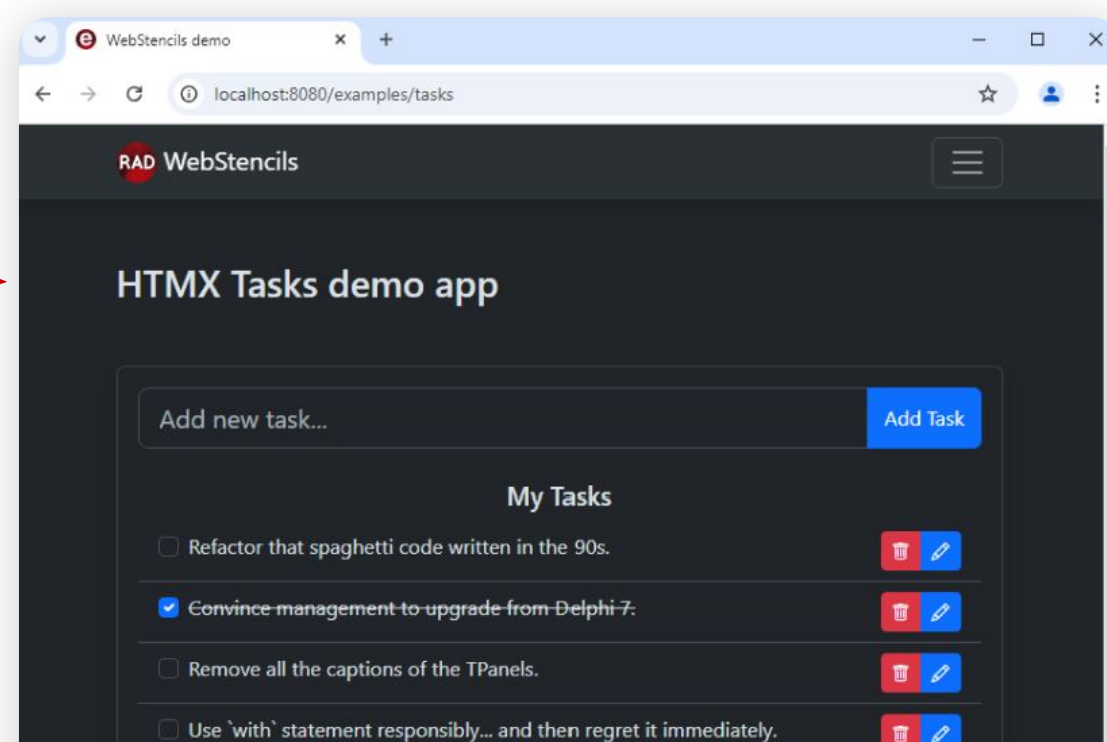
Placeholder in a template file where to place actual content

@ExtraHeader { ... }

Optional block of code to be added as extra header information

@RenderHeader

Where to add the extra header of the page





RAD



What's New in <RAD Studio 12.2>

WebStencils Template Library Components

TWebStencilsProcessor

Processes an individual file

Standard WebBroker Producer

Properties InputFilename, InputLines...

TWebStencilsEngine

Provides shared settings to individual

TWebStencilsProcessor components

Create WebStencilsProcessor comp, if needed

Properties: Dispatcher of type **IWebDispatch**,

RootDirectory, **DefaultFileExt**...

#	Company	Last Name	First Name	City
1	Realfire	Birdseye	Carmine	Pittsburgh
2	Nlounge	Andree	Josy	Bobigny
3	Browsezoom	Doreward	Korrie	Kalanchak
4	Quinu	Tomeo	Karyn	Kasli
5	Skiptube	Salleir	Alia	Añelo
6	Cogibox	Hallard	Harley	Xinfengjie
7	Layo	Maunton	Dukey	Sadabe
8	Skyba	Ridout	Fowler	Jeding
9	Pixonyx	Earney	Janessa	Maharagama
10	Trilia	Daybell	Jaclyn	Qa'en
11	Yotz	Dunlea	Ase	Lumbang
12	Gabspot	Beccera	Mariam	Simnas
13	Edgetag	Croom	Agata	Sidomukti
14	Wikibox	Tomasino	Roxy	Maoqitun
15	Yata	Feldbaum	Kirsteni	El Corozo



RAD



What's New in <RAD Studio 12.2>

Improved Mobile Development

Significant improvements for Delphi mobile development

Apple platforms

- Linker updated to the most recent Apple version
- Link iOS static libraries written in Xcode, like Firebase iOS SDK

Android support

- Options to compress Dex files and compress native library files
- The targetSdkVersion of the manifest file can be configured in the IDE
- Import Android libraries ('.aar' files)
- Works with the Facebook SDK for Android and Firebase Analytics

Quality in 12.2

Visible improvements
in all product areas





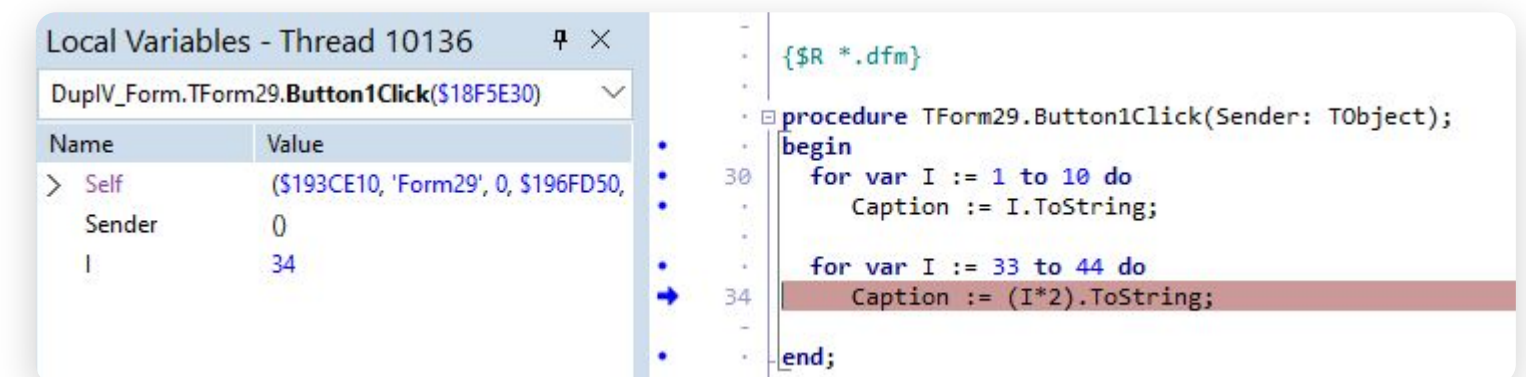
RAD



What's New in <RAD Studio 12.2>

Debugger Quality

- Several improvements in the various RAD Studio debuggers
- Delphi debugger handles repeated inline variables with the same name
- Improved the debugger error handling and the remote debuggers communication timeouts.





RAD



What's New in <RAD Studio 12.2>

DelphiLSP Quality

The DelphiLSP engine improvements

- **Symbol Hints** Hints work on most symbols in the class interface, including fields (was not available in 12.1)
- **Local Constants and Variables** Hovering over local constants and variables can now resolve their definitions
- **Forwarded Type Declarations** The engine now resolves forwarded type declarations in some cases

```
1 unit Unit2;
.
. interface
.
. uses
.   Winapi.Windows, Winapi.Messages, System.SysUtils, SysUtils,
.   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Menus;
.
. type
10 TForm2 = class(TForm)
.   MainMenu1: TMainMenu;
.   pr var TForm2.MainMenu1: TMainMenu - Unit2.pas (11)
.   public
.     { Public declarations }
.   end;
.
. var
.   Form2: TForm2;
20
. implementation
.
.   {$R *.dfm}
.
. end.
```



Delphi RTL, VCL, and FireMonkey Quality

Delphi RTL

- New **TParallelArray** class (offers faster parallel operations, like sorting)
- New **TOrderedDictionary** class
- Support for the **REG_NONE** type in **TRegistry** (key with no type)

VCL

- Primarily quality work around **frames** scaling, **FormsTabBar**, styling issues, High DPI issues. *Quite extensive work.*

FireMonkey

- Further work on Android deployment, intensive quality work for many areas, with additional focus on edit and memo controls on mobile



RAD Server Improvements

- **WebStencils** template engine support 🌋 🎉 *(RAD Server can do Web!)*
- Reworked RAD Server Push Notification (and Firebase APNS)
 - Added required support for HTTP/2 (uses libCurl also on Win)
- RAD Server **Swagger Documentation** enhancements
 - New **IEMSApiDocumentService** interface (customization of the metadata in the generated swagger doc files)
 - New **IEMSApiDocumentFilter** interface (filter the API being documented)
 - Swagger support for YAML format in addition to JSON
 - **APIDocAttributes** demo updated to use Delphi multiline string literals



Quality Overall

Customer reports fixed

- Addressed approximately 350 customer reported issues
- Implemented 28 customer feature requests
- Reported on the old or the new portal

Addressed issues in all areas, the top ones are

- | | |
|--------------|-------------------|
| ○ IDE | ○ Compiler\C++ |
| ○ VCL | ○ RTL\Delphi |
| ○ FireMonkey | ○ Compiler\Delphi |



What's New in <RAD Studio 12.2>

Thank You!

fernando.rizzato@embarcadero.com