

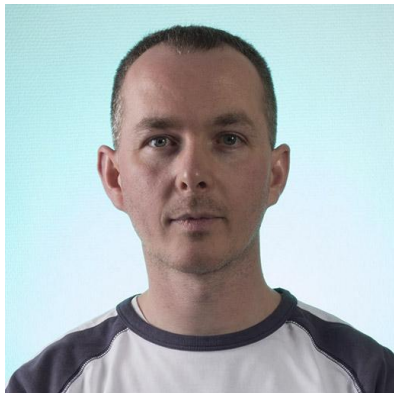


RAD Studio 13

Nouveautés de la VCL



Qui suis-je ?



Patrick PREMARTIN

Formateur Delphi et web, développeur freelance, streamer.
MVP Embarcadero

Pour me contacter :

<https://www.linkedin.com/in/patrickpremartin/>

<https://olfsoftware.fr>

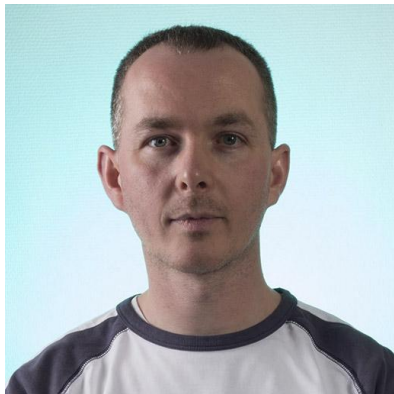
Aussi visible sur Twitch :

<https://www.twitch.tv/patrickpremartin>

Projets open source :

<https://github.com/DeveloppeurPascal>

Quelques uns de mes sites



Patrick PREMARTIN

Catalogue des livres sur Delphi et le Pascal

<https://delphi-books.com>

Blog sur le développement en Pascal sous Delphi

<https://developpeur-pascal.fr>

Auto apprentissage de Delphi en VOD

<https://apprendre-delphi.fr>

Utilitaires pour développeurs Delphi et C++Builder

<https://getitnow.embarcadero.com/vendor/patrick-premartin/>

Visual Component Library (VCL)

La VCL est la librairie de composants visuels (et non visuels) fournie comme framework et type de projet dans Delphi depuis son lancement en 1995.

Cet ensemble de composants et de librairies fournit un accès le plus proche possible à Windows, ses APIs et ses SDKs pour développer des logiciels à destination de Windows.

Encore aujourd'hui la majorité des projets sous Delphi sont des projets VCL. L'environnement de développement de RAD Studio en est également un.

Comme à chaque nouvelle version Windows bénéficie d'une attention particulière.

Quelles versions de Windows pour nos programmes ?

Dans la version 13 Florence de Delphi et C++Builder les projets VCL ciblent essentiellement Windows 10, Windows 11, Windows Server 2022 et 2025.

Selon les éléments utilisés dans les projets il était encore possible de compiler un programme Windows 32 bits et l'exécuter sous Windows XP jusqu'à cette nouvelle version. Ce n'est plus faisable sans bidouiller car Embarcadero utilise maintenant UI Automation de Microsoft qui n'a pas de correspondance sur ce vieux système d'exploitation.

Dans cette version Embarcadero a également fait du ménage et retiré ce qu'il restait de code pour assurer la compatibilité avec Windows 98.

SDKs et APIs de Windows

Dans RAD Studio 13 Florence les projets VCL supportent les SDK de Windows 7, 10 et 11 dont les mappages en Pascal ont été mis à jour aussi bien en 32 qu'en 64 bits.

Les mises à jour de SDKs et APIs Windows représentent une cinquantaine d'interfaces pour WinRT et plus de 300 unités (plus de 40 Mo de codes sources) pour le reste.

S'il vous manque quelque chose dans les unités installées en standard, vous le trouverez dans le paquet "WinMD" téléchargeable gratuitement depuis GetIt.

Accessibilité et tests de logiciels automatisés

Microsoft a introduit UI Automation afin d'améliorer l'expérience utilisateur avec des lecteurs d'écran.

Embarcadero a fait les modifications nécessaires dans le coeur de la VCL et ses composants pour le prendre en charge et améliorer l'accessibilité de nos projets (et de l'IDE).

UI Automation peut également servir pour automatiser les interactions avec les programmes ce qui est utilisé par des logiciels de tests de logiciels. Vous pouvez les utiliser pour vous assurer des non régressions et de la cohérence de vos GUI.

Utiliser le navigateur Edge dans nos projets

Les API de Edge WebView 2 ont été mises à jour.

Le SDK de Microsoft est à télécharger sur GetIt (ou chez l'éditeur). Vous en avez besoin pour le développement mais aussi chez vos utilisateurs.

L'utilisation des composants TWebBrowser et TEdgeBrowser bénéficie des nouveautés par une simple recompilation.

Les démos installables en option ont été mises à jour pour montrer l'utilisation des nouvelles propriétés.

Utiliser le navigateur Edge dans nos projets

Si vous ne connaissez pas le composant TEdgeBrowser et le fonctionnement du SDK Edge WebView 2 de Microsoft regardez la présentation qu'en a fait Grégory Bersegeay en mars 2021.

<https://www.youtube.com/watch?v=bVgk6LdmNGQ>

WinUI 3 a été mis à jour

Les unités liées à Windows App SDK ont été mis à jour.

Un exemple est disponible avec les “samples” installables en option.

Une démo plus complète de l'utilisation de WinUI 3 est disponible au téléchargement sur GetIT.

Pour les tester vous devrez avoir installé les dépendances demandées par ce SDK de Microsoft. Lisez le fichier RTF joint aux projets pour les étapes à suivre.

Mises à jour de la barre de titre des fenêtres

La barre de titre des fenêtres Windows supporte maintenant l'application d'un style VCL. Il suffit de l'activer sur la propriété `CustomTitleBar.StyleColors` de la fiche.

Ca s'applique au fond de la barre de titre, aux boutons et aux composants (TButton, TSpeedButton, TCheckBox, TRadioButton, TToolBar, TEdit, TComboBox, TFormTabsBar, TActionToolBa et TActionMainMenuBar) qu'on y placerait par l'intermédiaire d'un TTitleBarPanel.

La propriété Hint des boutons présents sur une barre de titre est maintenant fonctionnelle.

Mises à jour aussi pour les applications MDI et la prise en charge du HighDPI.

Mises à jour de la barre de titre des fenêtres

Si vous ne connaissez pas le composant TTitleBarPanel et la personnalisation de la barre de titre d'une fenêtre en VCL, consultez la présentation que Paul Toth en a fait en mars 2021.

Version courte en musique :

<https://www.youtube.com/watch?v=DnzUJLZGcl4>

Version complète :

<https://www.youtube.com/watch?v=VnCN-6l1Ed8>

Les listes de contrôles

On pouvait afficher une infobulle au niveau des listes de contrôles (TControlList). Il est maintenant possible d'en afficher une par élément de la liste.

Ajout de la propriété ShowItemHint permettant d'activer ou désactiver (par défaut) cette fonctionnalité.

Ajout de l'événement OnGetItemHint pour fournir la valeur à afficher lorsque le composant la demande.

Les listes de contrôles

Le bouton de listes de contrôles (TControlListButton) peut maintenant fonctionner avec une sous liste de choix.

Ajout de clbkSplitPushButton et clbkSplitToolButton à son style pour afficher la flèche d'ouverture de la liste.

Ajout de la propriété DropDownMenu pour gérer la liste et son affichage.

Vous pouvez voir ce nouveau look de bouton en action sur le catalogue de GetIt.

Les listes de contrôles

Si vous ne connaissez pas le composant `TControlList` je vous recommande vivement de vous y intéresser.

C'est une vue sur un ensemble de données qui ne charge que les informations et composants nécessaires à son affichage le rendant très rapide sur n'importe quelle volume de données.

Contrairement à une `TListBox` ou une `TListView` l'affichage de plusieurs millions d'enregistrements est instantané et ne coûte pas plus en espace mémoire que l'affichage de 10 éléments (en dehors bien entendu de la gestion de l'ensemble de données lui-même).

Les listes de contrôles

Serge Girard a montré de nombreuses utilisations de ce composant lors d'une présentation en ligne en mai 2021.

Elle est visible sur

<https://apprendre-delphi.fr/plongee-dans-le-composant-tcontrollist-de-la-vcl-par-serge-girard.html> et

https://www.youtube.com/watch?v=jCd3PDsbN_E&list=PLHLdMyq6m8_sTwOTGwziNlcn443POnmDd&index=9

Vous y trouverez les liens vers le blog de Serge et le dépôt de code des exemples liés à cette présentation.

Liste des fenêtres ouvertes sous forme d'onglets

La remise à niveau du fonctionnement des projets MDI a été l'occasion pour Embarcadero d'introduire de nouveaux composants. Parmi eux le TFormTabsBar associé à la propriété VisualManager sur TForm.

Ce composant affiche la liste des fenêtres ouvertes sous forme d'onglets. Il fonctionne en MDI mais aussi en mode normal.

Ajout des propriétés Transparent et ShowBottomLine pour modifier son affichage.

Ajout de l'événement OnGetTabCaption appelé pour renseigner le nom à indiquer pour chaque onglet.

Liste des fenêtres ouvertes sous forme d'onglets

Ajout de la propriété `MouseMiddleButtonClose` permettant la fermeture d'un onglet (et sa fiche) d'un clic sur le bouton du milieu des souris équipées.

On peut connaître l'onglet et son index à partir de coordonnées relatives dans le composant à l'aide des méthodes `TabFromPoint` et `TabIndexFromPoint` devenues publiques.

Les menus d'actions

Deux composants peuvent afficher le menu principal d'une fenêtre en VCL : TMainMenu et TActionMainMenuBar.

Le premier se gère manuellement dans l'IDE ou par code.

Le second se base sur les actions référencées dans un TActionManager.

Jusque là seul TMainMenu prenait en charge son débordement des écrans.

Désormais le composant TActionMainMenuBar s'adapte aussi à la hauteur maximale de sa zone d'affichage. Un scrolling vertical se déclenche si nécessaire.

Les menus d'actions

L'événement `OnSelectMenuItem` a été ajouté à `TActionMainMenuBar` pour nous permettre de manipuler l'élément de menu lorsqu'il est sélectionné par l'utilisateur.

C'est la (seule) méthode propre pour accéder aux options de menu générées à partir des informations contenues dans les actions.

A propos des actions et listes d'actions

Si vous n'utilisez pas les actions dans vos projets VCL ou FireMonkey, peut-être que cette présentation de Ray Konopka lors de la conférence en ligne Delphicon 2023 vous convaincra d'essayer.

<https://www.youtube.com/watch?v=RX7zaoBqnfA>

Styles VCL

Le look du composant `TToggleSwitch` a été modifié pour qu'il ressemble plus à celui de Microsoft.

La propriété `FormBorderSize` a été ajoutée au `TStyleManager` de `VCL.Themes` afin de choisir si on veut changer la taille de la bordure des fenêtres depuis le style actif ou prendre celle de Windows.

Styles VCL

Les styles VCL Windows ont été mis à jour.

Si vous les utilisez dans vos projets sans passer par les options du projets pour les embarquer automatiquement en ressources de l'exécutable, assurez vous de bien utiliser les bons^(*).

^(*) Les styles installés dans le dossier de la dernière version de RAD Studio, Delphi ou C++Builder.

Faire de jolis(?) dessins

De nouvelles surcharges ont été ajoutées aux méthodes `AngleArc`, `Draw`, `LineTo`, `MoveTo`, `FloodFill` et `TextOut` du `TCanvas` pour utiliser des `TPoint` comme coordonnées.

La librairie `Skia4Delphi` a été mise à jour en version 7.

Vous pouvez l'activer sur un projet afin de bénéficier de nombreux formats d'images comme les SVG ou des animations, de l'affichage de textes avec mise en forme, d'un canevas géré par la librairie `Skia`, ...

Des boutons classés par catégories

Le composant TCategoryButtons regroupe des boutons par catégories, repliables, comme le fait l'IDE dans un concepteur de fiches avec la palette de composants.

Ajout de la propriété CanCollapse sur le composant TButtonCategory.

Ajout de la propriété Enable sur le composant TButtonItem.

Etendre l'IDE pour les projets VCL

RAD Studio permet la création d'extensions à travers l'Open Tools API (OTAPI).

A partir de la version 13 Florence nous avons accès à la résolution de la fiche active à utiliser dans nos éditeurs grâce à la fonction `GetFormDPIForVCLEditors` de l'unité `VCLEditors`.

Une nouvelle surcharge de la procédure `RegisterFieldControl` de l'unité `DrpCtrls` permet de personnaliser le type de champ généré lorsqu'on le glisse depuis la liste des champs d'un ensemble de données FireDAC vers le concepteur de fiches VCL.

Des composants IA en libre service

A partir de RAD Studio 13 Florence Embarcadero propose un ensemble de composants et bibliothèques pour utiliser des moteurs d'IA dans nos projets.

Il fonctionne comme FireDAC avec un système de driver par moteur d'IA et un composant commun de connexion qui donne ensuite accès aux fonctionnalités communes sans se préoccuper de l'API réellement utilisée.

Cet ensemble de composants est téléchargeable depuis GetIt sous le nom de SmartCore AI Component Pack.

Autres mises à jour

La propriété `DropAlign` de la classe `TDragDockObject` est maintenant accessible en lecture et en écriture.

Ajout d'un événement `Onchange` sur le composant `TMonthCalendar`.

Ajout des événements `OnBeforeResize` et `OnAfterResize` sur le composant `TSplitter`.

Ajout d'une propriété `EllipsisPosition` au composant `TDBText` pour indiquer sous quelle forme et où ajouter des points de suspension indiquant que le text est trop long pour être affiché.

Encore ! Encore ! Encore !

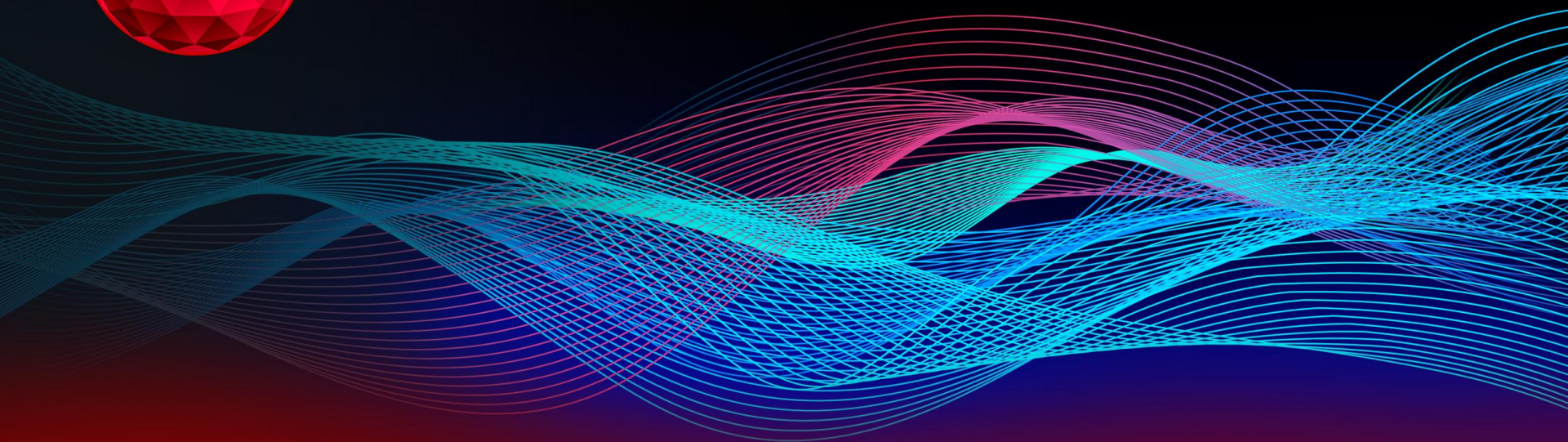
Voilà pour la liste exhaustive des modifications officialisées par Embarcadero pour la VCL dans ses annonces liées à RAD Studio 13 Florence.

En réalité il y en a plein d'autres, pas forcément visibles, dont tous les projets VCL bénéficieront par simple recompilation.

RAD Studio 13 Florence ne s'arrête pas à ces changements. Il y en a de tous les côtés. C'est la raison de cette série de vidéos vous présentant les nouveautés de l'IDE, de Delphi, de C++Builder, de la RTL, de la VCL, de FireMonkey, autour du web et de l'IA.



RAD Studio 13



A propos de RAD Studio 13 Florence

Le site d'Embarcadero pour plus d'infos et télécharger une version d'évaluation :
<https://www.embarcadero.com>

La conférence de lancement du 10 septembre 2025 :
<https://www.youtube.com/watch?v=6Vf-Xo6LpL4>

La documentation en ligne :
https://docwiki.embarcadero.com/RADStudio/Florence/en/What%27s_New

Téléchargez le programme d'installation si vous possédez une licence :
<https://my.embarcadero.com>

A propos de cette présentation

Une vidéo de cette présentation est visible sur la playlist “RAD Studio 13 Florence” de

- la chaîne Barnsten France sur YouTube

<https://www.youtube.com/@BarnstenFrance>

- la chaîne “Quoi de neuf” sur PeerTube

<https://videos.apprendre-delphi.fr/c/quoideneuf/video-playlists>

Les PDF, liens et notes de versions sont en téléchargement depuis

<https://github.com/DeveloppeurPascal/Quoi-de-neuf-dans-RAD-Studio-Delphi-Et-CPPBuilder>

Se former à Delphi 13 Florence ?

Pour vos besoins de formation sur Delphi, le langage Object Pascal, la VCL ou FireMonkey vous pouvez contacter Patrick Prémartin depuis <https://olfsoftware.fr>

Il vous proposera un programme à la carte ou vous redirigera vers des ressources en ligne selon vos besoins. Il peut également vous initier à la programmation pour Internet ou aux interactions entre vos projets et le web.

Ces formations peuvent être effectuées dans le cadre de Qualiopi et prises en charge par les OPCO si vous êtes en entreprise française.

Vous pouvez aussi contacter l'équipe de Barnsten qui fera au mieux pour vous renseigner.

Se former à C++Builder 13 Florence ?

Pour vos besoins de formation sur C++ et C++Builder vous pouvez contacter Laurent Navarro depuis <https://www.altidev.com>

Il vous proposera un programme à la carte ou vous redirigera vers des ressources en ligne selon vos besoins. Il propose également des formations Delphi, web et bases de données.

Ces formations peuvent être effectuées dans le cadre de Qualiopi et prises en charge par les OPCO si vous êtes en entreprise française.

Vous pouvez aussi contacter l'équipe de Barnsten qui fera au mieux pour vous renseigner.

Obtenir RAD Studio 13 Florence

Pour acheter une licence ou mettre à jour la vôtre contactez l'équipe de Barnsten depuis <https://www.barnsten.com>

Ils répondront également à vos questions commerciales comme techniques ou vous redirigeront vers l'interlocuteur le plus apte à vous renseigner dans votre langue.

Conclusion

RAD Studio 13 Florence est une version majeure qui réserve encore quelques surprises.

Si votre licence le permet installez là sur vos postes de travail et entamez la migration et les tests de vos projets.

Si vous n'êtes pas à jour, testez la version d'évaluation téléchargeable gratuitement depuis le site d'Embarcadero.

Et si vous avez des questions contactez l'équipe de Barnsten.

Rappels

Etre MVP Embarcadero n'est pas un métier. C'est un partenariat entre nous et l'éditeur.

Nous ne sommes pas rémunérés pour ce que nous produisons, en revanche nous ne payons pas nos licences des logiciels de développement.

Les informations concernant le programme MVP d'Embarcadero sont publiques et disponibles sur <https://www.embarcadero.com/embarcadero-mvp-program>

Vous y trouverez aussi un annuaire des MVP pour Delphi et C++Builder.

Rappels

La création de cette série de vidéos, de ces PDF et des projets de démo aura pris environ une semaine (plus d'une quarantaine d'heures et ce n'est pas fini).

Si ce travail vous semble utile, dites-le, partagez-le et soutenez-le financièrement dans la mesure de vos possibilités.

Toute contribution est la bienvenue, sous forme de don, d'achat de licences de mes logiciels et librairies, de commandes de formations personnalisées ou sur Udemy, de demandes de prestations techniques et de conseils.

Si vous ne pouvez pas, un “merci” est aussi le bienvenu.

Rappels

Pour un soutien financier vous pouvez me faire un don par ces sites :

 GitHub Sponsors^(*) : <https://github.com/sponsors/DeveloppeurPascal>

 Ko-fi : https://ko-fi.com/patrick_premartin_fr

 Patreon^(*) : <https://www.patreon.com/patrickpremartin>

 Liberapay : <https://liberapay.com/PatrickPremartin>

 Twitch^(*) : <https://www.twitch.tv/subs/patrickpremartin>

(*) à privilégier s'il vous faut une facture ou un justificatif comptable

Rappels

Je distribue mes logiciels utilitaires en shareware.

Certains sont téléchargeables depuis GetIt, les autres depuis leur site ou les dépôts de code.

Pour acheter une licence d'utilisateur final rendez-vous sur :

<https://store.olfsoftware.fr>

Rappels

Mes projets open sources sont disponibles depuis des dépôts de code (GitHub, Codeberg, Framagit) sous licence AGPL.

Si cette licence ne vous convient pas vous pouvez obtenir une licence développeur individuelle classique en passant commande sur :

<https://store.developpeur-pascal.fr>

Rappels

Si vous préférez des goodies il reste quelques mugs en vente exclusivement sur ce site :

<https://goodies.medianim.com/search?query=delphi>



Rappels

Vous pouvez aussi me contacter pour me proposer une mission de formation (dev web ou Delphi), d'assistance technique ou de développement ponctuel.

Je peux également vous conseiller ou vous aider à utiliser tout ce que je diffuse en ligne ou ailleurs.

<https://olfsoftware.fr/contact/>