

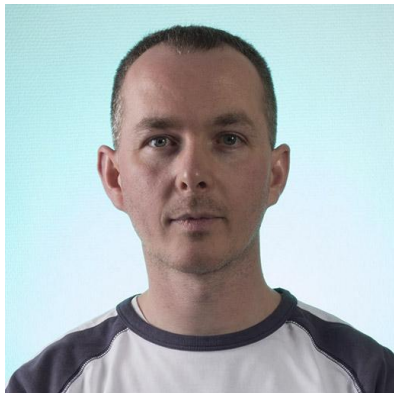


# Delphi<sup>®</sup> 13

## Les nouveautés pour Delphi

mbarcadero<sup>®</sup>

# 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>

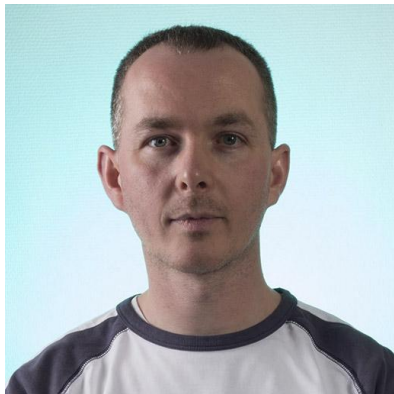
Aussi visible sur Twitch :

<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/>

# “Together” c’est fini...

Les outils de modeling, de refactoring et de formatage de code basées sur “Together” (dépendant de .Net) étaient dépréciées depuis quelques années.

C’était encore installable en option dans RAD Studio 12.x Athens.

Ca n’est plus disponible à partir de RAD Studio 13 Florence.

Parmi les extensions disponibles pour remplacer tout ça il y en a 3 qui sortent du lot: MMX Explorer, GExperts, CnWizard.

Embarcadero suggère de passer par MMX Explorer pour le refactoring.  
Et j’ai opté pour GExperts pour le formatage de code.

# Changements pour CodeInsight (audit de code)

Possibilité d'utiliser l'assistant CodeInsight Classic (celui d'avant Delphi 11 Alexandria) dans Outils / Options / Editeur / Langage / Delphi / Audit de code

En l'activant certaines informations ne seront plus disponibles au niveau de l'éditeur de code et dans les fenêtres autour car l'IDE se base sur le serveur LSP pour les récupérer.

Sur le même écran on peut aussi activer le serveur LSP en 64 bits lorsqu'on utilise l'IDE en 32 bits.

LSP 64 bits est utilisé par défaut par l'IDE en 64 bits.

# Mise à jour des compilateurs

A part pour Windows 32 bits et Android 32 bits les compilateurs sont des programmes en 64 bits.

Les compilateurs en ligne de commande dcc32 et dcc64 sont des programmes 64 bits prenant en charge tout l'espace mémoire nécessaire pour ne plus coincer sur les gros projets que vous voulez compiler en 32 ou en 64 bits.

Bien entendu les compilateurs prennent aussi en charge des évolutions spécifiques et celles qui arrivent au niveau du langage Pascal.

# Nouvelles directives de compilation

Ajout de l'option `$WEAKPACKAGEUNIT` pour éviter des conflits de dépendances entre paquets. Elle est à OFF par défaut pour maintenir le fonctionnement historique.

Elle doit être mise à ON dans les unités que vous voulez embarquer sans risque dans des packages différents.

[https://docwiki.embarcadero.com/RADStudio/Florence/en/Weak\\_packaging](https://docwiki.embarcadero.com/RADStudio/Florence/en/Weak_packaging)

# Nouvelles directives de compilation

L'équivalent existait déjà pour C++Builder. Il est maintenant possible de gérer une pile de configuration des options de compilation dans vos unités.

`{ $PUSHOPT }` empile la configuration actuelle.

`{ $POPOPT }` dépile la configuration stockée lors du `{ $PUSHOPT }` précédent.

Entre les deux vous pouvez paramétrer les options de compilation comme vous le désirez. Ca peut notamment servir dans des unités partagées entre plusieurs projets ou redistribuées et dépendant d'une configuration spécifique.

[https://docwiki.embarcadero.com/RADStudio/Florence/en/Option\\_OPT\\_\(Delphi\)](https://docwiki.embarcadero.com/RADStudio/Florence/en/Option_OPT_(Delphi))



# Nouvelles directives de compilation

On pouvait déjà tester l'état d'une option de configuration "raccourcie". Maintenant on peut aussi tester les options de configuration "longues".

{`$IFOPT`} a été adapté pour simplifier le travail des codeurs et améliorer la lisibilité des codes sources.

[https://docwiki.embarcadero.com/RADStudio/Florence/en/IFOPT\\_directive\\_\(Delphi\)](https://docwiki.embarcadero.com/RADStudio/Florence/en/IFOPT_directive_(Delphi))

# Mise à jour des débogueurs

L'IDE en 64 bits utilise un débogueur mis à jour vers LLDB 20 pour la compilation Delphi vers Windows 64 bits (comme pour C++Builder).

Certaines informations retournées au niveau du débogage ne sont pas disponibles dans l'IDE en 32 bits.

Des améliorations ont été faites sur les fenêtres d'affichage des variables et leurs valeurs notamment pour les énumérations, les propriétés d'objets et les variables de classes.

Des améliorations ont été apportées pour les chaînes stockées en Unicode qui n'étaient pas toujours restituées correctement en évaluations lors d'un debug.

# Mise à jour des SDK, APIs et plateformes

Delphi 13 Florence cible :

- Windows 10 et 11,
- Windows Server 2022 et 2025,
- macOS Sonoma, Sequoia et le tout nouveau Tahoe,
- iOS 18 et la version 26 sortie en septembre pour iPhone, iPad et iPad Pro
- Android 13 à 16 en utilisant la version 35 du SDK
- Linux et Windows WSL2 en console et “démon”,
- Ubuntu et d'autres distributions à interfaces graphiques avec FMX Linux
- des sites et serveurs web avec des outils intégrés, en option ou payants

# Mise à jour des SDK, APIs et plateformes

Les APIs de WinRT ont été mises à jour. Ca représente plus d'une cinquantaine d'interfaces accessibles depuis les unités System.Win.WinRT, WinRT.Utls et d'autres.

Mise à jour des APIs liées au “Projet Reunion” de Microsoft, à savoir les SDK de Windows en 32 et 64 bits liées à Windows 7, 10 et 11.

Mise à jour de l'intégralité des entêtes d'API de Windows (WinMD) représentant plus de 300 fichiers et 40 Mb de code pour Delphi à télécharger depuis GetIt si vous en avez besoin.

Un gros nettoyage a également été effectué pour retirer des déclarations, dépendances et fichiers obsolètes comme ce qui concernait Windows 98.

# Mise à jour des SDK, APIs et plateformes

J'en parle plus en détail dans la présentation concernant FireMonkey en RAD Studio 13 Florence.

Les SDKs de macOS, iOS et Android ont mis à jour.

Les unités de mappages ont été adaptés. Ca entraîne quelques modifications (mineures) de paramètres qui pourraient avoir un impact sur certains projets existants.

De nouveaux composants ont aussi été modifiés et ajoutés en conséquence.

Des unités ont été retirées car elles étaient basées sur des bibliothèques obsolètes ou retirées des OS.

# Evolutions du langage Pascal dans Delphi 13 Florence

Dans Delphi 13 Florence nous retrouvons une nouvelle série de nouveautés au niveau du langage Pascal qui vont rompre la compatibilité descendante si vous les utilisez.

Ne vous en servez que si tous les intervenants sur vos projets sont sur la version 13 Florence (ou une version ultérieure).

Si vous partagez des codes sources avec des développeurs dans de vieilles versions ou en Community Edition<sup>(\*)</sup>, ne jouez pas avec tout ça... sinon, faites vous plaisir !

(\*) au moment de l'écriture de ce doc Delphi Community Edition était basée sur Delphi 12.1 Athens

# Evolutions du langage Pascal dans Delphi 13 Florence

Embarcadero avait déclenché des tremblements dans la force en proposant les variables en ligne et l'inférence de types dans Delphi 11 Alexandria.

La possibilité de déclarer des variables n'importe où dans le code et parfois de ne pas spécifier leur type était assez perturbant pour les pascaliens puristes.

Attendez-vous à de nouveaux remous avec cette version...

# L'opérateur ternaire débarque !

Vous utilisez parfois les fonctions `IfThen()`<sup>(\*)</sup> dans vos projets ? Et bien cessez maintenant !

Nous avons désormais un VRAI opérateur ternaire en Pascal dans Delphi.

```
b := if (2=5) then true else false;
```

```
i := if (b) then 42 else 33;
```

C'était probablement la plus ancienne et plus demandée des fonctionnalités à ajouter au Pascal. Quelle sera la prochaine ?

(\*) <https://developpeur-pascal.fr/demander-l-ajout-d-un-operateur-ternaire-au-pascal-dans-delphi-ou-utiliser-la-fonction-ifthen.html>



# Etre ou ne pas être ?

On peut savoir si une variable descend d'une classe par l'opérateur de condition "is".

Il est maintenant possible de tester si une variable ne descend pas d'une classe par l'opérateur "is not".

```
if not (Obj is TMaClasse) then DoSomething;
```

```
if (Obj is not TMaClasse) then DoSomething;
```

Ca ne changera pas la face du monde mais ça devrait éviter de se tromper avec des AND et OR selon quelles conditions on leur ajoute... 😊 👁️

# Tu l'as ou tu l'as pas ?

Dans la foulée nous avons aussi un changement sur les ensembles.

A “in” s’ajoute “not in” pour savoir si un élément est absent d’un ensemble.

```
if not (Couleur in [bleu, blanc, rouge]) then b := 5 else b := 1;
```

```
if Couleur not in [bleu, blanc, rouge] then b := 5 else b := 1;
```

# Je s'appelle Groot.

Une nouvelle fonction intrinsèque permet de récupérer sous forme de chaîne de caractère le nom de son paramètre.

```
s := nameof(i) + nameof(s);
```

Ca devrait simplifier certaines utilisations de RTTI.

# Free speech mais pas trop !

Nous avons déjà des directives de compilation pour masquer des informations et avertissements du compilateur. Le problème c'est que `{ $HINTS ON }` et `{ $HINTS OFF }` bloque aussi des choses potentiellement utiles.

On préfère parfois travailler sans rien cacher sous le tapis. Dans ce cas seuls les messages d'infos qui ne sont pas anormaux doivent être masqués.

L'un d'entre eux est le W1035 : “résultat de fonction indéfini”. Il peut arriver lorsqu'on sort d'une fonction par une exception sans avoir rempli `Result`.

Dans ce cas on peut maintenant utiliser la directive “`return`” au niveau de sa déclaration ou des procédures qui l'utilisent.

# Nouvelles contraintes pour types génériques

On peut définir des contraintes sur les types génériques comme fournir une classe, une interface, spécifier qu'on doit avoir un constructeur, ...

A ces contraintes s'en ajoutent deux de plus : “interface” et “unmanaged”.

Ca va permettre de faire du transtypage sur le type générique utilisé et de prendre en charge d'autres types d'utilisations à voir prochainement.

# Initialize, Finalize, Assign sans Self

Le paramètre Self était obligatoire depuis l'ajout des opérateurs Initialize, Finalize et Assign sur les RECORD.

Ce paramètre devient implicite et disparaît sur une nouvelle surcharge de ces opérateurs.

[https://docwiki.embarcadero.com/RADStudio/Florence/en/Custom\\_Managed\\_Records#Implicit\\_Self\\_parameter\\_declaration](https://docwiki.embarcadero.com/RADStudio/Florence/en/Custom_Managed_Records#Implicit_Self_parameter_declaration)

# SmartCore AI Component Pack

L'ensemble de composants SmartCode AI Component Pack est maintenant disponible sur GetIt.

Voulu aussi modulaire que FireDAC, il fournit un ensemble de composants “drivers” et utilisateurs pour des LLMs et IA génératives. Le tout est bien entendu extensible.

L'IDE s'en sert notamment pour Smart CodeInsight.

<https://blogs.embarcadero.com/introducing-the-smartcore-ai-components-pack/>

[https://docwiki.embarcadero.com/RADStudio/Florence/en/SmartCore\\_AI\\_Component\\_Pack](https://docwiki.embarcadero.com/RADStudio/Florence/en/SmartCore_AI_Component_Pack)

# Delphi Upgrade Advisor Wizard

Un nouvel outil de diagnostic et d'optimisation de projets est disponible sur GetIt.

Il sert à améliorer le fonctionnement et la compilation de “vieux” projets en vérifiant les chemins d'accès utilisés, les unités renommées entre deux versions de Delphi, les unités dont le préfixe n'est pas utilisé dans les Uses, les références circulaires et corrige les USES par des scripts utilisant reFind.

<https://blogs.embarcadero.com/introducing-the-delphi-upgrade-advisor-wizard/>

[https://docwiki.embarcadero.com/RADStudio/Florence/en/Delphi\\_Upgrade\\_Advisor\\_Wizard](https://docwiki.embarcadero.com/RADStudio/Florence/en/Delphi_Upgrade_Advisor_Wizard)



# C'est tout... pour le moment !

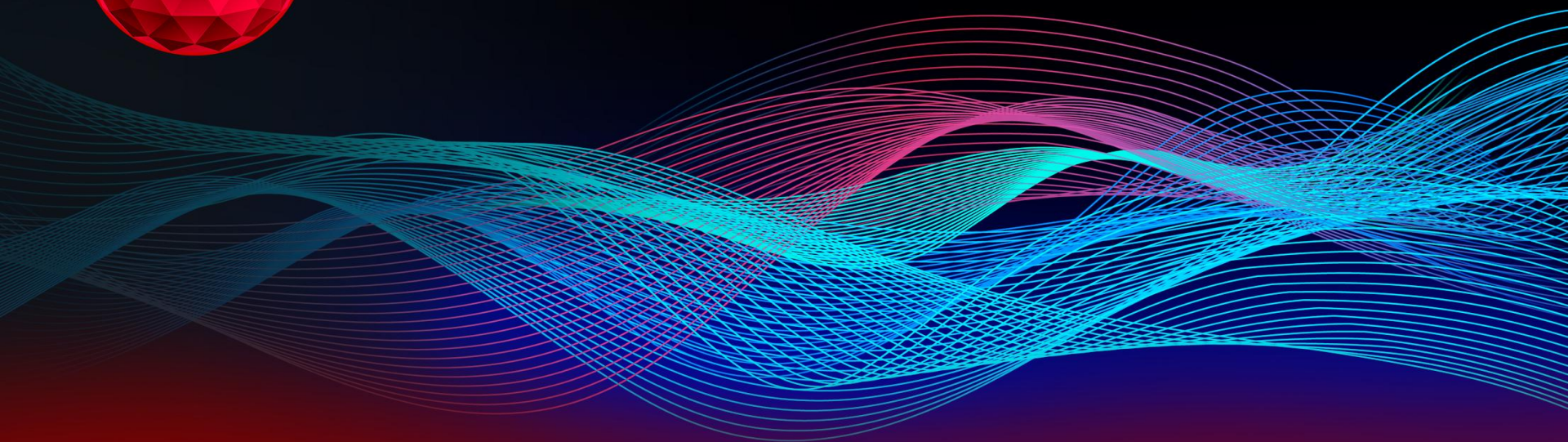
Voici pour le tour des changements liés au langage Pascal et à Delphi.

Embarcadero continue les mises à niveau pour suivre les évolutions des systèmes d'exploitation ciblés tout en poursuivant la migration progressive de tous ses outils vers le 64 bits.

Il reste beaucoup de choses à dire sur les nouveautés dans l'IDE, C++Builder, la RTL, la VCL et FireMonkey. Je vous parlerai aussi web et IA dans les autres présentations de cette série.



# 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](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<sup>(\*)</sup> : <https://github.com/sponsors/DeveloppeurPascal>

 Ko-fi : [https://ko-fi.com/patrick\\_premartin\\_fr](https://ko-fi.com/patrick_premartin_fr)

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

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

 Twitch<sup>(\*)</sup> : <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/>