

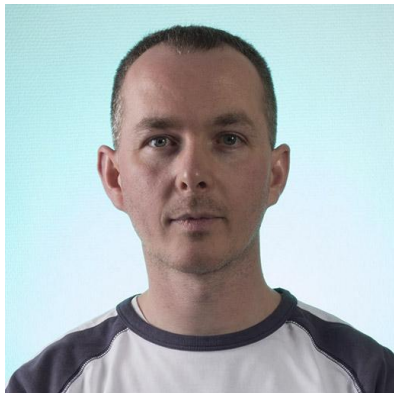


RAD Studio 13

Nouveautés de la RTL



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>

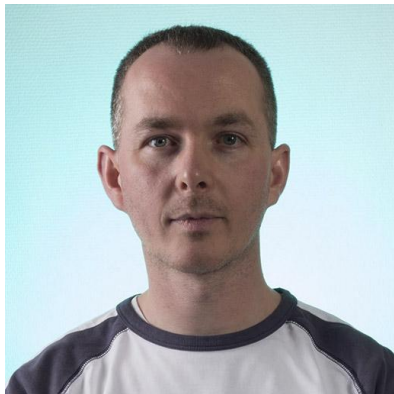
Aussi visible sur Twitch :

<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/>

Parlons RTL au sens large, le socle commun

Dans cette présentation je vais vous parler de ce qui est commun à Delphi et C++Builder mais aussi commun aux projets console, VCL (pour Windows), FireMonkey (pour Windows, Mac, iOS, Android et Linux), services ou serveurs.

Ca englobe la “vraie” RTL (les unités System.*) mais aussi FireDAC, JSON, REST et d’autres choses qui ne méritaient pas forcément une présentation dédiée même si elles ont un impact sur le reste.

Mise à jour des SDKs pour Windows

Embarcadero a mis à jour les SDKs de Windows :

- l'API de WinRT est disponible en totalité
- les entêtes des SDKs pour Windows 7, 10 et 11 sont directement utilisables ou disponibles sur GetIt dans le pack WinMD qui contient plus de 300 unités liées à des APIs de Microsoft
- les dépendances et APIs en lien avec Windows 98 ont été retirées ce qui devrait être transparent pour vos projets

Mise à jour des SDKs pour Windows

Embarcadero a mis à jour les SDKs de Windows :

- les API de Edge WebView2 ont été mises à niveau, le SDK est toujours installable sur demande depuis GetIt.
Ca impacte TWebBroser et TEdgeBrowser en VCL comme en FireMonkey.

Mise à jour des SDKs pour iOS, macOS et Android

Comme c'est le cas pour Windows, Embarcadero a fait des mises à jour pour iOS, macOS et Android.

Côté macOS et Android (SDK 35 depuis RAD Studio 12.3 Athens) c'est transparent.

Côté iOS les entêtes des APIs ont été mis à jour ce qui entraîne des changements dans les signatures de procédures ou fonctions. Si vous ne les utilisez pas en direct pas de changement pour vous, dans le cas contraire attendez-vous à des erreurs de compilation à corriger dans vos projets.

Saisie des nombres entiers

Des procédures et fonctions ont été ajoutées à l'unité System.StrUtils pour prendre en charge les différents modes de saisie des nombres entiers autorisés dans les codes sources Pascal de Delphi:

- saisie décimale classique : ('+' | '-')? [0-9]* }
- saisie hexadécimale : ('\$' | 'X' | 'x' | '0X' | '0x') [0-9A-Fa-f]*
- saisie binaire : '%' [0-1]*

Notez que le souligné est autorisé dans les trois notations.

Saisie des nombres entiers

Les types `TIntegerProperty` et `TInt64Property` ont été modifiés pour en bénéficier. On peut saisir des nombres entiers comme on veut dans l'inspecteur de propriétés du concepteur de fiches.

Dans vos projets vous pouvez remplacer `TryStrToInt()` et `StrToInt()` par `TryLiteralToInt()` et `LiteralToInt()` si vous désirez laisser cette possibilité à vos utilisateurs.

Collections génériques : TThreadedQueue<T>

La méthode Count a été ajoutée à la collection TThreadedQueue<T> de System.Generics.Collections.

Plus besoin de bidouiller la file d'attente pour en obtenir le nombre d'éléments. Si vous utilisez cette classe dans vos projets faites le changement.

Collections génériques : TOrderedDictionary<K,V>

Les classes TOrderedDictionary<K,V> et TObjectOrderedDictionary<K,V> ont été ajoutées en version 12 Athens.

Il fallait appeler la méthode de classe NewInstance pour les instancier. Ca a pu amener à des blocages selon les utilisations. Ces deux classes ont maintenant leur constructeur.

Vous DEVEZ remplacer NewInstance par l'appel de Create() si vous utilisez ces dictionnaires de données triées.

Etant des descendants de TDictionary<K,V> un transtypage est possible sans perte de fonctionnalité.

Synchronisations et verrouillages de threads

Les constructeurs des classes `TMutex`, `TSemaphore` et `TEvent` de l'unité `System.SyncObjs` s'occupent de l'initialisation de leur propriété `LastError` avant de retourner à leur tour une éventuelle erreur.

Si vous gérez des threads et utilisez ces méthodes de synchronisation, de verrouillage ou de signalement, vérifiez que vos projets tournent correctement (en particulier vos “workers”).

Synchronisations et verrouillages de threads

Pour éviter des cas de verrouillages mortels (aka “deadlocks”) l’interface `IReadWriteSyncState` a été ajoutée à l’unité `System.SysUtils`. La classe `TMultiReadExclusiveWriteSynchronizer` prenant en charge des accès multiples en lecture pour un accès en écriture l’implémente.

Ainsi on y retrouve un `WriterID` correspondant à l’identifiant du thread faisant l’écriture lorsqu’il y en a une.

Ca ne devrait pas avoir d’impact sur les projets utilisant ou pas les threads en dehors des opérations de débogage depuis l’IDE.

Changement liés à RTTI

Un changement d'ancêtre a été fait pour TRttiDataMember et TRttiIndexedProperty. Elles descendent maintenant de TRttiPlaceMember qui descend à son tour de leur précédent ancêtre commun TRttiMember.

Cette modification donne accès à 3 nouvelles propriétés : DataType, IsReadable et IsWritable sans casser les héritages et dépendances qui pouvaient exister dans nos projets.

Changement liés à RTTI

La nouvelle fonction `GetStorageSize()` permet d'obtenir la taille occupée en mémoire par le type dont on lui passe les infos.

System.JSON évolue aussi

La librairie fournie pour prendre en charge les structures de données codées en JSON a aussi ses modifications dans cette nouvelle version.

Certains changements sont liés à l'utilisation des APIs de moteurs d'IA et aux composants SmartCore AI disponibles depuis GetIt mais ils pourront nous servir aussi dans des projets de manipulation de données en JSON et de consommation d'APIs web classiques.

System.JSON évolue aussi

TJSONNumber qui était un descendant de TJSONString change d'ancêtre pour la classe TJSONValue.

Suite à ce changement, la classe TJSONNumber n'est plus scellée. On peut en créer des descendants.

System.JSON évolue aussi

La méthode `TJSONValue.GetValue<T>` change de fonctionnement lorsqu'on lui passe une valeur par défaut.

Si pour une raison ou une autre la valeur n'est pas trouvée, convertissable au type demandé ou à `Null`, c'est la valeur par défaut qui est retournée.

Si vous traitez des données "Null" dans des données en JSON méfiez vous car ce changement pourrait impacter vos projets.

Utilisez plutôt la version de `GetValue<T>()` ou `TryGetValue<T>()` sans valeur par défaut ou vérifiez si votre élément `IsNull` avant d'en demander la valeur.

System.JSON.Readers évolue aussi

La fonction ReadUInt64 est ajoutée à la classe TJSONReader

System.JSON.Serializers évolue aussi

On peut mettre des valeurs par défaut lors de la lecture de données JSON. Leur sérialisation va nous permettre de les ignorer. Ca permet de n'avoir que les données modifiées par rapport à des valeurs de base fournies.

Ajout de la méthode `CanWrite()` prenant une valeur en paramètre à la classe `TJsonConverter`.

Ajout de la propriété `TJsonSerializer` de type `MemberSerialization` avec la valeur "Field" par défaut à la classe `TJsonMemberSerialization`.

JSON dans les APIs REST

Pour prendre en charge les nouveautés liées à la sérialisation de données JSON les classes associées bougent aussi au niveau des librairies REST.

Les valeurs `JsonIgnoreDefault` et `JsonIgnoreSpecial` ont été ajoutées à `TJsonOption` dans l'unité `REST.Json`.

Ajout de l'énumération `TJsonValueSerialization` dans l'unité `REST.Json.Types`.

Ajout de la propriété `ValueSerialization` à la classe `TMarshalUnmarshalBase` dans l'unité `REST.JsonReflect`.

Sur les bases de données (unité Data.DB)

Sur GetIt vous pouvez trouver la librairie “Rudy’s Big Numbers” pour prendre en charge les “gros” nombres entiers, décimaux et rationnels.

Ce type de nombre commence à apparaître dans les moteurs de bases de données. Embarcadero a donc ajouté un nouveau type de champ (TFieldType.ftLargeUInt) pour prendre en charge les longs entiers non signés.

C’est pris en charge par les drivers de FireDAC pour Oracle, MySQL et dans le composant TFDMemTable.

Sur les bases de données (unité Data.DB)

A part parcourir la liste de tous les composants instanciés à la recherche des TDataSource pointant sur un TDataSet il n'était pas possible de trouver toutes les sources liées à un ensemble de données.

Embarcadero nous donne maintenant accès à cette information avec deux nouvelles propriétés sur la classe TDataSet : TDataSources et TDataSourceCount.

FireDAC n'est pas en reste

En plus de la prise en charge des champs de type entier long non signé, le driver FireDAC pour Oracle va permettre de se connecter avec l'authentification Oauth 2.0 aux serveurs.

Le support pour Oracle passe à sa version 19c (ou 19.13).

Du côté de Firebird, FireDAC prend désormais en charge le parallélisme des opérations pour le composant TFD FBBBackup.

Le driver FireDAC de PostgreSQL améliore sa prise en charge des types tableaux.

FireDAC n'est pas en reste

RecsMax et RecsSkip sont maintenant supportés par TFDMemTable lorsqu'un filtre est activé.

Pour permettre plus de flexibilité et nous ouvrir des portes, plusieurs méthodes ont été virtualisées dans les classes TFDCustomConnection et TFDCustomTransaction.

La valeur sfFreeFormJSON a été ajoutée à l'énumération TFDDStorageFormat pour permettre de traiter des données JSON "libres".

FireDAC n'est pas en reste

Le composant TFDMemTable est maintenant capable de charger un fichier JSON sans en connaître la structure et de sauvegarder le même type de fichiers à l'aide du nouveau type de format de stockage TFDStorageFormat.sfFreeFormJSON qu'il suffit d'utiliser pour les méthodes LoadFromFile() et SaveToFile().

Ne nous attendons pas à des miracles : TFDMemTable reste un composant d'accès à une table. Les données traitées de cette façon seront mieux prises en charge si ce sont des tableaux d'objets.

Client / serveur

Du côté de DataSnap il est maintenant possible de contrôler le certificat utilisé lors d'une connexion en https entre un client et son serveur grâce à la propriété `ValidatePeerCertificate` présente sur les composants `TFDConnection` et `TDSCClientCallbackChannelManager`.

<https://docwiki.embarcadero.com/Libraries/Florence/en/FireDAC.Comp.Client.TFDConnection.ValidatePeerCertificate>

<https://docwiki.embarcadero.com/Libraries/Florence/en/Datasnap.DSCommon.TDSCClientCallbackChannelManager.ValidatePeerCertificate>

Autres changements

Mise à niveau de DUnitX à la version 0.4.0 complétée par les mises à jours jusqu'au 26 juin 2025. Le commit 3ec1192 est le dernier pris en compte.

La fonction TStringHelper.Contains de System.SysUtils propose maintenant une version sans contrôle de la casse.

La fonction TPath.Exists() a été ajoutée dans System.IOUtils pour tester la validité d'un chemin (vers un fichier ou un dossier).

Les méthodes ReadUInt64 et WriteUInt64 ont été ajoutées à la classe TRegistry de System.Win.Registry.

Autres changements

Un nouveau constructeur a été ajouté à la classe TStringList de System.Classes pour nous permettre d'obtenir une instance juste avec ses options (TStringsOptions) et son comportement face à l'ajout d'éléments déjà présents (TDuplicates).

Dans System.Types la fonction de classe Zero() a été ajoutée à TSizeF. Elle retourne un RECORD initialisé avec $cx = cy = 0.0$

Stop ou encore ?

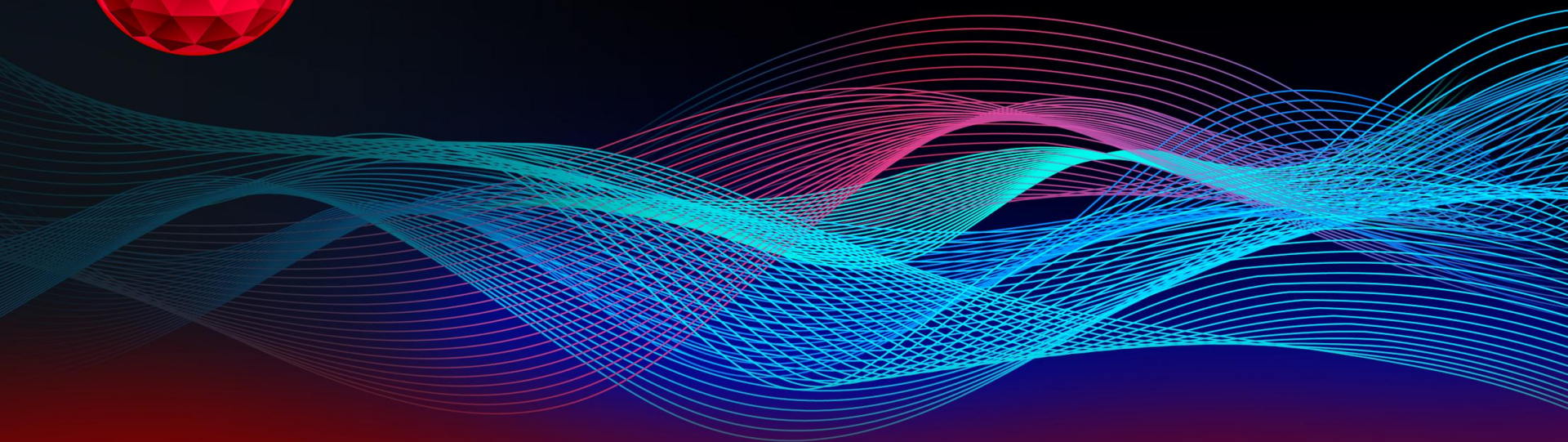
Nous avons fait un tour des modifications communes à tous types de projets et à la RTL pour la version 13 Florence de Delphi et C++Builder.

Maintenant on peut passer aux autres catégories.

On se retrouve pour les autres vidéos de cette série et découvrir les nouveautés et modifications dans l'IDE, la VCL, FireMonkey, Delphi, C++Builder, le web et 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/>