

# Nouveautés de RAD Studio 12 Athens

Nouveautés de la version :

[https://docwiki.embarcadero.com/RADStudio/Athens/en/What%27s\\_New](https://docwiki.embarcadero.com/RADStudio/Athens/en/What%27s_New)

Notes d'installation :

[https://docwiki.embarcadero.com/RADStudio/Athens/en/Installation\\_Notes](https://docwiki.embarcadero.com/RADStudio/Athens/en/Installation_Notes)

Notes de version :

[https://docwiki.embarcadero.com/RADStudio/Athens/en/Release\\_Notes](https://docwiki.embarcadero.com/RADStudio/Athens/en/Release_Notes)

Liste des tickets publics traités sur cette version :

[https://docwiki.embarcadero.com/RADStudio/Athens/en/New\\_features\\_and\\_customer\\_reported\\_issues\\_fixed\\_in\\_RAD\\_Studio\\_12.0](https://docwiki.embarcadero.com/RADStudio/Athens/en/New_features_and_customer_reported_issues_fixed_in_RAD_Studio_12.0)

Plateformes prises en charge :

[https://docwiki.embarcadero.com/PlatformStatus/en/Main\\_Page](https://docwiki.embarcadero.com/PlatformStatus/en/Main_Page)

RAD Studio 12 Questions and Answers! :

<https://blogs.embarcadero.com/rad-studio-12-questions-and-answers/>

Features Matrix :

<https://www.embarcadero.com/products/rad-studio/features/feature-matrix>

## En bref

- nouvel installeur et outil de choix des fonctionnalités installées
- mise à jour des bibliothèques DUnitX et Indy
- prise en charge de plusieurs fichiers à la fois dans GetIt (installation locale)
- InterBase 2020, IBLite/IBToGo 2020 en update 5 avec un quota réhaussé pour l'édition développeur d'InterBase et toujours un déploiement "gratuit" de IBToGo sur les plateformes mobiles dans le cadre des licences Enterprise/Architect.
- AWS SDK for Delphi (dispo sur GetIt) avec de nombreuses API Amazon Web Service directement utilisables dans nos projets. (téléchargement gratuit dans le cadre de la licence Enterprise/Architect)
- 

## Sur l'IDE

- Suppression des dépendances à .Net dans l'IDE pour les fonctionnalités installées par défaut.
- L'outil de refactoring, le reformatage et la gestion des modèles UML sont désormais installés en option (ils dépendent toujours de .Net).

- Lors d'une recherche de mots dans les fichiers il est possible de faire une recherche dans une arborescence avec un filtre sur les sous-dossiers.
- Grosse optimisation de l'utilisation de GDI avec la mise en place des images dormantes dans la VCL.
- Intégration d'assistants liés aux composants Konopka pour la saisie de chaînes de caractères (1 ligne et multi lignes), ajout de configuration rapides au niveau des composants VCL TButton, TRadioGroup et TPanel.
- Des améliorations ont été apportées dans les fenêtres d'aide à la saisie, d'informations sur les erreurs, sur la pile d'appel en débogage, ...  
Le contenu est plus clair et mieux présenté avec notamment l'utilisation de couleurs.
- On pouvait déjà enregistrer la configuration du bureau. On peut maintenant enregistrer la configuration de l'éditeur dans les options de l'IDE.
- Petit remaniement des options de l'IDE avec les options de l'éditeur qui remontent d'un niveau dans l'arborescence pour être accessibles en direct plutôt que depuis les options de l'interface utilisateur.
- En débogage la pile d'appels et la fenêtre d'historique peuvent maintenant être copiées vers le presse papier en utilisant les options de leur menu contextuel.
- La recherche intégrée à l'IDE (depuis la barre de titre et en haut de certaines boîtes de dialogue) a fait l'objet de correctifs et améliorations.
- Le concepteur de fiches également concernant les écrans à haute définition (notamment lorsque le concepteur de fiches est séparé de l'éditeur de code).
- 

## Dans l'éditeur de code (en général)

- La barre de navigation propose une mise en forme de la syntaxe des méthodes listées et suit le thème colorimétrique de l'IDE.
- 

## Dans l'éditeur de code pour Delphi

- CodeInside peut à nouveau se déclencher tout seul avec la réactivation de l'invocation automatique (désactivée par défaut, à activer depuis les options de l'IDE). D'autres améliorations ont été ajoutées sur l'affichage et la pertinence des propositions de complétion.
- Les mots clés du langage sont maintenant présents dans la fenêtre de complétion.
- Le serveur LSP (pour Delphi) se base sur la compilation du projet. En cas d'anomalie dans une unité il peut ne plus proposer les bonnes informations pour la complétion et signaler des erreurs qui n'en sont pas. Si un Ctrl+F9 ne suffit pas à déterminer où est le problème il est aussi possible de relancer le serveur LSP depuis le menu "outils".
- Les couleurs associées aux types TColor et TAlphaColor sont maintenant affichées avec la couleur correspondante et son code hexadécimal dans la fenêtre d'auto complétion.
- Les chaînes de caractères saisies dans l'IDE ne sont plus limitées à 255 caractères.
- Les avertissements liés à des classes génériques affichent des informations plus reproductibles et plus claires selon les cas.

-

## Dans l'éditeur de code pour C++Builder

- Visual Assist de Whole Tomatoes a été intégré à l'IDE. Cet outil permet d'ajouter CodeInsight à l'éditeur de code des projets C++Builder. En plus de l'assistance à la saisie basée sur une analyse du code (et non la compilation comme dans Delphi), de nombreuses fonctionnalités ont été ajoutées.
- Nouvel assistant de création d'applications "WinAPI".
- 

## Dans Delphi

- Les chaînes de caractères peuvent maintenant être gérées sous forme de blocs multilignes. (pas encore pris en charge par le formateur de code)
- Les types NativeInt et NativeUInt sont désormais des alias vers Integer/UInt et Int64/UInt64 plutôt qu'une redéclaration de type. Attention, ça peut avoir des effets de bords entre les compilations 32 et 64 bits suite au passage à la version 12 Athens.
- Conformément aux recommandations de l'IEEE les comparaisons de valeurs dont le résultat est un NaN deviennent sont assimilées à un false.
- Il est possible de générer un graphique d'inclusion des unités afin de détecter et résoudre des références circulaires (ou simplement nettoyer les USES inutiles). C'est le compilateur qui les liste dans un fichier texte au format GraphViz.
- 

## Dans C++Builder

- C++Builder 12 Athens ne pouvant pas générer des binaires utilisables sur les magasins d'applications pour iOS et Android, Embarcadero a retiré ces deux cibles de compilation. Les projets FireMonkey en C++Builder sont maintenant que pour Windows 32 et 64 bits en attendant le retour de compilateurs et bibliothèques à jour. Il faut utiliser la version 11.3 Alexandria pour cibler iOS 16 ou Android en 32 bits.
- Clang 15 est disponible (en preview). Il faut l'activer dans les options.
- 

## Dans la RTL

<https://blogs.embarcadero.com/3-x-12-rtl-data-and-internet-enhancements-in-delphi-12/>

- GetCompilerVersion et GetRTLVersion font leur apparition dans l'unité System pour nous fournir les numéros de version du compilateur et de la bibliothèque à l'exécution plutôt que sous forme de directives de compilation.
- Les exceptions de type "floating point" sont désactivées sur tous les compilateurs par défaut (mais peuvent être réactivées dans les options de compilation) pour suivre les

évolutions du marché des compilateurs (quasiment tout le monde les masquant).

[https://docwiki.embarcadero.com/RADStudio/Athens/en/Floating\\_Point\\_Operation\\_Exception\\_Masks](https://docwiki.embarcadero.com/RADStudio/Athens/en/Floating_Point_Operation_Exception_Masks)

- Les index utilisés sur les collections sont maintenant des NativeInt et non des Integer (impact possible en recompilation 64 bits sur de gros volumes de données traitées si vous n'êtes pas sur le bon type d'entier).
  - Les messages concernant les dépassements de capacité sur les TList ont été revus pour fournir des informations utiles pour traiter l'erreur.
  - Le type TArray permet de tester la présence d'une valeur et d'en trouver la position à l'aide de nouvelles méthodes.
  - TEncoding.UseBOM et TEncoding.IsBufferValid permettent de vérifier si un contenu est compatible avec l'encodage désiré et d'activer ou non le BOM pour le chargement et la sauvegarde de textes en UTF ou ASCII.
  - ToString<T> a été ajouté sur le TArray pour sérialiser le contenu d'un tableau avec les délimiteurs de notre choix.
  - On peut utiliser notre méthode de chiffrement / déchiffrement sur les fichiers ZIP en implémentant l'interface IZipCryptor.
  - Les fichiers de plus de 4GB sont autorisés dans la librairie ZIP en 64 bits.
  - Les événements onThreadStart et onThreadTerminate font leur apparition sur TThread pour ajouter du code d'initialisation et de finalisation d'un processus secondaire.
  - On peut maintenant passer outre le nombre maximum de threads lancés dans le pool de threads en activant UnlimitedWorkerThreadsWhenBlocked (par défaut à True). Ca n'est pris en compte qu'en cas de blocage du pool de threads.
- <https://docwiki.embarcadero.com/Libraries/Athens/en/System.Threading.TThreadPool.UnlimitedWorkerThreadsWhenBlocked>
- TPath.Combine() permet maintenant de lier plus de 2 éléments d'un chemin d'accès et d'éviter d'appeler des TPath.Combine() les uns dans les autres.
  - TPath.GetDesktopPath retourne le chemin vers le bureau.
  - TPath.GetAppPath retourne le chemin vers le dossier contenant l'exécutable en cours.
  - CreateImplementation a été ajoutée sur TRttiInvokableType.
  - Des énumérateurs ont été ajoutée sur TFile (GetLinesEnumerator) et TTextReader.

## Du côté d'Internet et des encodages

- JSON Data Binding Wizard a été ajouté pour générer des unités de déclaration d'objets à partir de fichiers JSON. Il est disponible pour les projets Delphi par ajout d'un nouveau fichier depuis la rubrique "web".
- [https://docwiki.embarcadero.com/RADStudio/Athens/en/The\\_JSON\\_Data\\_Binding\\_Wizard](https://docwiki.embarcadero.com/RADStudio/Athens/en/The_JSON_Data_Binding_Wizard)
- IBLite et IBToGo sont utilisables sur le simulateur iOS. Ca permet de faire des tests et du débogage de projets même pour iOS 17 sans avoir besoin d'iPhone ou d'iPad.
  - Un encodeur a été ajouté pour le format Sqids.
- <https://docwiki.embarcadero.com/Libraries/Athens/en/System.NetEncoding.TSqidsEncoding>

- Des correctifs et améliorations ont été faits sur les classes et composants d'accès à Internet. THttpClient, TNetHttpClient, TRESTClient, TRESTRequest et d'autres sont concernés.
- OnRedirect et OnUpdateCookie ont été ajoutée sur THttpClient et TNetHttpClient pour accéder plus facilement aux cookies dans une requête.
- THttpClient.Delete et THttpClient.BeginDelete ont une nouvelle version avec un stream en paramètre.
- Le type MIME "application/vnd.retailer.v8+json" est reconnu comme du "application/json" par TRESTClient.
- TMultipartFormData prend en charge des fichiers volumineux à l'aide d'un nouveau stream.
- La couche web de la RTL utilise "curl" sur Linux. Pour les autres plateformes ce sont les API du système. si on veut utiliser "curl" partout il suffit d'ajouter l'unité System.Net.HttpClient.Linux
- Sur THttpRequest, THttpClient, TNetClient, TCustomRESTClient on a maintenant un événement OnReceiveDataEx et sur certains ReceiveDataExCallback pour traiter les données arrivées sans attendre la fin de leur réception.  
<https://docwiki.embarcadero.com/Libraries/Athens/en/System.Net.HttpClient.IHttpRequest.OnReceiveDataEx>  
<https://docwiki.embarcadero.com/Libraries/Athens/en/System.Net.HttpClient.THttpRequest.ReceiveDataExCallback>
- 

## RAD Server

- Le gestionnaire de mémoire FastMM5 est utilisé dans la nouvelle version de RAD Server.
- On peut utiliser Sqids pour encoder des éléments d'URL.  
[https://docwiki.embarcadero.com/RADStudio/Athens/en/Tutorial:\\_Using\\_Sqids\\_encoding](https://docwiki.embarcadero.com/RADStudio/Athens/en/Tutorial:_Using_Sqids_encoding)
- Les informations de connexions étaient gérées sous forme de token. On peut aussi y accéder avec les méthodes ExportLogin et ImportLogin de TCustomBackendAuth.
- La console de RAD Server a été ajoutée aux outils de l'IDE.
- OnNeedClientCertificate est ajouté côté client comme OnValidateCertificate pour simplifier l'utilisation de certificats personnalisés
- TBackendAuth.LoggedInValue.ExpiresAt fournit la date d'expiration en heure locale.
- 

## FireDAC

- TFDQBE ajoute la possibilité de faire des "requêtes par l'exemple" dans les ensembles de données. Ça permettrait de changer les requêtes classiques par une saisie simplifiée de critères (et donner la main à certains utilisateurs finaux).
- 3 versions de SQLite sont maintenant embarquées avec FireDAC:
  - la dernière version (3.42) de la librairie (sans chiffrement ni mot de passe)

- la version SQLite EE (à condition d'avoir une licence auprès de [sqlite.org](http://sqlite.org)) avec chiffrement
  - la version historique (SQLite 3.31.1 FDE) avec les mots de passe gérés par embarcadero (pour des raisons de compatibilité avec des projets existants).
- => impact possible sur du code existant, vérifiez les fonctionnalités utilisées dans vos projets et basculez sur le bon driver
- => Embarcadero suggère de passer à IBToGo pour éviter des couacs à l'avenir, sachant que la migration de la version 3.31.1 (avec mot de passe) vers la 3.42 (sans mot de passe) est complexe car les deux drivers ne peuvent pas être chargés simultanément en linkage statique, rendant l'opération infaisable sur mobiles où on n'a pas accès aux fichiers en dehors de l'application d'origine.
- Une propriété SecurityOption a été ajoutée sur TFDConnection afin de filtrer les requêtes avant de les passer au moteur de base de données. Ca peut éviter des piratages ou des modifications de données sur des bases théoriquement en lecture seule. Utilisez-la !  
<https://docwiki.embarcadero.com/Libraries/Athens/en/FireDAC.Stan.Option.TFDSecurityOptions>
  - L'unité FireDAC.Phys.SQLiteWrapper.Stat.pas pointe sur la version à jour de SQLite, il faut utiliser la nouvelle unité FireDAC.Phys.SQLiteWrapper.FDEStat.pas pour fonctionner avec la version SQLite FDE historique.
  - Au niveau de TFDPhysSQLiteDriver le moteur SQLite FDE a été ajouté de façon séparée, il reste actif avec siDefaut mais si vous voulez vous assurer de pointer dessus dans l'avenir mettez plutôt siFDEStatic. Un linkage dynamique reste possible.
  - Le type "Decimal128" est maintenant supporté par les "bases" MongoDB.
  - En passant à SQLite 3.42 les opérations ALTER TABLE complètes seront prises en charge, dont le DROP sur les champs.
  -

## Dans la VCL

<https://blogs.embarcadero.com/3-x-12-vcl-enhancements-in-delphi-12/>

- Ajout d'images "dormantes" pour éviter d'utiliser inutilement GDI. Cf <https://docwiki.embarcadero.com/Libraries/Athens/en/Vcl.Graphics.TBitmap.Dormant>
- Réécriture complète concernant les nouvelles applications MDI. L'assistant de création de projet a été mis à jour pour prendre en charge ces évolutions et les écrans à haute résolution.  
Les projets existants héritent d'une partie de ces mises à jour mais une réécriture partielle est nécessaire pour bénéficier de tout ce qui est proposé.  
[https://docwiki.embarcadero.com/RADStudio/Athens/en/What%27s New#MDI Reworked for HighDPI and Styles](https://docwiki.embarcadero.com/RADStudio/Athens/en/What%27s%20New%20in%20MDI%20Revised%20for%20HighDPI%20and%20Styles)
- Le composant TFormTabsBar a été ajouté pour la prise en charge de fenêtres sous forme d'onglets, dans une application MDI historique mais d'une façon générale toute application VCL.
- Les fontes utilisées dans les composants VCL peuvent prendre en charge la résolution de l'écran.
- WebView2 a été mis à jour pour le composant TEdgeBrowser.

- La TListView supporte un affichage sous forme de tuiles.
- La nouvelle méthode GetControls retourne un énumérateur sur les listes de TControl et TComponent avec la possibilité de filtrer les éléments parcourus.
- La propriété ShowInTaskBar permet d'afficher n'importe quelle fiche d'une application dans la barre de tâches du système.
- Possibilité de prendre en charge le thème sombre "immersif" de Windows.
- Modification autour du double buffer et sa prise en compte dans les contrôles.  
<https://docwiki.embarcadero.com/Libraries/Athens/en/Vcl.Controls.TWinControl.DoubleBufferedMode>  
<https://docwiki.embarcadero.com/Libraries/Athens/en/Vcl.Controls.TWinControl.CanUseDoubleBuffering>
- 

## Dans FireMonkey

<https://blogs.embarcadero.com/3-x-12-firemonkey-and-android-enhancements-in-delphi-12/>

- Sur Android IFMXCameraService, IFMXTakenImageService et IFMXPhotoLibrary sont devenus asynchrones, ne bloquant plus le thread principal. Ca peut avoir des effets de bord notamment en ne bloquant pas les éléments de l'interface utilisateur déclenchant leur appel. Pensez à désactiver les boutons avant appel et les libérer sur la fin des actions.
- La liste des permissions Android de JManifest\_permission\_groupClass et JManifest\_permissionClass a été mise à jour. Les options correspondantes des projets FMX ont également été mises à jour.
- Modification de IFMXExtendedClipboardService pour limiter les risques de voir apparaître du contenu sensible (comme des mots de passe) dans l'aperçu du clipboard ajouté à Android 13.
- Prise en charge des nouveaux paramètres pour le splash screen des applications Android apparu en Android 12.  
<https://developer.android.com/develop/ui/views/launch/splash-screen/migrate>
- Mise à jour de la librairie Google Play Billing en version 6 conformément aux recommandations (impératives) de Google pour diffuser une application par le Play Store.
- Les applications pour iOS et Android supportent maintenant le mode "split screen" utilisé sur certains smartphones et les tablettes. Des changements au niveau du manifeste pour Android sont nécessaires pour les applications existantes. TWinFormPositionerService a été ajouté sur toutes les plateformes pour simplifier les traitements sur la position des fenêtres.
- TEdit et TMemo font l'objet d'une refonte sur leurs styles et fonctionnement entamée il y a quelques versions et poursuivie à destination de Mac et Windows. Les versions iOS et Android suivront.
- TEdit et TMemo prennent en charge l'opération "undo".
- Les propriétés AutoSelect et CharCase font leur apparition sur le TEdit.
- La saisie IME est disponible pour TEdit et TMemo nativement.

- Des changements ont été faits sur le clavier virtuel pour iOS (notamment concernant la barre de boutons personnalisables qui se trouve au-dessus) afin d'en faire le moins possible au niveau de FMX et le plus possible au niveau de l'OS.
- Des changements sont intervenus sur les champs de saisie pour Android (notamment le curseur et la sélection). Les styles de cette plateforme ont été modifiés.  
Si vous utilisez un style FMX standard, mettez le à jour. Si vous avez créé le vôtre, voyez quelles modifications sont nécessaires pour l'adapter.
- Les mouvements du curseur au doigt sont maintenant possibles sur les champs de saisie sur Android.
- Un outil de génération d'icônes a été ajouté aux options de projet pour créer automatiquement une icône pour les projets à partir d'une image classique ou vectorielle. Cet outil prend en charge les "adaptive icons" d'android et les splash screens.
- Le nouveau TFontManager (basé sur un service de plateforme) permet l'utilisation de fontes non référencées par l'OS dans notre application sous Windows et Mac.
- Nous avons désormais plus d'informations au niveau des opérations faites au doigt dans la classe TTouch (identifiant du mouvement, position, type d'opération enregistrée).
- En cas de double click les événements onMouseDown et onMouseUp sont appelés pour le premier clic mais aussi pour le second avec ssDouble dans le paramètre Shift.
- ErrorRadius de TLocationSensor est disponible pour iOS et Android (il l'était déjà sur Windows et Mac).
- Un nouveau style pour Windows 11 a été ajouté dans FireMonkey. Il est actif par défaut sur les projets si vous n'en avez pas importé un autre.
- Il est possible de désactiver le blocage global du canvas avec TCanvasStyle.DisableGlobalLock (utile pour les projets utilisant plusieurs processus pour dessiner, comme le fait Skia, mais risqué si vous ne savez pas ce que vous faites).
- Les colonnes (TColumn) des grilles (TGrid) ont un HeaderSettings et un HeaderStyleLookup pour les relooker plus facilement avec un style FMX.
- 

## Prise en charge des plateformes

- les SDK pour Android (API level 33), iOS (SDK 17) et Mac (Sonoma) ont été mis à niveau
- Xcode 15 est supporté
- le débogage sous iOS 17 est supporté pour le simulateur (sur Mac M1 et suivants) mais ne peut être exécuté sur les appareils (suite à des changements chez Apple).
- LLDB a été mis à jour sur Mac et Linux
- Les messages dans PAServer sont maintenant transmis aussi dans la fenêtre de messages de l'IDE.
- En Delphi le débogage de code anonyme affiche maintenant aussi les variables capturées (venant de l'extérieur du code anonyme).

- La librairie Ski4Delphi est intégrée par défaut (mais vous pouvez toujours utiliser le dépôt public si vous préférez) pour Delphi et C++Builder. Ce n'est pas activé par défaut sur les projets, à nous de l'activer si on veut bénéficier de ses fonctionnalités et composants en VCL comme en FireMonkey.  
<https://docwiki.embarcadero.com/RADStudio/Athens/en/Skia4Delphi>
- La version de Skia4Delphi proposée par Embarcadero embarque quelques fonctionnalités non disponibles sur la version open source : prise en charge de Vulkan (pour Android et Windows), des shaders utilisés pour les effets, l'impression en PDF depuis l'unité FMX.Printer, les animations sous formes d'images WebP. Ceci est valable pour Delphi et C++Builder.
- Embarcadero propose les unités correspondant aux entêtes d'API Windows utilisés dans les librairies. Une version tenant compte de l'intégralité des API Windows est aussi disponible sur GetIt depuis le dépôt WinMD de Microsoft.
-