

## RAD Studio 10 Seattle 機能評価ガイド

2015 年 9 月

エンバカデロ・テクノロジーズ



# 目次

|                                             |    |
|---------------------------------------------|----|
| 概要 .....                                    | 1  |
| お問い合わせ先 .....                               | 3  |
| 主な機能と開発者にとってのメリット .....                     | 4  |
| 1. 新しい Win32 向け C++ Clang ベースのコンパイラ .....   | 4  |
| 2. IDE の大幅な性能改善を実現 .....                    | 6  |
| オブジェクトインスペクタ、構造ビュー、非ビジュアルコンポーネントの非表示 .....  | 6  |
| マルチモニタサポートの拡張 .....                         | 7  |
| IDE の生産性強化 .....                            | 8  |
| 統合ツールの改善 .....                              | 9  |
| IDE 内で開発環境を拡張可能 .....                       | 10 |
| FireUI マルチデバイスデザイナー .....                   | 11 |
| 3. 強力な Windows 10 サポート機能 .....              | 13 |
| WinRT/UWP の統合 .....                         | 13 |
| 新しい Windows 10 VCL コントロール .....             | 15 |
| 高解像度モニタのサポートとマルチモニタの改善 .....                | 18 |
| VCL スタイルの改善 .....                           | 19 |
| Windows 向けの FireMonkey の強化 .....            | 20 |
| 4. モバイルおよび IoT サポートの拡張 .....                | 22 |
| Android サービスの利用 .....                       | 22 |
| 10 Seattle での iOS サポート .....                | 23 |
| IoT デバイスを活用したソフトウェアの開発 .....                | 25 |
| 5. データアクセス機能の強化 .....                       | 27 |
| FireDAC とは .....                            | 27 |
| FireDAC の MongoDB サポート .....                | 28 |
| 他の FireDAC の改善 .....                        | 29 |
| DataSnap の改善 .....                          | 30 |
| クラウドの拡張 .....                               | 30 |
| 6. RTL ライブラリのコア拡張 .....                     | 31 |
| XE8 で追加された新しい HTTP(S) クライアントライブラリ .....     | 31 |
| 10 Seattle で追加された JSON / BSON ストリーミング ..... | 32 |
| 7. 追加のコンポーネント／ツール .....                     | 33 |
| AppAnalytics .....                          | 33 |
| EMS (Enterprise Mobility Services) .....    | 35 |
| BeaconFence .....                           | 37 |
| Konopka's Signature VCL Controls .....      | 38 |
| CodeSite Studio .....                       | 38 |

RAD STUDIO 10 Seattle 追加情報 .....40

10 Seattle 製品ラインナップ .....40

    RAD Studio 10 Seattle Professional.....41

    FireDAC Client/Server Add-On Pack for RAD Studio 10 Seattle Professional .....41

    RAD Studio 10 Seattle Enterprise .....41

    RAD Studio 10 Seattle Ultimate.....41

    RAD Studio 10 Seattle Architect.....41

## 概要

このたびは、RAD Studio 10 Seattle を評価いただき、誠にありがとうございます。RAD Studio 10 Seattle は、エンバカデロ・テクノロジーズが提供するフル機能のソフトウェア開発ソリューションで、Windows 10、Mac、iOS、Android 向けの真のネイティブアプリケーションの構築に加え、クラウドや IoT とつながるアプリケーションの構築をサポートしています。この評価ガイドでは、RAD Studio 10 Seattle を使用した場合の主なメリットをいくつか解説し、Windows 10 アプリケーション開発とマルチデバイスアプリケーションの構築に利用できる新機能を紹介します。

20 年以上にわたり、RAD Studio のユーザーは、業務アプリケーション構築にフォーカスしてきました。RAD Studio 10 Seattle は、こうした日々の業務を支えるビジネスアプリケーションの構築をサポートすることはもちろんですが、最近のモバイル化や IoT といった新しい要求にも対応できるようにすることにも注力しています。RAD Studio が特定の業界向けのアプリケーションでさまざまな実績を上げていることは、RAD Studio の能力を示すものであるといえます。例えば、小売、財務、製造、産業用オートメーション、科学技術といった分野、そして MRI 装置のような特定のハードウェア向けの制御などにおいても活用されています。

RAD Studio 10 Seattle を用いれば、新しい Windows 10 VCL コントロール、プラットフォームスタイル、ユニバーサル Windows プラットフォームサービスコンポーネントなどを用いて、すばやく、かつ容易に既存の VCL および FireMonkey アプリケーションを Windows 10 に対応させることができます。さらに、プロジェクトで利用できるメモリサイズが約 2 倍になるなど、大幅な統合開発環境（IDE）の機能強化により、開発作業を効率化できます。

RAD Studio 10 Seattle は、32-bit および 64-bit Windows 10 を含む、複数のデスクトップ、モバイル、クラウド、データベースプラットフォーム向けのアプリケーション構築を 5 倍以上の生産性でサポートしています。特に Windows 向けには、新たに Windows 10 に対応した VCL（Visual Component Library）によるコンポーネントベースのネイティブアプリケーション開発が劇的な生産性を実現しています。マイクロソフトが、自社の OS の最新バージョンをリリースしたことを受けて、RAD Studio 10 Seattle では、VCL、Windows プラットフォーム向けの API および SDK の統合、Windows 向けの開発のサポート全般について大幅な機能強化を施しました。さらに、C++においても、RAD Studio 10 Seattle では、サポート強化を行い、C++ Win32 コンパイラについて、新たに LLVM および Clang ベースの実装を行い、C++11 言語標準をサポートしました。

新バージョンは、「Windows 10 サポート」という強力なフォーカスポイントを表すために、「10 Seattle」と命名されました。これは、これまでの「RAD Studio XE」シリーズから、新たに「RAD Studio 10」シリ

ーズがスタートしたことでもあり、それぞれのバージョンについては、今回の「Seattle」のように都市名の愛称が付けられます。

RAD Studio ではさらに、Windows、Android、iOS、Mac OS X 向けのマルチデバイスアプリケーションを、単一のツール、単一のプログラミング言語、単一のフレームワークを用いることで、極めて短時間で構築可能にします。開発したアプリケーションは、実行時ランタイムにはまったく頼らずに、各サポート対象のプラットフォームで高いユーザーエクスペリエンスを提供するネイティブかつ安全なアプリケーションを作成することができます。RAD Studio では、複数 OS の複数の画面サイズの UI を単一のフォームから構築し、OS ネイティブの機能を活かしながらも、単一コードによる開発が可能です。このようなマルチデバイス開発とネイティブコードへのコンパイルの組み合わせは他に類を見ないものです。RAD Studio を用いれば、最高のアプリケーションパフォーマンスを実現するうえで障害となるスクリプト言語や仮想マシンを使用せずに、すばやく開発できるようになります。

RAD Studio 10 Seattle は、「接続性」という点でも強力な機能を提供しています。一般的なクライアント／サーバー型のデータベースサポートから、HTTP/HTTPS ライブラリの統合強化、Bluetooth、Bluetooth LE のサポート、「アップテザリング」と呼ばれるモバイルとデスクトップアプリを容易に連携可能にする技術など、さまざまなサポートがあります。これらの技術を用いることで、開発者は、既存の Windows VCL アプリケーションをモバイルや IoT デバイスと連携する、より高度なアプリケーションへと拡張できます。

そして、多くのアプリケーションは単独で動作するのではなく、他のサービス、例えばクラウドサービス、オンプレミスのミドルウェア、エンタープライズデータベースなどに接続して動作します。RAD Studio 10 Seattle では、EMS（Enterprise Mobility Services）と呼ばれるミドルウェアサービスにより、これを強力にサポートしています。EMS には、モバイル向けのプッシュ通知のサポート（Apple および Google のプラットフォームを利用）、柔軟なユーザー／ログイン管理など、いくつもの新機能が搭載されています。EMS に加え、RAD Studio では、独自に中間層を構築できる SDK を提供する DataSnap、REST API へのアクセス、ネイティブライブラリを利用できる新しい HTTP(S) クライアントフレームワーク、Parse、Kinvey をダイレクトサポートした BaaS コンポーネント、AWS および Azure 向けのクラウド統合など、モバイルおよびデスクトップ向けの強力な接続機能が用意されています。

さらに、VCL および FireMonkey の既存アプリケーションおよび新規アプリケーション向けに、アプリケーションの利用状況を把握できる新機能 AppAnalytics も用意しています。これにより、エンドユーザーがどのようにアプリケーションを使用しているかを理解できるアナリティクス情報を収集できます。

RAD Studio 10 Seattle は、Windows 10、Mac、モバイル、IoT など、多様なアプリケーションの迅速な構築をサポートする開発ツールスイートです。さらに、RAD Studio 10 Seattle を用いれば、Windows 向けの強力なコンポーネントライブラリを用いた開発、マルチプラットフォーム向けの単一コードベースによるネイティブアプリケーション開発を実践でき、パフォーマンスとユーザーエクスペリエンスを犠牲にすることなく、高い開発生産性を維持したアプリケーション構築が可能になります。

RAD Studio 10 Seattle では、以下の 2 つのプログラミング言語をサポートしています。

- 強力な言語機能を利用できる C++ 言語（C++11 標準に対応しアップデート）
- より使いやすくなった最新の Object Pascal 言語（C# などで採用されている新しい言語機能も搭載）

この評価ガイドは RAD Studio 10 Seattle の機能評価の出発点であることにご留意ください。RAD Studio 10 Seattle には、このガイドで紹介しきれないほどの数々の機能があります。エンバカデロでは、これらの機能を最大限に活用するのに役立つ補足的な情報、ビデオ、ウォークスルー、ガイドなどを用意しています。本製品の最新情報については、機能一覧、製品情報ページなどを参照してください。巻末には、その他のリンク一覧を記載しています。

## お問い合わせ先

RAD Studio 10 Seattle の評価に関してご不明な点、ご質問などがございましたら、下記までお問い合わせください。

エンバカデロ・テクノロジーズ インフォメーションサービスセンター

TEL : 03-4577-4520 Email : [japan.info@embarcadero.com](mailto:japan.info@embarcadero.com)

# 主な機能と開発者にとってのメリット

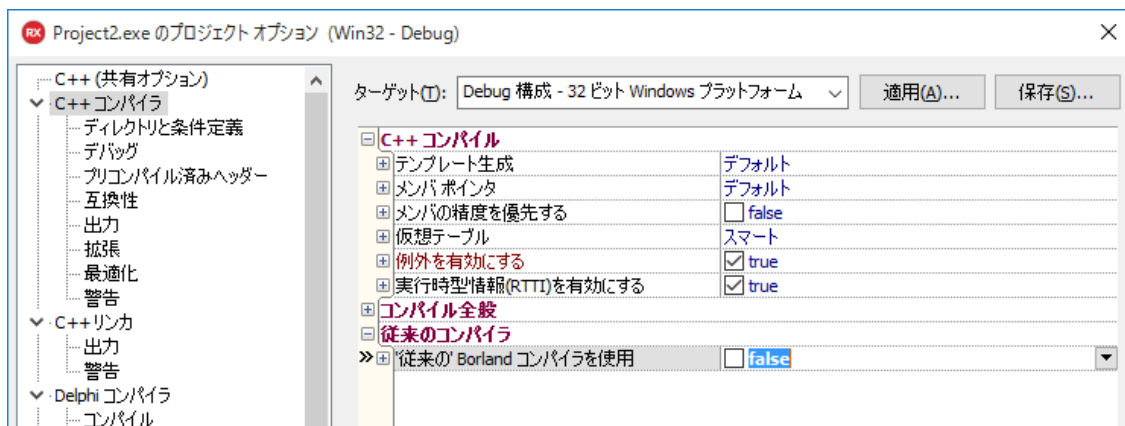
RAD Studio（および C++Builder、Delphi）は開発ツール市場で長きにわたり揺るぎない存在感を示してきましたが、近年のマルチプラットフォームサポートにより、その機能は大幅に拡張されています。現在では、12を超える数のターゲットプラットフォームをサポートするネイティブコンパイラ、広範なクロスプラットフォームランタイムライブラリ、エンタープライズクラスのデータベースアクセスサポート、従来からの VCL Windows UI ライブラリに加えて、マルチデバイスをサポートする FireMonkey プラットフォームが提供されています。

これらの機能をすべて 1 つのドキュメントで詳細に説明しようとする膨大な量になりますし、RAD Studio のオンラインドキュメントには、何千ページものチュートリアルと何万ページもの API ドキュメントが含まれています。したがって、本書では、中核となる機能のいくつかを主に取り上げ、バージョン 10 Seattle で導入された新しい機能にフォーカスすることにします。この機能評価ガイドに加えて、旧バージョン向けの機能評価ガイドも、同じく参考になるでしょう。

## 1. 新しい Win32 向け C++ Clang ベースのコンパイラ

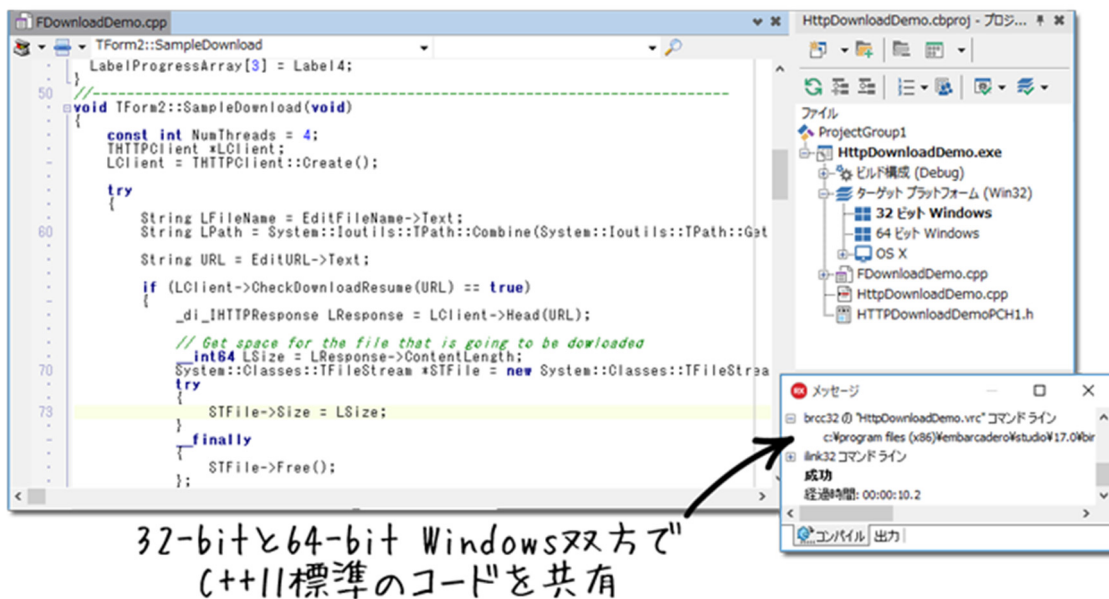
RAD Studio 10 Seattle には、Win32 をターゲットとした新しい C++言語のコンパイラが搭載されています。従来からの bcc32 コンパイラも後方互換性のために引き続き搭載されていますが、新しいコンパイラ bcc32c は、LLVM アーキテクチャをベースとしており、標準の Clang コンパイラをビジュアルコンポーネントベースの開発のために拡張したものです。そして、このコンパイラは、新しい Clang / LLVM 3.3 をベースとしています。

デフォルトでは、既存の C++プロジェクトを移行したり、新しい C++プロジェクトを作成するときに、従来の C++コンパイラが使用されます。必要に応じて、C++コンパイラオプションの設定を変更して、新しいコンパイラを有効にしてください。



Win32 向けのコンパイラアーキテクチャの変更は、このプラットフォーム向けにも、C++11 言語仕様のフルサポートを提供することになりました。これには、auto、nullptr、スマートポインタ、ラムダ式、ムーブセマンティックス、範囲ベースの for ループ、override と final、強い型付け enum 型、非メンバーの begin() / end()、static\_assert、type traits などが含まれます。

この変更によって、RAD Studio（および C++Builder）では、Win32 と Win64 でのコードの同等性が保障されることになりました。



つまり、RAD Studio 10 Seattle は、RAD 向けの拡張を伴う Clang ベースの C++コンパイラを Windows とモバイルの双方に提供し、Windows 10 向けの開発に対応した唯一の RAD C++環境であることができます。これは、Windows に特化した開発にも、クロスプラットフォーム開発にも利用でき、密接に統合された Windows 向けの VCL およびクロスプラットフォーム向けの FireMonkey フレームワークと、C++11 言語のフルサポート、モバイル向けの C++ARC（Automatic Reference Counting）ベースのメモリ管理を含む強力な機能を活用できるのです。新しい C++コンパイラの搭載により、従来の C++コードを抱える開発者にとっても、新たに C++開発を行う開発者にとっても、RAD Studio 10 Seattle は、導入を検討すべき製品となったことは間違いありません。

#### メモ：

bcc32c（新しい Clang ベースの Win32 C++コンパイラ）では、Win64、モバイル向けコンパイラと同様、Unicode 文字列リテラルおよび文字定数の表記の際に、u を頭に付ける必要があります。古い Windows 向け C++アプリケーションコードを移植する際には注意してください。

例） bcc32: "メッセージ" → bcc32c: u"メッセージ"

## 2. IDE の大幅な性能改善を実現

RAD Studio 10 Seattle では、IDE とビルド管理システムを大幅に改善し、利用可能なメモリをおよそ 2 倍に拡張しました。これにより、開発者は、より安定性があり、利用可能なプロジェクトサイズがより拡大した、ハイパフォーマンスな開発環境を利用できるようになり、複数のプロジェクトやマルチプラットフォームプロジェクトを持つプロジェクトグループといった、大規模なプロジェクトのビルドも可能になりました。

ここ数年、多くの機能が IDE に追加されるとともに、開発者のコード量も増大する傾向にあり、その結果、IDE でメモリ不足の問題が発生していました。RAD Studio 10 Seattle とその内部ライブラリの多くで、Windows Large Memory Address モデルをサポートするようにリファクタリングされました。これにより、IDE の利用可能メモリが、2GB から 4GB に拡張されたのです。

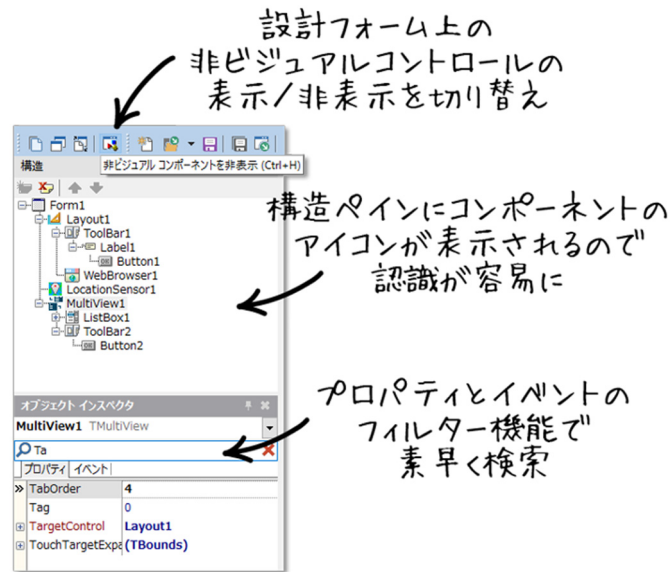
プロジェクトグループで複数のプロジェクトを再ビルドしたり、同じプロジェクトで複数のターゲット向けにビルドするようなより複雑なケースに対応するために、内部でコンパイルされたユニットのキャッシュアーキテクチャも完全に再設計されました。これにより、利用可能メモリが減少してきたときには、これらのユニットキャッシュをシステムが破棄するようにしました。

さらに、サードパーティツールや独自のプラグインなどによって拡張を加えるなどして、より複雑な状況下で仮に IDE がクラッシュしたとしても、RAD Studio 10 Seattle に新たに搭載された自動保存リカバリの機能により、保存されていない作業結果を復元することができます。

これらは、大幅な機能拡張ですが、これが IDE の強化ポイントのすべてではありません。今回のリリースでは、このほかにも、これまで RAD Studio ユーザーから寄せられてきた、いくつもの開發生産性強化の改善が加えられています。以下は、その主要な強化点とその画面ショットです。

### オブジェクトインスペクタ、構造ビュー、非ビジュアルコンポーネントの非表示

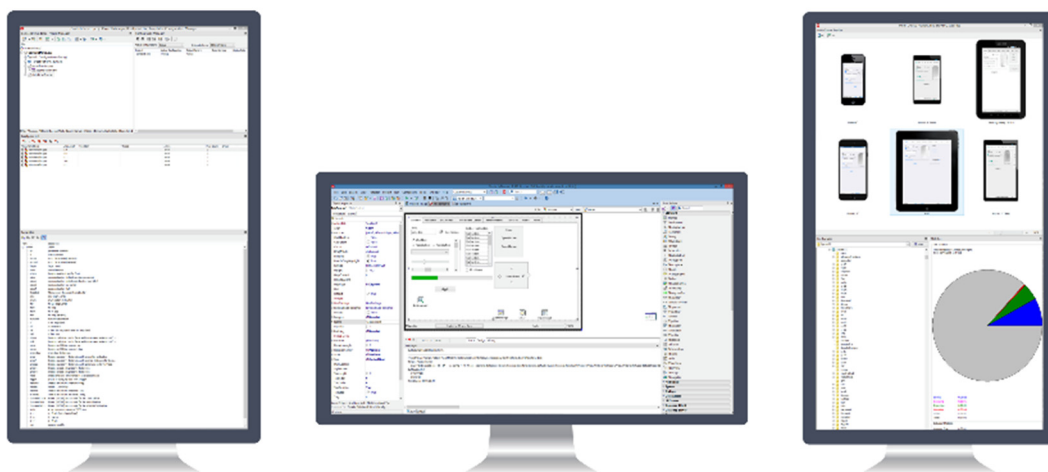
今回のリリースでは、開発者がコンポーネントのプロパティ/イベントの設定に使用するオブジェクトインスペクタ、デザイナ上に配置されたコンポーネントの階層構造を表示する構造ビューなど、従来から製品に搭載されてきた基本機能に改善が施されています。最初の改善は、オブジェクトインスペクタのフィルター機能です。これは、プロパティ名やイベント名を、検索ボックスに数文字のテキストを入力してフィルタリングできるものです。数多くのプロパティリストの中から、すばやく目的の項目を見つける場合に便利です。また、構造ビューに表示されるコンポーネントリストには、それぞれコンポーネントのアイコンも表示されるようになりました。これにより、目的のコンポーネントを見つけるのが容易になります。



また、ツールバーには、ビジュアルデザイナーで非ビジュアルコンポーネントの表示／非表示を切り替えるボタンが追加されました。これにより、非ビジュアルコンポーネントを多数配置したフォームでの UI 設計で、設計画面が混乱することを防ぐことができます。

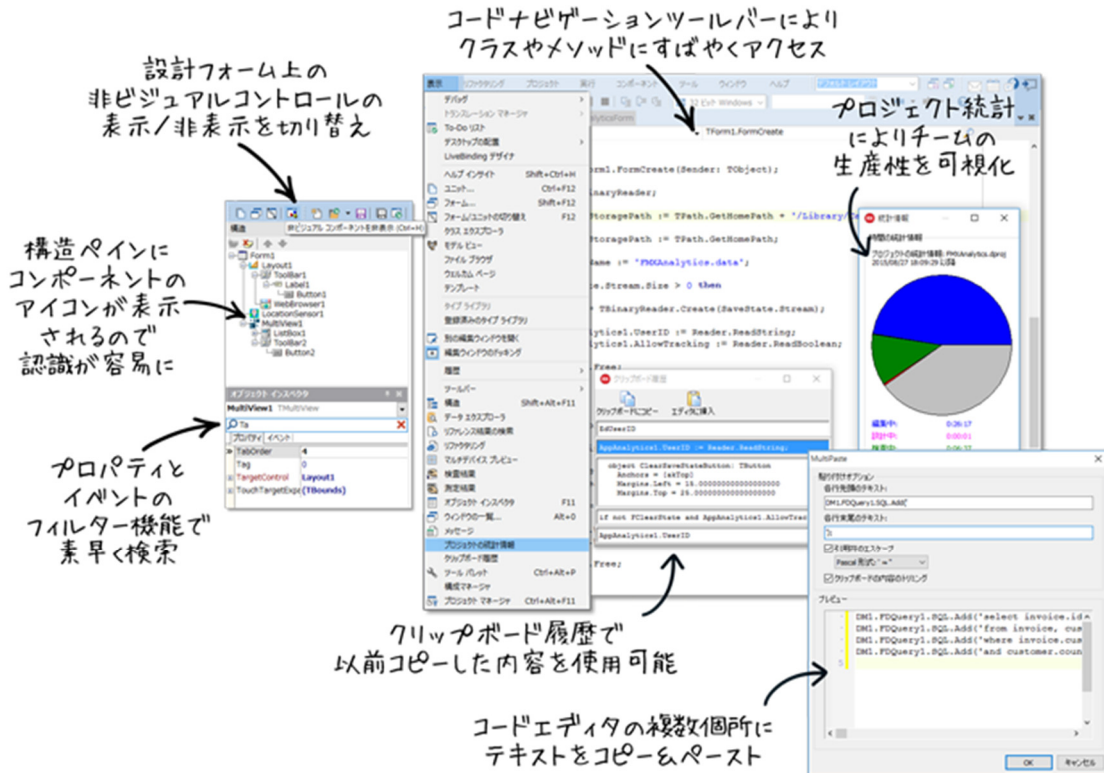
## マルチモニタサポートの拡張

RAD Studio 10 Seattle では、さらに IDE のマルチモニタのサポートを拡張しました。コアライブラリが Windows 8.1 以降をサポートし、各モニタの DPI に応じて適切な解像度でそれぞれのスケールリングに対応するようになりました。さらに、複数モニタで、それぞれについてフルスクリーンモードに設定できるようになり、マルチモニタ環境でのウィンドウの配置が簡単になりました。



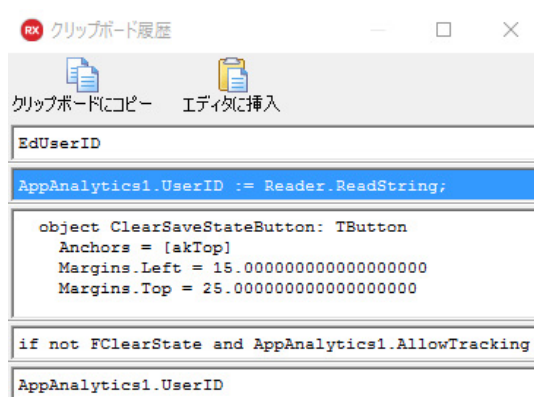
## IDE の生産性強化

IDE の生産性サポート機能についても、大幅な機能強化が施されています。以下は、その概要をまとめたものです。



これらの機能のいくつかは、前バージョンでは外部プラグインとして提供されてきたものです。これが、新バージョンでは、IDE のコア機能に統合されています。

- 活動をトラッキングしチームの生産性をよりよく理解するためのプロジェクト統計情報
- 貼り付け、切り取り、コピーの操作を保持し、複数の貼り付け操作を一度に行えるようになるクリップボード履歴



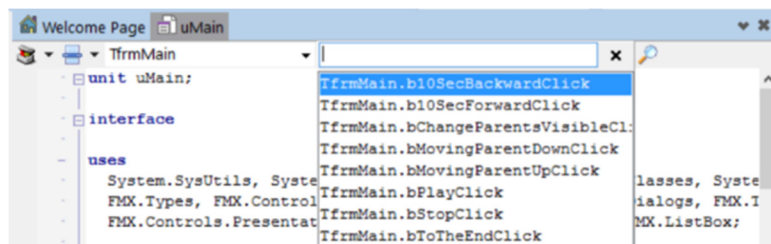
- 複数のソースコード行への同じ貼り付け操作を一度に行えるマルチペーストサポート
- スマートキー
- 実行フローがエディタ上でどこにジャンプするのかを見ることができる実行フローコントロールハイライト
- エディタの構造強調表示

```

Attr: Integer;
begin
  Path := TPath;
  Attr := faRes;
  FindFirst(Pat
  if F.name <>
  begin
    ListBox1.It
    while FindN
    ListBox1.
  end;
  FindClose(F);
end;

```

- コード ナビゲーション ツールバー



また、関数の同期と呼ばれる Object Pascal (Delphi) 向けのリファクタリングの新機能も追加されました。これは、いずれかの箇所で修正された引数リストを同期させることにより、同じ関数の宣言と定義を同一に保つことができるものです。

## 統合ツールの改善

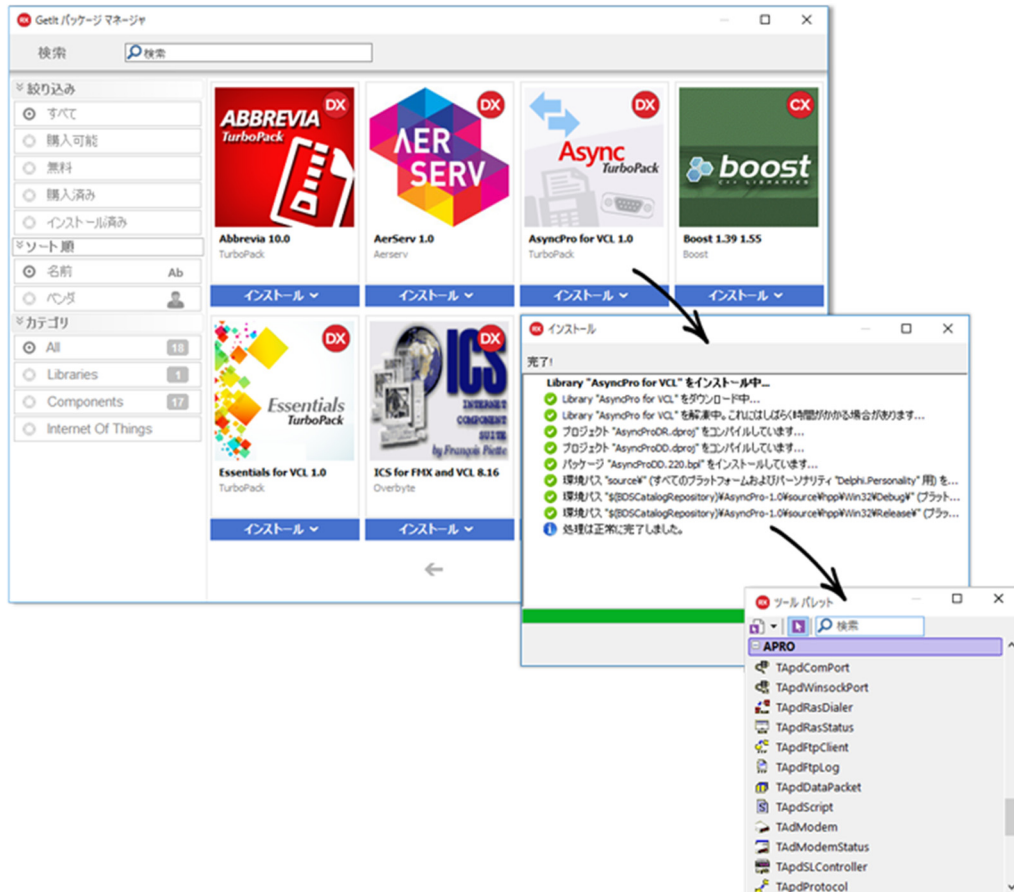
IDE に統合されたいくつかのツールにも改善があります。特に、ユニットテストフレームワーク DUnitX (XE8 から導入されたオープンソースプロジェクト) で、新たにモバイルサポート (Android および iOS) が追加されました。DUnitX は、最新の Delphi ユニットテストフレームワークであり、以下のサイトでホストされています。

<http://github.com/VSoftTechnologies/DUnitX>.

RAD Studio には、バージョンコントロールシステムのサポートも統合されています。Subversion、Git、Mercurial をサポートしていますが、今回 Subversion ライブラリのバージョンが 1.8.3 にアップデートされました。

## IDE 内で開発環境を拡張可能

今回のバージョンでは、GetIt ライブラリマネージャも改善されました。GetIt は、コンポーネントおよびライブラリのオンラインリポジトリを閲覧するためのカタログツールです。リポジトリには、RAD Studio で利用できる追加のライブラリ、Object Pascal および C++向けの著名なオープンソースライブラリ、エンバカデロが新たに提供を開始した追加のアドオン製品やコンポーネントが含まれます。



GetIt は、エンバカデロの GetIt サーバーから提供されるソースコードライブラリ、コンポーネント、その他の機能を、簡単に検索、ダウンロードし、アップデートがあれば更新することができるライブラリマネージャです。インストールの手順は極めてシンプルです。リストから選択（あるいは検索）し、C++か Object Pascal のいずれのバージョンかを選択、（必要に応じて）ライセンス規約に同意すれば、あとは、自動的にダウンロード、インストール、IDE への設定が行われます。

このほかの「オンライン接続」機能には、コミュニティツールバーがあります（XE8 のアップデート版から提供）。これは、エンバカデロのオンラインコミュニティ <http://community.embarcadero.com> へのアクセスを提供するものです。ツールバーには、未読メッセージを通知するボタン、イベントへのアクセ

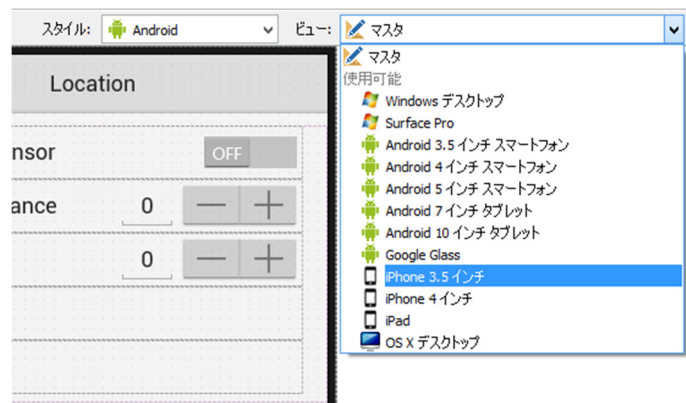
ス、Answer 機能へのアクセスが可能です。これにより、オンラインコミュニティへの質問は、IDE からワンクリックで行えるようになりました。

## FireUI マルチデバイスデザイナー

RAD Studio の特筆すべき機能、つまり、さまざまな種類のデバイスやフォームファクター向けに最適化可能な単一の共有 UI を構築できる機能を紹介しない訳にはいきません。

RAD Studio では、共有 UI を作成し、デスクトップフォームファクタとモバイルフォームファクタなどの複数のビューを使って、それぞれのデバイス向けに最適化を行うことができます。まず、中核となる UI を「マスター」ビューとして作成します。マスタービューは、サポートされているあらゆるプラットフォーム向けにスタイル設定できます。あるデバイスやオペレーティングシステムに特化したビューを作成する際に、デバイスの種類ごとに UI コンポーネントを（さらにそれ以外の非ビジュアルコンポーネントさえも）配置したり、プロパティを変更したりして、デバイス固有のカスタマイズが可能です。

次の図には、使用可能な定義済みデバイスフォームファクタのリストが表示されています。これらのリストのいずれのフォームファクターもアクティブにすることができ、そのビューがいったんアクティブになると、図の右側にあるように、それが先頭セクションに表示されます。マルチデバイスデザイナーでは、このほかに、マスタービューのスタイルも選択することができます。この例では、Android を選択しています。

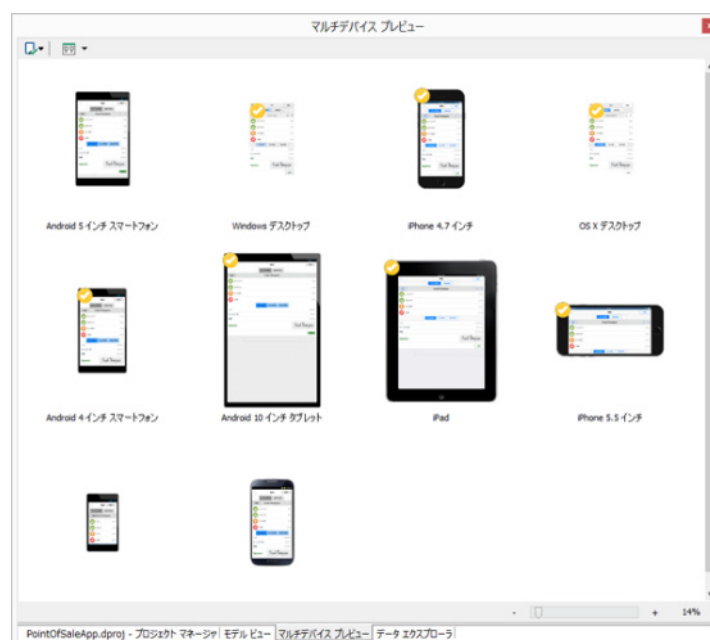


特定のビューで生じた差異は、別の FMX ファイルに保存されます。このファイルには、特定のビューとマスタービューの差分のみが格納されます。つまり、マスターに変更を加えることで、その変更内容は、対象の値をオーバーライドしていない各ビューに自動的に反映されます。一方、各ビューでは、コンポーネント個々のプロパティやコンポーネント全体をマスター値に戻すこともできます。つまり、マスターには、すべてのビューに反映されるべき変更をいつでも加えることができ、特定のビューには、そのビューにのみ最適化された変更を加えることができるのです。デバイスごとにレイアウトをカスタマイズ

ズできるだけでなく、プラットフォームごとに必要に応じてカスタムスタイルを読み込むこともできます。

それでも、必要な「フォームクラス」またはオブジェクトインスタンスはすべてのフォームファクターにわたって 1 つだけで済むので、開発者は単一のソースファイルを作成し、すべての共通 UI イベントをその 1 つの共有ソースファイルで管理することができます。この機能により、デバイス固有のビューをカスタマイズ／最適化できるので、真に単一ソースのマルチプラットフォーム開発ソリューションとなり、コード中に、プラットフォーム固有の IFDEF 指令を記述しなければならない状況はいっそう少なくなります。これらのビューにおいては、そのコンポーネントのプロパティの値、位置、表示／非表示は、それぞれ異なる可能性があり、イベントに関連付けられているハンドラさえ、ビューによって異なることがある点に注意してください。

特定のフォームファクターのビューに着目し、その UI を修正したい場合には、FireUI のカスタマイズ機能は便利です。しかし、実際にはターゲットとするすべての解像度のすべての OS について、どのように表示されるかを確認する必要があります。そこで、FireUI ではプレビューペインを用意し、異なる OS（Android、iOS、Windows、OS X）の異なるデバイスのフォームファクター（スマートフォン、タブレット、デスクトップの各解像度）を選択し、その表示を確認できるようにしました。プレビューペインでは、多様なデバイスを様々な配置、ズームレベルによって表示でき、デザイナー上でのビジュアルコンポーネントの追加／変更に対し、ライブ同期して更新します。もちろんプレビューには、適切なプラットフォームスタイルが使用され、マルチデバイスデザイナーのカスタムビューで行ったカスタマイズが反映されています。



類似のプレビュー機能は、いくつかのプラットフォームでデザイナー向けに提供されていますが、RAD Studio のマルチデバイスデザイナープレビューは、iOS と Android の双方に対応しており、Windows と OS X への対応でも利用できることが他のツールと大きく異なる点です。さらに、カスタムフォームファクタを定義し、あらかじめ定義されたものに追加して利用できるのも、企業が標準デバイスとして選定した特定のデバイス向けに、解像度やピクセル比などを定義してプレビューを行うことも可能です。

この革新的なマルチデバイスデザイナーとプレビュー機能は、他のモバイル開発あるいはマルチデバイス開発のフォーム設計機能をはるかに凌駕する強力なものです。

### 3. 強力な Windows 10 サポート機能

新しい C++コンパイラや IDE の機能強化もすばらしい新機能ですが、「強力な Windows 10 サポート」こそが、今回のバージョンの最も重要なポイントです。これは、いわゆる宣伝文句で言っているのではありません。というのは、マイクロソフトが「WinRT」という新しいアーキテクチャを作成し、開発者にこのアーキテクチャへの移行を促しています。これには多くの修正が必要ですが、RAD Studio では、従来の Win32/Win64 API ベースのアプリケーションを、WinRT API の多くを活用できる同様のアプリケーションへと移行できるように、Windows 10 サポートを実現しています。現時点で RAD Studio は、Windows 10 への移行パスとして最適なこれらの機能を有した唯一のツールなのです。

RAD Studio の VCL (Visual Component Library) は、ネイティブ Windows アプリケーション開発で、最もすぐれたオブジェクト指向コンポーネントライブラリの地位を守ってきました。今回、10 Seattle で、Windows 10 の新しい UI をサポートする UI コントロールや、WinRT の機能と統合されたコンポーネント（通知やコントラクトなど）の提供によって拡張が加えられたことにより、改めてよく設計された洗練されたライブラリであることが認識されました。すでに数百万のアプリケーションが VCL によって記述されていますが、これらが将来、すなわち Windows 10 に容易に対応することができるのです。

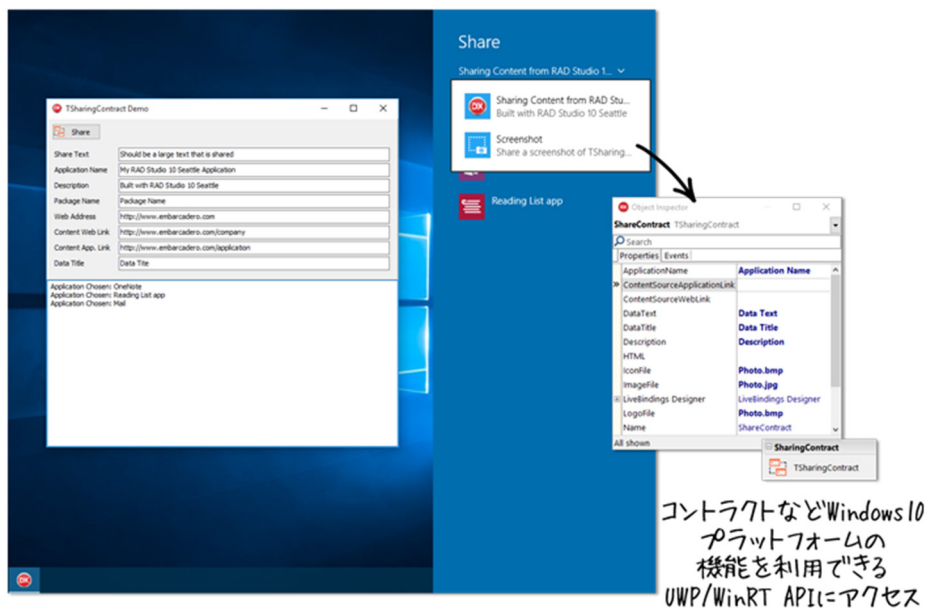
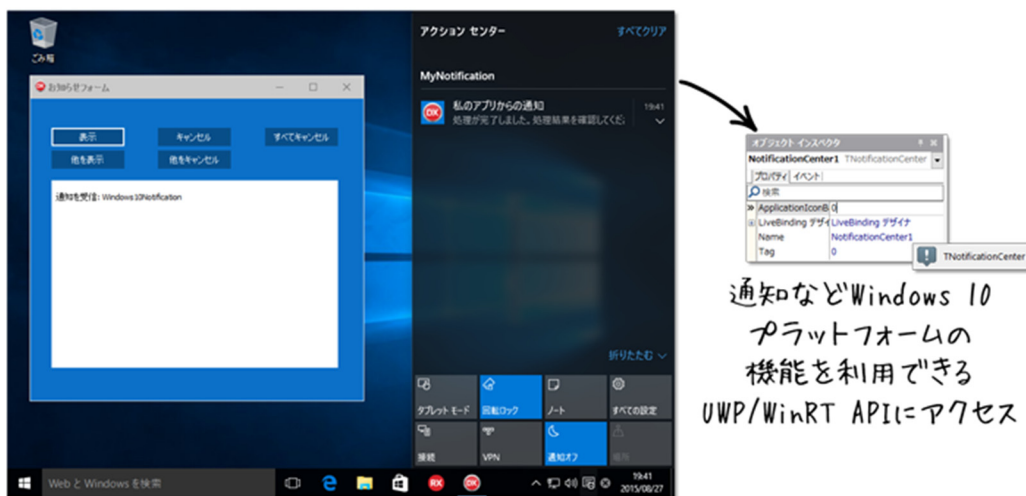
#### WinRT/UWP の統合

上述したように、Windows 10 との統合の重要な要素は、VCL および FireMonkey アプリケーションでマイクロソフトの新しい UWP (Universal Windows Platform) の機能をサポートすることです。これは、RAD Studio 10 Seattle で、WinRT API を、ダイレクトマッピングレイヤー (WinRT API の Object Pascal 言語へのマッピングを含む) を使って利用できるようにしたことと、WinRT で使用するデータ型のサポートにより可能になりました。

これらの API サポートによって、Windows 10 のいくつかの新機能が利用できるようになっただけでなく、Windows 10 の通知やコントラクトのように、非ビジュアルコンポーネントとしてこれらを利用可能にしたことで、VCL および FireMonkey アプリケーションで、簡単に利用できるようになりました。

さらに、エンバカデロでは、マイクロソフトの「Project Centennial」の動向も注視しています。Project Centennial は、従来の Windows アプリケーションを Windows Store アプリとして配布できるツールキットと仕組みを提供するものです。これにより、RAD Studio で作成するアプリケーションでより多くの WinRT 機能が利用できるようになり、RAD Studio でコンパイルしたアプリケーションを Windows 10 Store で公開できるようになります。エンバカデロでは、Project Centennial が利用可能になった時点で、この機能のフルサポートを提供することを計画しています。

以下は、製品に含まれる VCL アプリケーションの通知およびコントラクトサポートのデモ画面2点です。

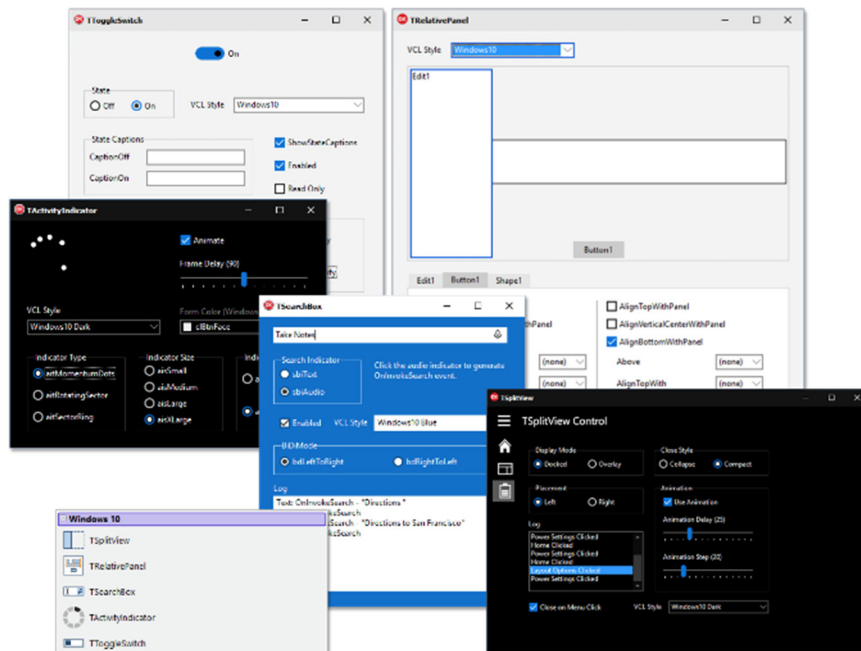


## 新しい Windows 10 VCL コントロール

VCL は、RAD スタイルのコンポーネントベース開発を可能にする、Windows 開発者向けの実績あるライブラリです。数百の機能豊富なビジュアルおよび非ビジュアルコンポーネントが用意されています。RAD Studio 10 Seattle では、Windows 10 の UI および UX のサポートを容易にする大幅な拡張が加えられています。新しいネイティブ VCL コントロールは、Windows 10 UI 要素のコモンコントロールにマップ（これらはマイクロソフトの WinRT コントロールによって提供）されていますが、独立性を保ち、既存の VCL アーキテクチャをベースにしています。このことは、RAD Studio 10 Seattle で構築した Windows 10 UI コントロールを使ったアプリケーションは、Windows 8 や Windows 7 でも動作するということを意味しています。

5 つの新しい UI コントロールの画面ショットを以下に示します。そのあとで、それぞれについて紹介しましょう。

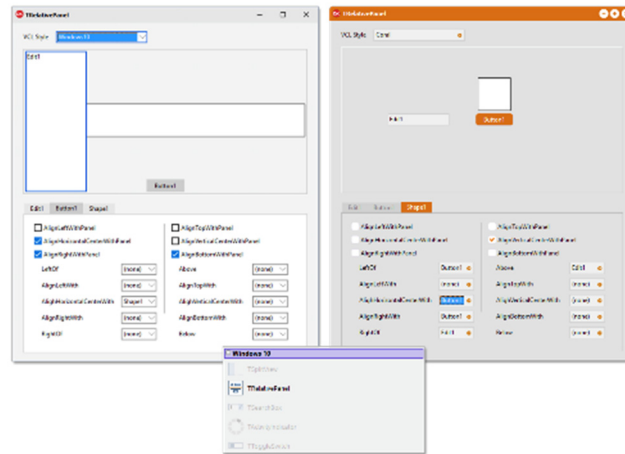
- RelativePanel
- ToggleSwitch
- SearchBox
- SplitView
- ActivityIndicator



これらのサンプルは、それぞれ、製品デモの Object Pascal/VCL および CPP/VCL フォルダに格納されています。

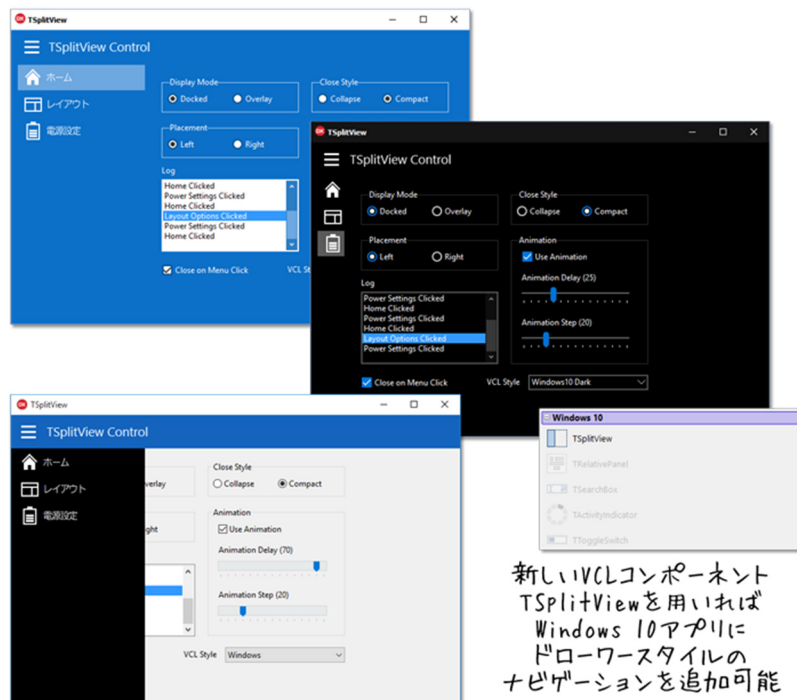
## TRelativePanel

RelativePanel は、子オブジェクトの位置や整列方法を、他のオブジェクトや親のパネルとの関係で設定できる新しいレイアウトパネルです。例えば、あるテキストを常にパネルの左側に配置したり、常に下に配置するといったことが指定できます。マス目状にレイアウトを定義できないようなユーザーインターフェイスを構築したいときに、GridPanel の代わりとして TRelativePanel を使ってください。



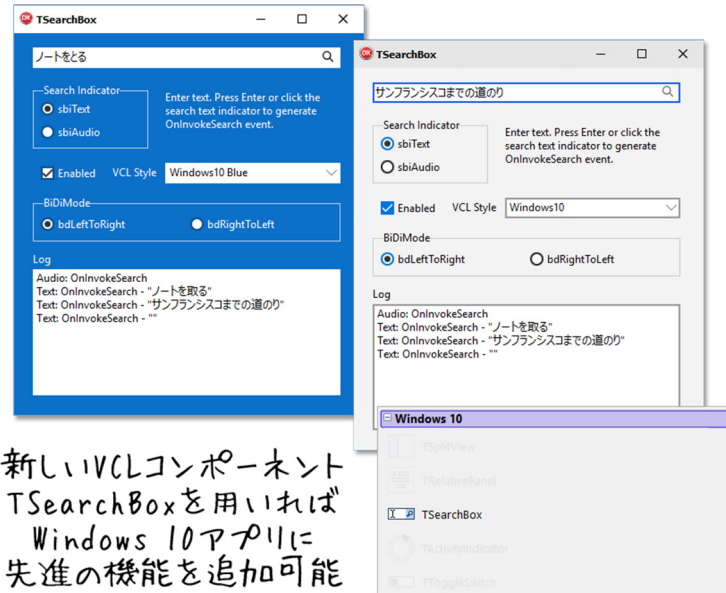
## TSplitView

SplitView コントロールは、一時的なコンテンツを簡単に表示／非表示可能にします。例えば、トップレベルのナビゲーションメニューにこれを使い、必要に応じてナビゲーションメニューの内容を表示したり非表示にできます（これを Windows 10 では「ハンバーガーメニュー」と呼んでいます）。



## TSearchBox

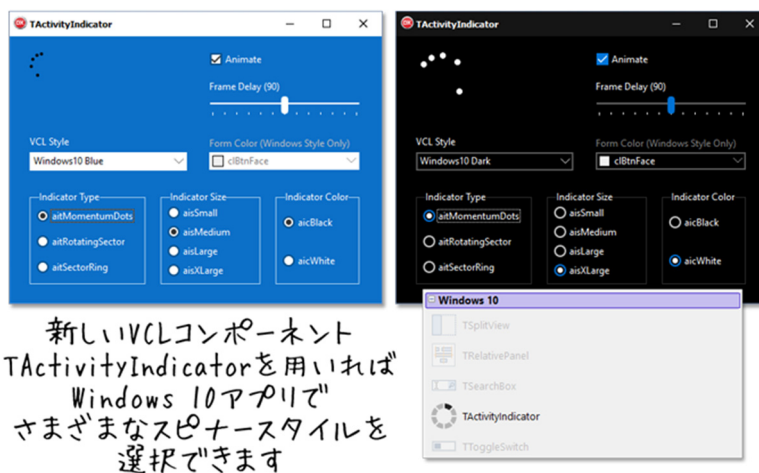
検索機能を提供する Edit コントロールです。FireMonkey の SearchBox と同等です。



新しいVCLコンポーネント  
TSearchBoxを用いれば  
Windows 10アプリに  
先進の機能を追加可能

## TActivityIndicator

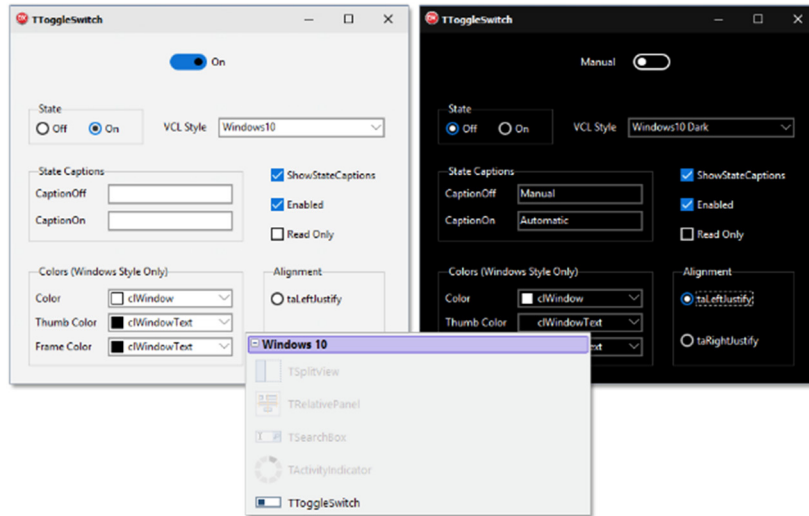
このコントロールは、進捗量が不定のプログレスリングインジケータを表示します。このインジケータの表示形式にはいくつかの種類があり、大きさ、アニメーションのスピード、色などいくつかのプロパティによってカスタマイズができます。



新しいVCLコンポーネント  
TActivityIndicatorを用いれば  
Windows 10アプリで  
さまざまなスピナースタイルを  
選択できます

## TToggleSwitch

このコントロールは、実際のスイッチを模したトグルスイッチで、オン／オフの操作が可能です。コントロールには、on と off の2つの状態があり、それぞれにキャプションを設定したり、削除したりすることができます。

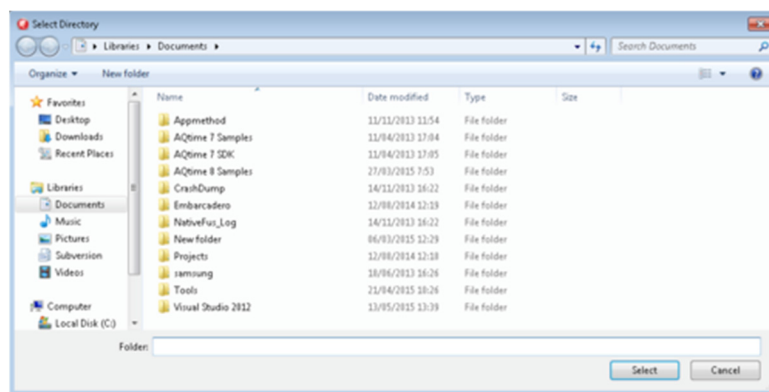


## 高解像度モニタのサポートとマルチモニタの改善

WinRT/UWP サポートと新しい VCL コントロールのほかに、今回のリリースで追加された Windows 10 サポートには、いくつかの機能改善が含まれます。そのひとつが、VCL アプリケーションにおける 4K モニタのサポート、より正確には、IDE のプロジェクトオプションで直接アプリケーションマニフェスト設定を変更できるようになったことです。マニフェスト設定は、最近のバージョンの Windows でアプリケーションの動作を変更します。これにより、低解像度を模倣して出力を拡大して、しばしばぼやけた UI になるのではなく、フル高解像度の出力が得られます。

その他の改善には、マルチモニタシステムのサポート改善があります。これは、Windows 8.1 と Windows 10 でマイクロソフトがサポートしたように、VCL においても、異なる DPI の複数モニタをハンドルできるようにするものです。

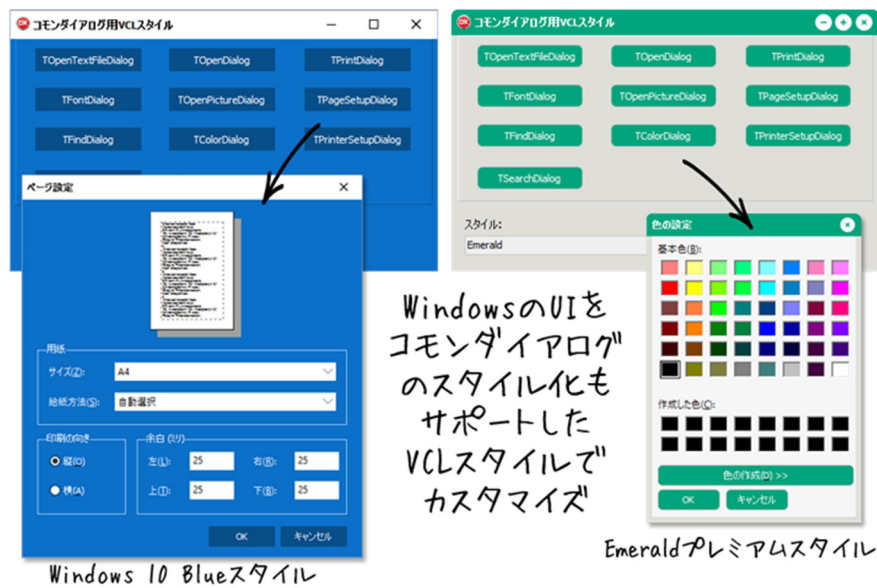
そして、VCL の SelectDirectory 関数もアップデートされました。この関数は、フォルダを選択するシステムダイアログボックスを表示します（この機能は、RAD Studio IDE に対しても VCL ライブラリと同様にアップデートが適用されています）。次の画面ショットは、新しい SelectDirectory ダイアログです。



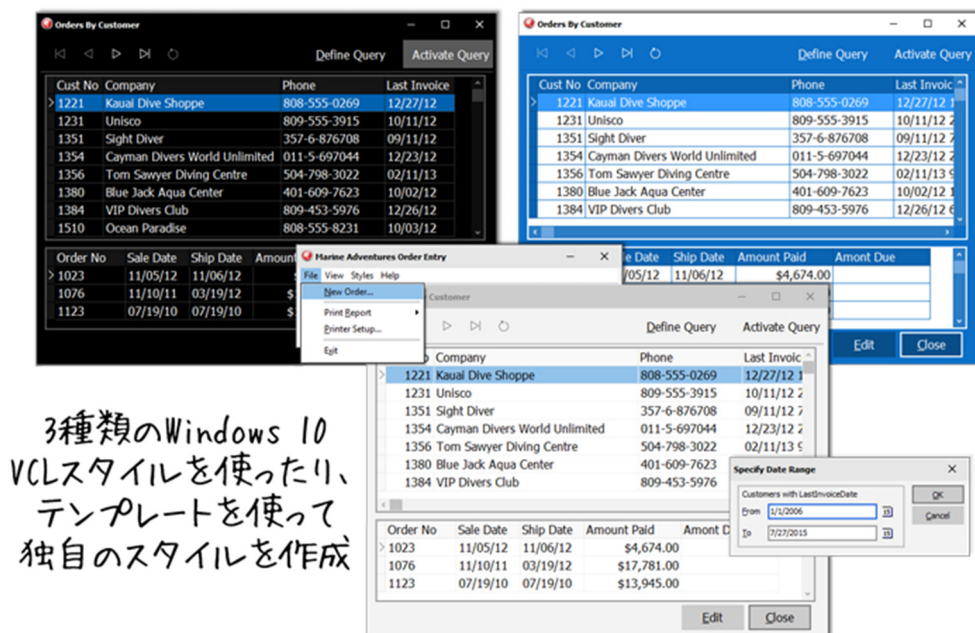
## VCL スタイルの改善

VCL では、数年前に UI スタイリングテクノロジーを導入しました。このテクノロジーは、継続的に改善されており、新しいスタイル（今回は Windows 10 固有のスタイル）の提供とともに機能改善もなされています。

機能改善の面では、VCL スタイルで、新たにコモンダイアログ（OpenDialog, ColorDialog など）のスタイルをサポートしました。



以下は、同じアプリケーションにいくつかの Windows スタイルを適用した例です。



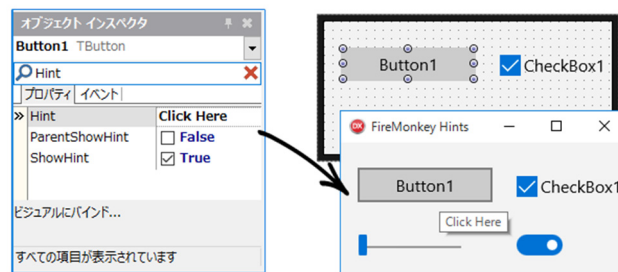
RAD Studio でアプリケーションにスタイルを設定するのはとても簡単で、プロジェクトオプションで、有効なスタイルを選択するだけです。いくつものスタイルがあらかじめ用意されており、ユーザーが実行時に選択したり、別ファイルとして後で提供することもできます。

## Windows 向けの FireMonkey の強化

Windows 10 の重要性を考慮し、今回のリリースでは、VCL だけでなく FireMonkey についても Windows デスクトップアプリケーション構築に関する大幅な改善を加えています。FireMonkey アプリケーションでも、Windows 10 の WinRT 統合（およびコンポーネント）によって提供される機能は同じものを利用できますが、UI 側についてはいくつかの大きな違いがあります。それは、FireMonkey がマルチデバイス開発向けに設計されていることに起因します。

注目すべき点は、Windows 10 通知を利用するために使用するコンポーネントは、iOS と Android の通知にも使えるため、いずれも同じコードで利用できるということです。

FireMonkey の Windows 向け UI に関する改善については、以前のバージョンでは、利用できなかった Windows 向け機能、例えば、マウスオーバーヒントの表示（この機能はモバイルでは不要ですが、デスクトップアプリケーションでは重要です）が利用可能になったことなどがあります。



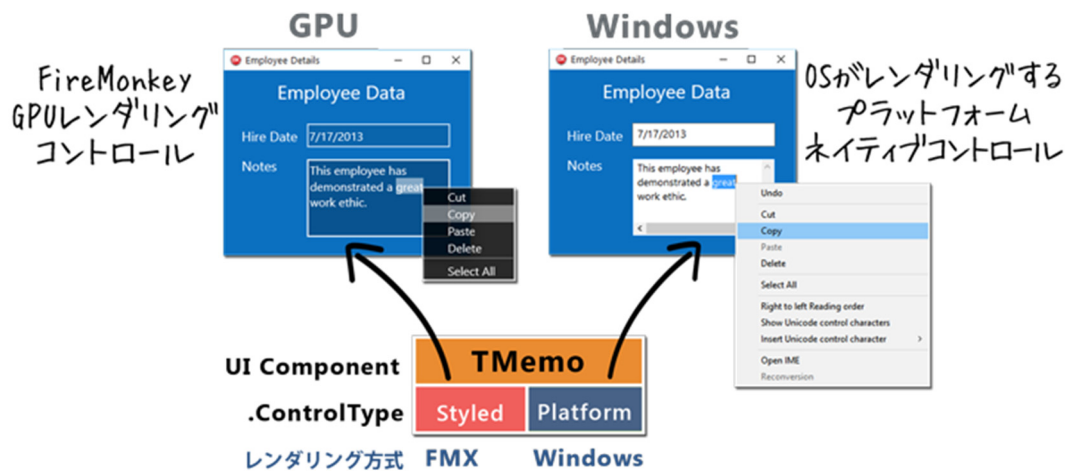
FireMonkey UIコントロールにヒントを表示可能

そして、Windows 上での FireMonkey の改善で最も大事なものは、Edit と Memo について Windows のネイティブスタイル表示をサポートしたことです。FireMonkey ライブラリは、開発者に同じビジュアルコントロールで、スタイル表示（FireMonkey のスタイルアーキテクチャによってコントロールをレンダリング）か、プラットフォーム表示（実行時に、ライブラリがネイティブのプラットフォームコントロールをインスタンス化）を選択でき、まったく同じプロパティによってコントロールを制御することを可能にしました。これにより、開発者は、プラットフォームコントロールを、単一コードベースでの開発をあきらめ、コードを変更するといった労力を伴うことなく、利用可能になりました。

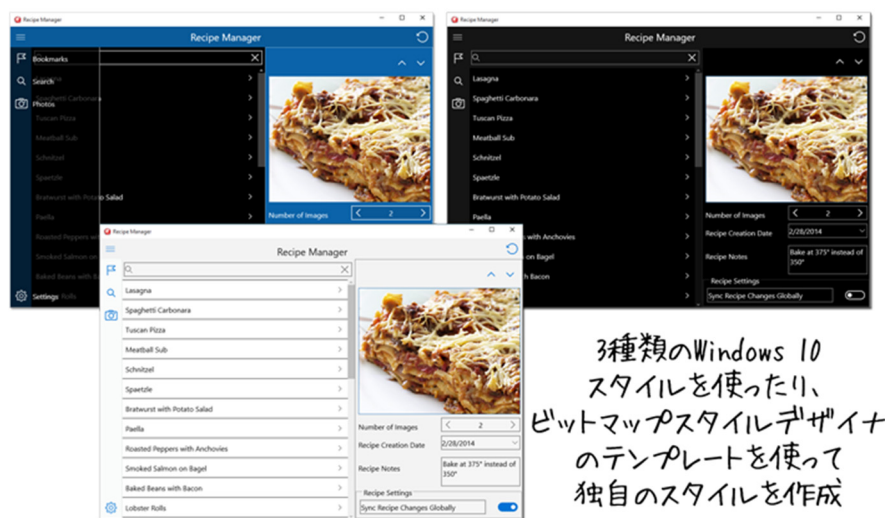
いくつものコントロールでプラットフォームネイティブレンダリングが選択可能になったのに加えて、FireMonkey では、Z オーダー（ユーザー側から見たコントロールの位置関係）の管理についても改善さ

れました。これは、Windows および iOS で、プラットフォームコントロールのレンダリングのために行われたものです。これにより、開発者は、同じフォーム内にプラットフォームコントロールとスタイルコントロールを混在させることが可能になり、ネイティブプラットフォームで最適な UX を提供できます。

このアーキテクチャは、はじめに iOS で追加されました（後のセクションで説明します）が、RAD Studio 10 Seattle では、Windows プラットフォーム向けにも拡張し、Edit および Memo コントロールを従来のスタイル表示と、Windows ネイティブの EDIT コントロール（単数行または複数行）をラップしたもののいずれかを選択できるようになりました。



このほかの Windows 10 向けの FireMonkey の機能強化には、テキストとイメージの両方をコピー／ペーストできるようにするクリップボードサポートの改善、Windows 10 のルック＆フィールに適合させる MultiView および Switch コンポーネントの改善、そして Windows 10 FireMonkey スタイルのサポートがあります（下図参照）。



これらのスタイルは、他の FireMonkey スタイルとともに、以下のフォルダに格納されています。

C:\Users\Public\Documents\Embarcadero\Studio\17.0\Styles

## 4. モバイルおよび IoT サポートの拡張

導入部と FireUI マルチデバイスデザイナの紹介で触れたように、RAD Studio は Windows 開発向けの強力なソリューションであるだけでなく、単一コードによるマルチデバイス開発が可能な、モバイル向け開発ツールの中ではユニークな機能を提供する強力なツールでもあります。

RAD Studio は、マルチプラットフォーム向けにスタイル表示とプラットフォームネイティブ表示の双方をサポートするユーザーインターフェイスコントロールが用意された FireMonkey フレームワークと、Windows、OS X、Android、iOS といった複数のプラットフォーム向けのセキュアなネイティブアプリケーションを生成できる複数のコンパイラを搭載しています。コンパイラは、それぞれのターゲットごとに、Intel または ARM チップ向けのネイティブコードを生成します。

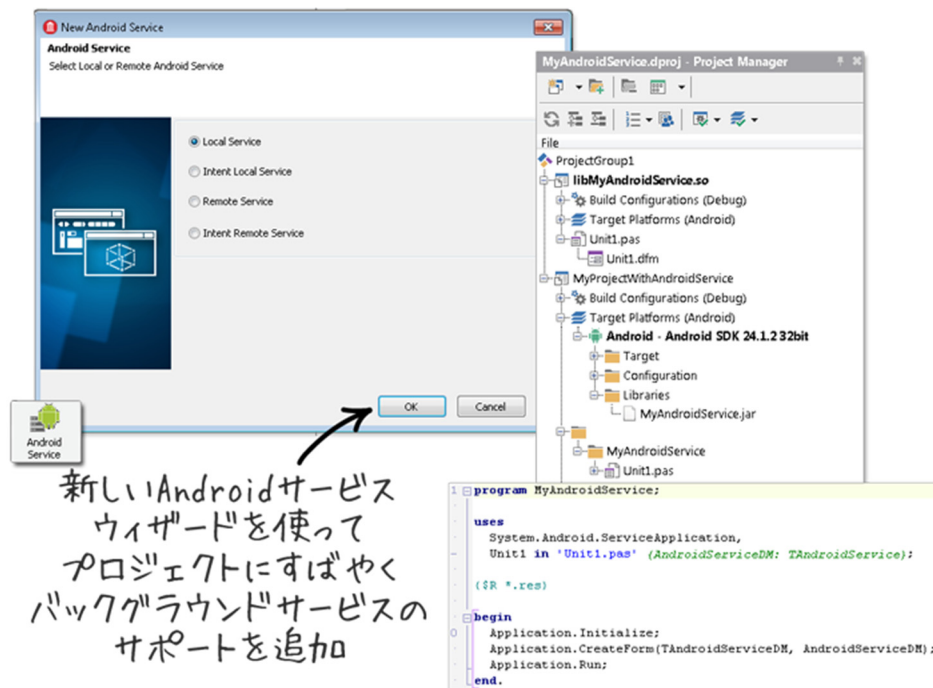
FireMonkey のアーキテクチャ、そのユニークな特徴、単一コードからマルチデバイスアプリケーションを構築可能にする開発モデルなどを解説するには、もっと多くの紙面を必要とします。そのため、以下のセクションでは、RAD Studio 10 Seattle のモバイル開発向けの新機能のみにフォーカスすることにします。

### Android サービスの利用

Android サービスアプリケーションの構築は、従来バージョンの RAD Studio でも Java を併用したプログラミングによって行うことはできましたが、新バージョンの RAD Studio では、簡単に Android サービスアプリケーションを作成し、通常の UI アプリケーションと同じ APK にバンドルすることができるようになりました。今回のバージョンでは、この機能は、Delphi (Object Pascal) のみで利用できます。C++ 言語のサポートは、将来のバージョンで提供される予定です。

Android サービスは、アプリケーションがフォアグラウンドになかったり、非アクティブなアプリケーションがスワップアウトされて終了しているときなどに有効なものです。アプリケーションがスワップアウトしている状態でも、サービスはコードをバックグラウンドで実行可能にします。

Android では、単一の APK に通常のアプリケーションとサービスを含めることができるようになりました。そして後者は、通知とインテントを使用できます。RAD Studio では、バックグラウンドサービスを開発し、モニタがデバイスに接続されたとか、リモートサービスなどによってこれを起動することができます。



RAD Studio の新バージョンでは、このほかにもいくつかの Android 固有の機能の改善が施されています。例えば、Android インテントの受信サポート、タッチアニメーションのサポートなどがあります。

## 10 Seattle での iOS サポート

iOS プラットフォームは、前バージョンの XE8 で特に強化され、iOS 64-bit ARM コンパイラの搭載や、プラットフォームコントロールに関連する点など、数多くの UI 開発の改善が施されました。

今回のリリースではこれがさらに強化されました。まず、iOS デバイスに配置された iOS 64-bit とユニバーサルアプリのフルデバッグサポートがあります。そして、プラットフォーム ScrollBox コントロールなどプラットフォーム表示の改善も継続しています。

iOS 向けプラットフォームネイティブレンダリングのコントロールには以下が含まれます。

- TMemo：複数行テキストエディタ
- TListView：スタイルバージョンでは提供されていない iOS 固有の機能やパフォーマンスに優れたリスト
- TSwitch
- TCalendar
- いくつかの Scrollbox コントロール



今回のリリースでは、このほかに、ListView コントロールにいくつかの改善が加えられました。これにより、コントロールの項目のレイアウトや構造を容易にカスタマイズできるようになりました。そして、コントロールがスタイル表示、プラットフォーム表示のいずれの場合でも、Visual LiveBinding によって、データとコントロールの要素を直接マッピングすることができます。

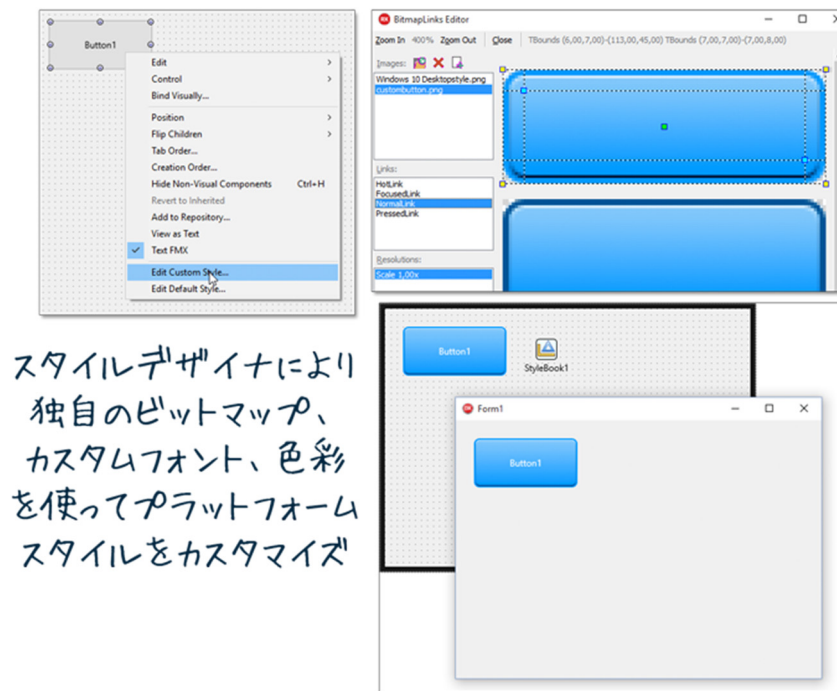


RAD Studio 10 Seattle では、新しい iOS 認証／プロビジョニング管理がサポートされました。これは、Xcode における Apple の要件の変更に対応したものです。RAD Studio の IDE では、OS X 上で実行されるプラットフォームアシスタントサーバー（PAServer）へのリモート接続を経由した、Apple 開発ツール

へのインターフェイスが提供されています。この改善により、RAD Studio の IDE から直接、特定のデバイスへの配置や Store への配置認証を行うことが可能になりました。

また、FireMonkey スタイルデザイナにも改善が加えられ、iOS だけでなくすべてのプラットフォームを包含する強化が施されています。ビットマップスタイルデザイナでは、新しいテンプレート、複数の異なるプラットフォーム向けのブランクスタイルの作成、Windows 10 スタイル向けのスタイルビューワーなどの改善が加えられています。

統合された FireMonkey スタイルデザイナの改善には、StyleBook コンポーネントで、さまざまなプラットフォーム向けのスタイルコレクションのサポートや、モバイル上の特定のコントロールへのスタイル設定が容易になるなどの改善が含まれます。以下は、あるコンポーネントのスタイルを編集している画面です。



## IoT デバイスを活用したソフトウェアの開発

RAD Studio では、IoT（Internet of Things：モノのインターネット）に関連して、数多くの機能がここ数回のリリースで追加されています。

- Wi-Fi デバイスとの HTTP プロトコル経由での通信を行うための REST / HTTP クライアントライブラリの統合
- Bluetooth および Bluetooth LE 通信のサポート

- iBeacons および AltBeacons 標準の双方に対応したビーコンのサポートと BeaconFence による位置検出（後述）
- 特定のデバイス向けのネイティブプラットフォーム SDK の統合
- Android および iOS 向けの組み込みデータベース IBLite（配布無料）を含む、モバイルデバイスでのローカルキャッシュ用のハイパフォーマンスデータベース

IoT ソリューション構築に役立つそのほかの重要な機能に AppTethering があります。AppTethering は、デスクトップアプリケーションとモバイルアプリをシームレスに統合可能にする技術で、ローカルガジェットと対話するモバイルデバイスから、既存の VCL アプリケーションなどに、直接情報を送信することができます。こうしたサポートを提供しているのは、モバイルデバイスが IoT デバイスとの通信に適したデバイスであり、デスクトップ PC やノート PC よりも広範囲に移動して、IoT デバイスとの協調が可能になるからです。

次の図は、モバイルアプリがカメラを使ってバーコードを認識し、その情報を通常の Windows デスクトップアプリケーションに送信している例です。



こうしたサポートに対し、10 Seattle では以下のような機能強化が行われています。

- Bluetooth/LE サポートとビーコンサポートの改善
- ビーコンに接続するデバイスのチューニングを行うための新しい TBeaconDevice クラス
- 新しい JSON ストリーミングライブラリによるデバイス通信でのデータ処理

## 5. データアクセス機能の強化

多様なデータベースへの接続性を提供するデータベースアクセスサポートは、RAD Studio のコンポーネントライブラリにおいて欠かせない要素です。現在 RAD Studio では、FireDAC と呼ばれるライブラリが、データアクセスの重要な役割を担っています。このライブラリは、多層、クライアントサーバー、そしてデスクトップおよびモバイルの双方のプラットフォームにおけるローカルデータベースのすべてで利用できる共通のデータアクセス技術を提供するものです。

FireDAC データベースコンポーネントおよびドライバは RAD Studio、Delphi、C++Builder に完全に統合されています。FireDAC を用いれば、Delphi および C++Builder から InterBase、SQLite、MySQL、SQL Server、Oracle、PostgreSQL、DB2、SQL Anywhere、Advantage DB、Firebird、Access、Informix などの、多様なローカル／組み込みデータベース、モバイルデータベース、エンタープライズデータベースに高速かつネイティブに直接アクセスできます。XE8 では、多くの大企業でデータウェアハウスシステムに利用されている Teradeta データベースを新たにサポートしました。RAD Studio 10 Seattle では、これに加えて、NoSQL データベースとして広く使われている MongoDB をサポートしました。

RAD Studio が、このように広範なデータベースアクセスをサポートしていることは重要です。他の開発ツールベンダーとは異なり、特定のデータベースに偏ることなく、すべての主要なエンタープライズデータベース（Oracle、Microsoft、IBM、SAP など）、主要なオープンソースデータベース（MySQL、SQLite、Postgress など）をサポートしています。RAD Studio では、エンバカデロが提供するハイパフォーマンスの軽量データベース InterBase XE7 を強力にサポートしていますが、これも他のデータベースの利用を阻害するものではありません。

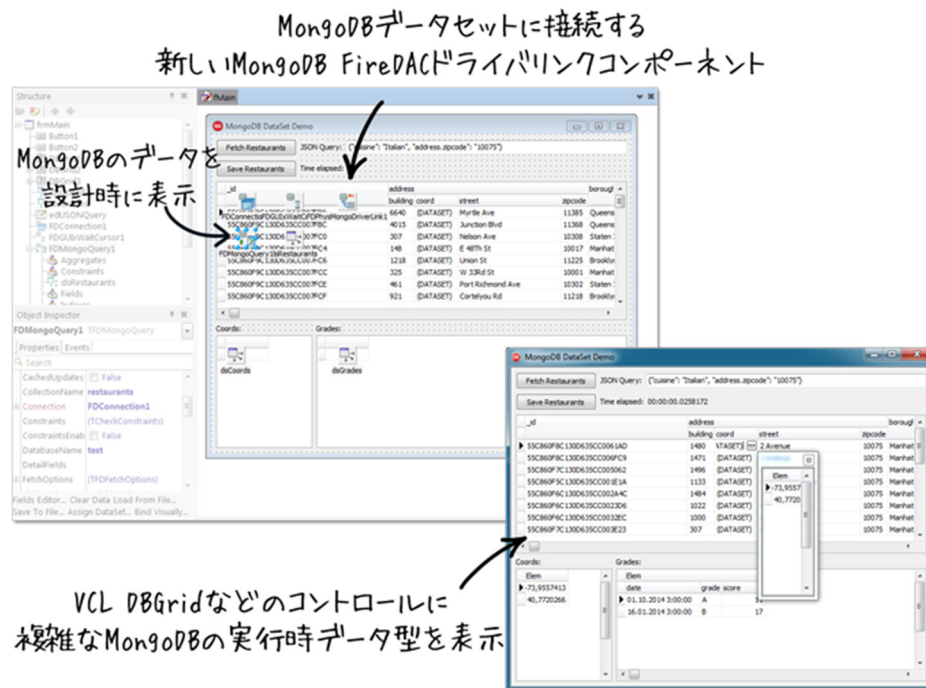
### FireDAC とは

FireDAC は、RAD Studio の共通データベースアクセスエンジンです。ハイパフォーマンスで簡単に利用できるエンタープライズデータベースアクセス機能を提供します。共通のデータアクセス機能を提供しながらも、それぞれのデータベースの機能もしっかりサポートしており、特に主要なデータベースシステムのパフォーマンスにフォーカスした機能が提供されています。FireDAC で提供されるドライバは、すべてソースコード付き（Enterprise 版以上）で提供されており、大変オープンです（いくつかのデータベースではベンダーのクライアントライブラリを必要とする場合があります）。

FireDAC の特徴については数多くありますが、要約するならば「FireDAC により、データベースアプリケーション開発が簡単かつ楽しいものになる」ということです。

## FireDAC の MongoDB サポート

RAD Studio 10 Seattle の FireDAC に関連する機能強化には、MongoDB サポートがあります。FireDAC がこれまでサポートしてきたリレーショナルデータベースとは大きく異なる MongoDB の特性を考慮して、他のリレーショナルデータベースとは異なる、このデータベース固有のサポートならびにコンポーネントが提供されることになりました。



FDConnection コンポーネントが MongoDB ドライバを新たにサポートし、基本的な操作については、他のデータベース向けのコードと共有することが可能になっています。しかし、MongoDB の NoSQL 固有の機能を活用するためには、他のクラスやコンポーネントを利用したほうが便利です。

以下は、MongoDB API をラッピングするクラスです。

- TMongoConnection: MongoDB サーバーへの接続、サーバー上のデータベースおよびオブジェクトコレクションへのアクセスを管理
- TMongoDatabase: 設定した MongoDB データベースのデータベース、コレクション、ユーザーを管理
- TMongoCursor: MongoDB カーソルから結果セットドキュメントをフェッチ

- TMongoCollection: MongoDB コレクション（リレーショナルデータベースのテーブルに相当する、同じ「ファミリー」に属する要素セット）を表す

以下は、MongoDB 固有のコマンドオブジェクトです。

- TMongoUpdate: MongoDB アップデートビルダーに対応
- TMongoSelector: MongoDB セレクタービルダーに対応
- TMongoInsert: MongoDB インサートビルダーに対応
- TMongoIndex: MongoDB インデックス定義ビルダーに対応

さらに、MongoDB 固有のデータセットコンポーネントも用意されています。

- TFDMongoDataSet: MongoDB カーソルにアタッチされるデータセット
- TFDMongoQuery: データベースコレクションに対して実行されるクエリー
- TFDMongoPipeline: 特定のコレクションに対して実行されるパイプライン

MongoDB の非テーブル型のデータ構造のために、RAD Studio のデータベースアーキテクチャのコア機能で多くのサポートを必要としました。TDataset クラスとそのアーキテクチャでは、すでに、他のデータベースやインメモリテーブルをサポートするために、抽象データ型とネストしたテーブルをサポートしていました。今回、この新しいデータセットが NoSQL データ構造をサポートしたことで、従来の DBGrid のようなコントロールで、こうした抽象データ型をグリッド上に表示できることを認識できることでしょう。

関連する RTL 側での大きな強化点として、JSON および BSON データ構造の処理改善があります。これらの形式は、MongoDB との接続においても利用されます。JSON および BSON のサポートについては、次の RTL に言及したセクションで解説します。

## 他の FireDAC の改善

このほかにも、FireDACにはいくつかの重要な改善が加えられています。これらは、他のデータアクセス技術とのギャップを埋めるものであったり、データアクセスライブラリの継続的な改善などです。

例えば、MERGE / INSERT OR REPLACE の SQL コマンド、及び dmAppend、dmUpdate、dmAppendUpdate モードのサポートを含む、TFDBatchMoveSQLWriter コンポーネントの最適化と拡張などがあります。

また、FireDAC のキャッシュアップデートモードの改善もあります。これにより、従来の dbExpress での ClientDataSet によるモデルに合致させることもできるようになりました。

## DataSnap の改善

RAD Studio 10 Seattle では、多層アーキテクチャとして製品に長く搭載され、幅広く利用されてきた DataSnap に関連して、小さいけれども大変重要な改善が施されています。

XE8 で新しい HTTP クライアントライブラリが導入され（関連情報については後述）、DataSnap による HTTP および HTTPS 接続では、クライアント側の接続レイヤーを使うようになりました。System.NET ライブラリを使うことで、アプリケーションは、プラットフォームの HTTP クライアント機能を活用できます（すべてのサポートプラットフォームが対象）。これにより、HTTPS 接続時にも、OpenSSL クライアントライブラリを配布する必要がなくなりました。

さらに、DataSnap / FireDAC 統合についても改善が加えられました（これにより DataSnap アプリケーションで FireDAC の活用を強化できます）。FireDACJSONReflect ユニットの改善により、サーバーとクライアント（特にモバイルで有効）間でのデータ送受信で、データ圧縮をサポートしました。また、FDSchemaAdapter コンポーネントを DataSnap で利用した例を示すサンプルも、製品に含めるようにしました（以前は EMS モジュールでのみサンプルを提供）。

## クラウドの拡張

RAD Studio では、（HTTP クライアントコンポーネントによってカプセル化された）ネイティブライブラリや Indy ソケットライブラリを用いた、強力なインターネット接続機能を提供してきました

最近のバージョンでは、次のような機能を容易に利用できるコンポーネントが追加されました。

- クラウドサポート（Amazon Web Services と Microsoft Azure が対象）
- REST 呼び出しサポート（統合 URL 処理、権限付与、JSON 処理を含む）
- REST コンポーネントに基づいた BaaS（Backend as a Service）接続レイヤー（Parse や Kinvey などのサービスへの即時接続を提供、App42 サポートも別途利用可能）

多層開発に関しては、RAD Studio では、従来から DataSnap アーキテクチャを提供してきました。これは、もともと COM ベースのソリューションでしたが、その後 TCP/IP、HTTP、HTTPS、REST に移行しており、モバイルアプリケーションからのデータベースアクセスに適合しています。DataSnap は、カスタムサービスをプログラミングするための低レベル SDK と見なすことができます。この SDK は、多層開発向けの基盤を提供するもので、開発者は任意の多層ソリューションを構築できます。

RAD Studio 10 Seattle では、Microsoft Azure と Amazon AWS クラウドアクセスライブラリをアップデートし、System.Net ネイティブ HTTP / HTTPS をサポートするように変更しました（OpenSSL ライブラリの配布が不要になります）。

さらに、10 Seattle では、AWS の S3、Queue Service、SimpleDB API の AWS サービス（AWS Signature バージョン 4 を含む）の最新バージョンをサポートするようにアップデートされました。Azure についても、Blob、Table、Queue Services の最新バージョンをサポートしています。

## 6. RTL ライブラリのコア拡張

RAD Studio のランタイムライブラリは、Bluetooth サポート、新しいパラレルプログラミングライブラリの導入など、開発者コミュニティに大変好意的に受け入れられた機能により、XE7 で非常に大きな発展がありました。パラレルプログラミングライブラリは、並列 For ループ、タスク、フューチャー、Object Pascal の無名メソッドなどのいくつかの言語機能をベースとしています。

デスクトップ PC のみならず、モバイルデバイスにも広がるマルチコア CPU の機能を活かせる RAD Studio のネイティブコンパイラの威力は、極めて重要です。パラレルプログラミングライブラリは、最新の CPU の能力を有効活用するためのシンプルかつ抽象的な方法を提供します。これは、既存の VCL アプリケーションをリファクタリングする場合、新たにマルチデバイスモバイルアプリケーションを構築する場合の双方に当てはまります。

### XE8 で追加された新しい HTTP(S) クライアントライブラリ

最近のバージョンで追加された数多くのコア RTL 拡張の中でも、大変重要な機能強化が XE8 で行われました。新たに提供された HTTP / HTTPS クライアントライブラリは、すでに製品で提供していた REST クライアントライブラリや BaaS のサポートのベースとしても利用されているほか、RAD Studio 10 Seattle でも HTTP 接続を必要とする他のサブシステム（クラウドサポートや DataSnap HTTP 接続）にも適用されました。

新しい HTTP クライアントライブラリの重要性は、従来、RAD Studio で HTTP クライアントを構築するには、クロスプラットフォームの Indy ライブラリを活用してきたという事実に関連します。このライブラリはほぼ完全で、数多くのインターネットプロトコルをサポートしていますが、ある特定の問題、すなわち HTTPS サポートに関連した問題に直面しています。つまり、Indy では、SSL（Secure Socket Layer）をサポートするために、いくつかのターゲットプラットフォームにはプリインストールもされている OpenSSL ライブラリを利用していることです。

米国の輸出規制で、このライブラリの再配布を制限していることや、昨年いくつかのバグがこのライブラリ実装に発見されたことなどから、エンバカデロの開発者は、カスタムソリューションを配布するよ

り、プラットフォーム固有の HTTPS サポートに依存するほうがよい選択であると感じるようになりました。その結果、XE8 以降では、プラットフォーム固有の HTTP クライアントライブラリと、OS の SSL 層を使用するようにマップされた新しい NetHTTPClient および NetHttpRequest コンポーネントを使用して、HTTP および HTTPS 呼び出しを実装できるようになりました。このことは、開発者が（適切な SSL ライブラリを用意して提供しなければならないという）ハードルを取り除くだけでなく、プラットフォーム共通のライブラリの利用により、特定の SSL ライブラリを提供することによって発生する開発者側の責任も回避できるというメリットもあります。

## 10 Seattle で追加された JSON / BSON ストリーミング

RAD Studio 10 Seattle での RTL に関する新機能は、JSON 処理に関するものです。今回、JSON サポートは、広く利用されている JSON.NET ライブラリを使った実装に置き換えました。RAD Studio の実装では、.NET ライブラリの完全なポーティングに加え、Object Pascal と C++ で使用するための拡張が加えられています。RAD Studio には、従来の JSON 処理クラスも残されていますが、多くの場合、この新しいアーキテクチャに切り替えて、JSON ストリームの読み書きを行ったほうが、多くのメリットがあります。さらに、このライブラリでは 2 つの異なる形式、JSON と BSON（Binary JSON）をサポートしており、MongoDB データ型のための拡張も加えられています。

新しいライブラリでは、DOM モデルではなくストリーミングモデルを用いています。JSON データ型をメモリ上にオブジェクトのセットとして提供して読み書きを行うのではなく、ストリーミングのアプローチをとることで、データを直接処理し、より高速なパフォーマンス、メモリ使用量の削減、コーディングの柔軟性を実現できます。

これを実現するためのいくつかの内部クラスが、ライブラリに用意されています。

- 抽象クラス TJsonReader および TJsonWriter
- TJsonTextReader および TJsonTextWriter クラスによる JSON テキスト処理
- TBsonReader および TBsonWriter クラスによる BSON 処理
- TJSONArrayBuilder および TJSONObjectBuilder クラスを含む有用なメソッドビルダー
- JSON と BSON の前方移動のイテレーションを高速化する TJSONIterator クラス

この新しい JSON.NET ライブラリは、MongoDB データ処理の基盤としても利用されているほか、RAD Studio が提供するライブラリの多くのサブシステムで利用するように変更が加えられています。最終的には、従来の JSON 処理コードをすべてこの新しいライブラリに置き換えることを計画しています。

## 7. 追加のコンポーネント／ツール

エンバカデロでは、RAD Studio 開発者向けにさまざまな追加のコンポーネントやツールを提供しています（これらのうちのいくつかの製品は他の IDE でも利用できます）。いくつかの製品については、RAD Studio に既に搭載されていて、実行時（配布時）にライセンスの購入が必要になります。詳細については、各製品のページをご確認ください。

以下のセクションでは、それぞれのコンポーネント／ツールについての概要を紹介します。

### AppAnalytics

AppAnalytics は、エンバカデロが提供するバックエンドホスティングサービスです。AppAnalytics は、RAD Studio によって構築したアプリケーションを、ユーザーがどのように利用しているかを可視化する機能を開発者向けに提供します。RAD Studio ユーザーは、SaaS（software-as-a-service）ベースの Web インターフェイスを用いて、ユーザーの利用状況を収集／分析できます。

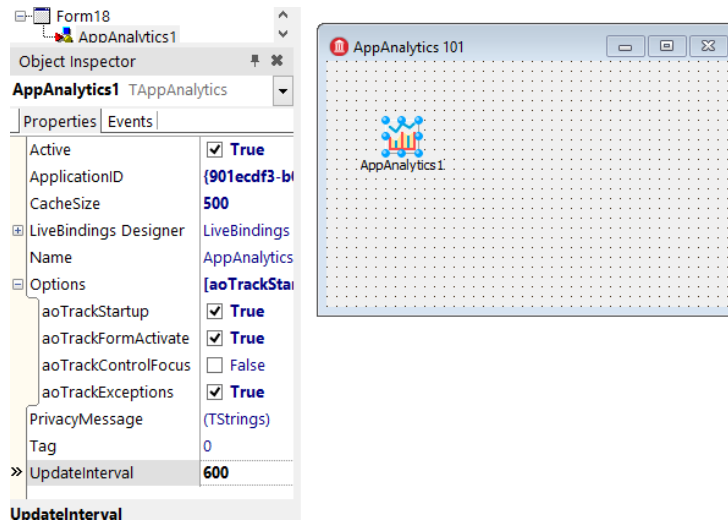
アプリケーションがどのくらいの回数起動、終了されたか？どの OS のどのバージョンを使っているか？どのフォームを表示しているか？これらは時間経過とともにどのように変化するか？こういった情報は、製品担当マネージャやマーケティングマネージャだけでなく、どの機能の開発に注力すればよいかといったことを判断する開発者にとっても有益です。企業環境では、社内のユーザーにどの程度アプリケーションが利用されているかをトラッキングするのも役立ちます。

アプリケーションのアナリティクス情報は、完全に匿名で収集されます。「ユーザーセッション」を使用すれば、個々のユーザーの行動を追跡することができますが、これはオプションで、その場合でもランダムに生成された GUID に基づいています。いずれの場合も、アプリケーションが利用状況のデータを収集するには、ユーザーが許諾することが必須要件になっています。

RAD Studio の他の多くの機能と同様、AppAnalytics はすべてのサポートプラットフォーム、モバイルでもデスクトップでも、FireMonkey でも VCL でも利用することができます。既存のアプリケーションにアナリティクス機能を追加するには、コンポーネントをひとつ配置して、簡単な設定を行い、再コンパイルして配布するだけです。開発者は、特定のメソッドの実行トレースやある UI 要素の操作などをトラッキングするために、カスタムイベントを用いて収集する情報を拡張することもできます。

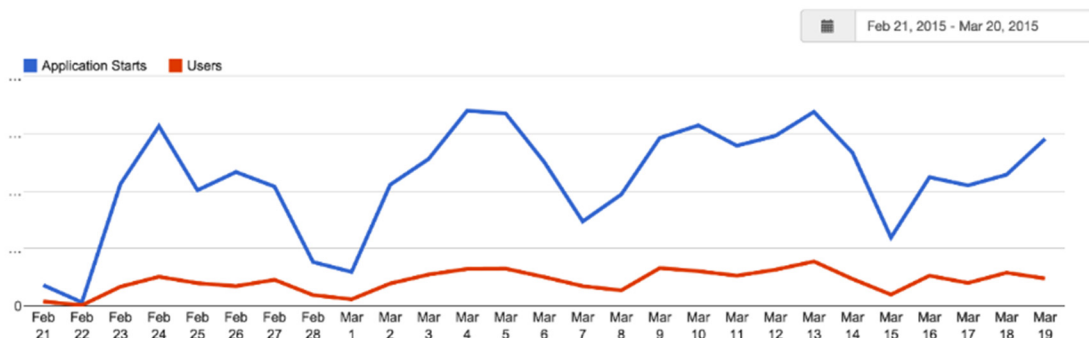
最もシンプルなケースは、新規にアカウントを作成し、既存の VCL アプリケーションにアナリティクスサポートを追加する例です。はじめに、<http://appanalytics.embarcadero.com> で、AppAnalytics アカウントを作成します（アカウントは無料で作成できます）。次に、このサイトで、アプリケーションを定義し名前を付けます。システムが生成したアプリ ID（GUID）をコピーします。VCL（または FireMonkey）アプリケーションを開き、メインページに AppAnalytics コンポーネントを配置します。先ほどコピーし

たアプリ ID を ApplicationID プロパティに設定します（以下の画面ショット参照）。コンパイルして実行します。



AppAnalytics の基本機能を用いるのに、このほかにコードを記述したり追加の情報を設定する必要はありません。これだけで高度な分析も可能です。デフォルトでは、アプリケーションの起動、フォームのアクティブ化、例外の発生をトラッキングします（上記の Options プロパティを参照）。コントロールのトラッキングを追加することもできますが、初期設定は無効になっています。

アプリケーションプロジェクトを開き、AppAnalytics コンポーネントを配置し、ApplicationID を設定。プログラムを実行すれば（この例では 3 回実行）、すぐに利用状況の収集が始まります。以下は、一定期間、実際のユーザーがアプリケーションを利用したデータの例です。



さらに、カスタムロギングにより、トラッキングする情報を拡張することができます。例えば、ボタンの OnClick イベントのイベントハンドラに、以下のコードを追加します。

```
AppAnalytics1.TrackEvent('Buttons', 'Button1');
```

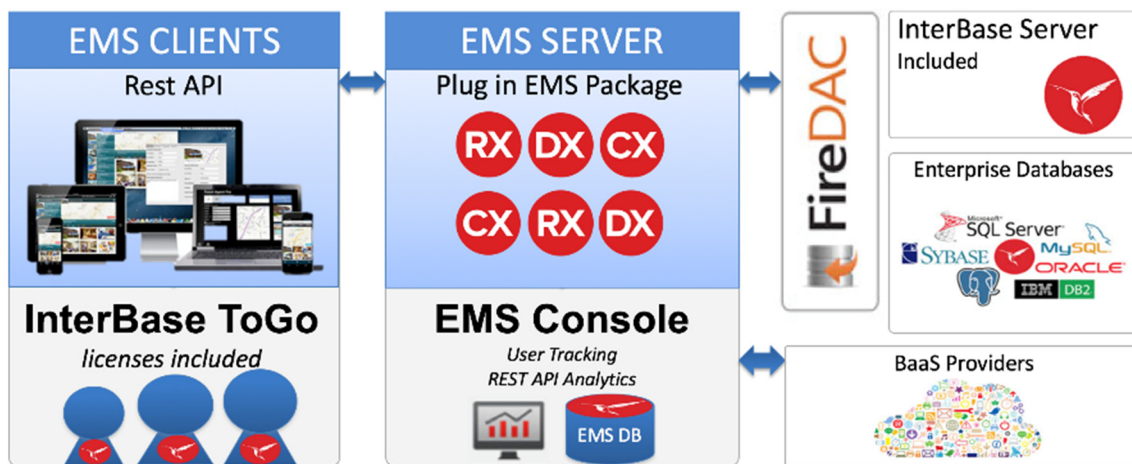
カスタムデータの収集もすぐに始まり、AppAnalytics サイトで情報を見ることができます（この機能は無料プランには含まれません）。

AppAnalytics コンポーネントは製品に含まれていますが、より高度な分析を行うには有料の AppAnalytics サービスを購入する必要があります。RAD Studio ユーザーは、自動的にアプリケーション開始などのイベントをトラッキングできる無料プランにサインアップしています。カスタムイベントやより高度な分析、フロー分析などの機能を利用するには、有料プランにアップグレードする必要があります。

## EMS (Enterprise Mobility Services)

RAD Studio XE7 から新たに導入された EMS (Enterprise Mobility Services) は、REST ベースの MEAP (Mobile Enterprise Application Platform) ミドルウェアスタックで、すぐに利用できるという特徴があります。サーバーはあらかじめコンパイルされており、基本的なユーザー管理サービスと認証サービスが組み込まれています。また、広範な REST API 呼び出しの解析機能により、ユーザーおよびユーザーとシステム間のやり取りを追跡するための Web ベースの REST コンソールが付属しています。

次の図は、EMS ソリューションの概要を示しています。バックエンドでは、既存のエンタープライズデータベース、EMS に組み込まれている InterBase サーバー、任意の BaaS プロバイダクラウドのいずれかを使用できます。中間層では、EMS は次の 3 つのモジュールで構成されています。



- EMS サーバーそのもの（開発者により作成されたカスタムパッケージを読み込んで、REST API やデータベースアクセスのホストになります）
- 別個の EMS コンソール Web サーバー（分析機能を提供します）
- InterBase を使用する EMS データベース（ユーザー構成と分析機能のログ データをホストし、EMS サーバーと EMS コンソールの双方から使用されます）

クライアント側には、任意のカスタム EMS サーバー REST API を呼び出すことができるアプリケーションがあり、暗号化されたデータベースがデバイス（またはデスクトップ）上に含まれています。

何から何まで開発者が作成するようなカスタムサーバーとは異なり、EMS には、ソリューションとすぐに使える状態のインフラストラクチャが用意されているため、開発と構成にかかる時間を短縮できます。サーバーでは、ユーザー管理、接続されているデバイスの追跡、API の使用状況の監視をサポートしています。HTTP ベースの別個のコンソールを使用すれば、何が行われているのかをリアルタイムに追跡できます。さらにクライアントコンソールでは、グループ、ユーザー、プッシュ通知（Apple または Google の双方のインフラを利用）の管理が可能です。

こうした環境を利用して、サーバー側の開発者は、Object Pascal か C++ のどちらかの言語を使ってパッケージを作成し、（位置または URI が一致する）特定のリソースを登録し、対応する要求を処理するコードを書くだけです。単純なシナリオでは、カスタム Web サービス API を実装するコードを書くだけで済みます。それに加えて、社内のリレーショナルデータベース内のデータにアクセスしてモバイルユーザーに返すための使いやすいコンポーネントも用意されています。

もう1つの重要な機能は、ユーザー、ユーザーのデバイス、ユーザーから呼び出されている API を追跡するための広範囲にわたるリアルタイム分析機能です。API の全体的な使用状況を確認し、個々のリソースについて詳しく調べ、個々の HTTP 呼び出しさえ参照できるので、ユーザーである顧客がアプリケーションをどう使用しているか、また、どのサービスやデータソースを主に扱っているかを追跡できます。

大半の企業がそうであるように、セキュリティを重視する場合には、その解決策も用意されています。EMS は、Windows Server OS で動作する Microsoft IIS に ISAPI DLL としてインストールされます。IIS を HTTPS サービスとしてインストールし、クライアント上で HTTPS サポートを有効にするだけで、使用するデバイスからサーバーへの通信はすべて安全に行われます。ファイアウォールを有効にし、サービスとデータベースサーバー間の接続のセキュリティを確保したら、準備完了です。

それでも、データをデバイス上にキャッシュし格納して、接続解除時でもアプリケーションを使用可能にしたり、帯域幅が低い状況でのパフォーマンス向上などを考えている場合は、デバイス上のデータのセキュリティも確保しなければなりません。InterBase ToGo Edition は、ローカルのモバイル InterBase エンジンですが、ここで述べたセキュリティ機能も提供しています。EMS では、InterBase ToGo もソリューションの一部になっています。

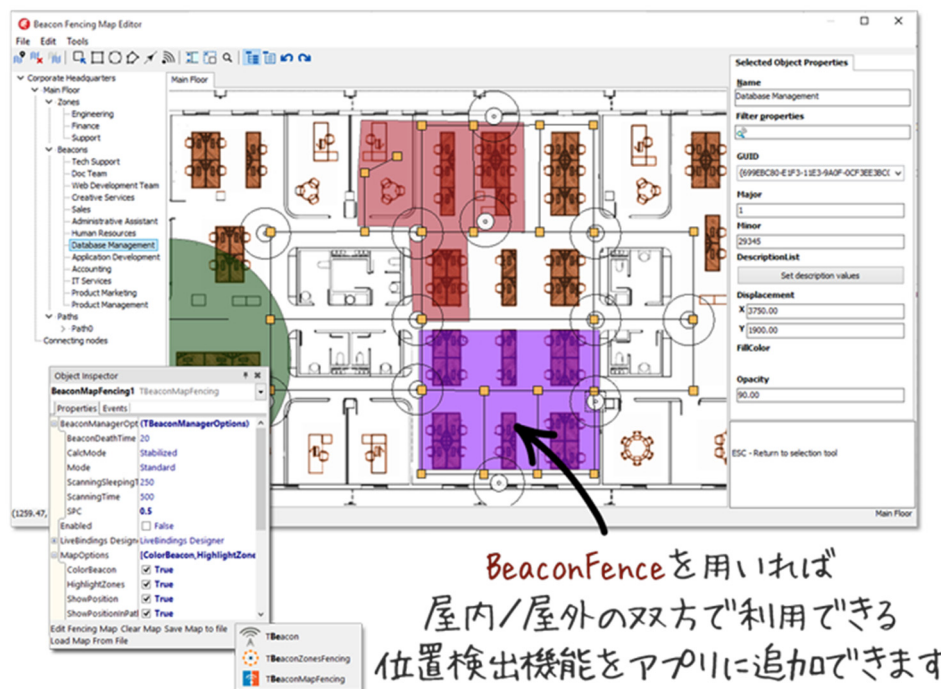
EMS は MEAP（Mobile Enterprise Application Platform）カテゴリに分類され、社内かカスタムクラウドのどちらかにホストされるようになっています。社内ホスティングの場合は、データベースに直接アクセスし、社内ファイアウォールで完全に保護されます。一方、カスタムクラウドホスティングの場合は、

一般にスケーラビリティが向上し帯域幅が増えることとなります。インストール可能なサーバーが提供されるため、料金の支払いが必要でかつパフォーマンスを制御できない専有のカスタムホスティングソリューションを使用しなければならない理由はありません。ホスティングを変更する必要があるときはいつでも、何の問題もなく、ソリューションの構成を自由に変更できます。

EMS は、中間層のセキュアソリューションでの API ホスティングとデータソースアクセス機能を提供します。また、パッケージの一部として、バックエンドデータを格納するための RDBMS と、モバイルデバイスにインストールするための暗号化をフルサポートしたローカル RDBMS が含まれています。柔軟な運用ライセンスオプションを用意しており、使用しているデバイスに関係なく、エンタープライズモビリティサービスにアクセスするユーザーの数に基づいた運用ライセンスを購入できます。

## BeaconFence

BeaconFence は、近接検知を利用するための開発者向けのソリューションです。BeaconFence の活用により、屋内（および屋外）で GPS が使えない状況でも、精密なユーザーの位置追跡が可能になります。精密な位置情報の追跡のために、ビーコンの配置レイアウトをビジュアルに管理。ゾーンを作成して、その中に入ったり、出たりした際には、コールバックイベントを発生させることができます。BeaconFence を用いれば、高精度の空間位置認識が可能となり、ビーコンによる近接検知のより高度な活用が実現できます。

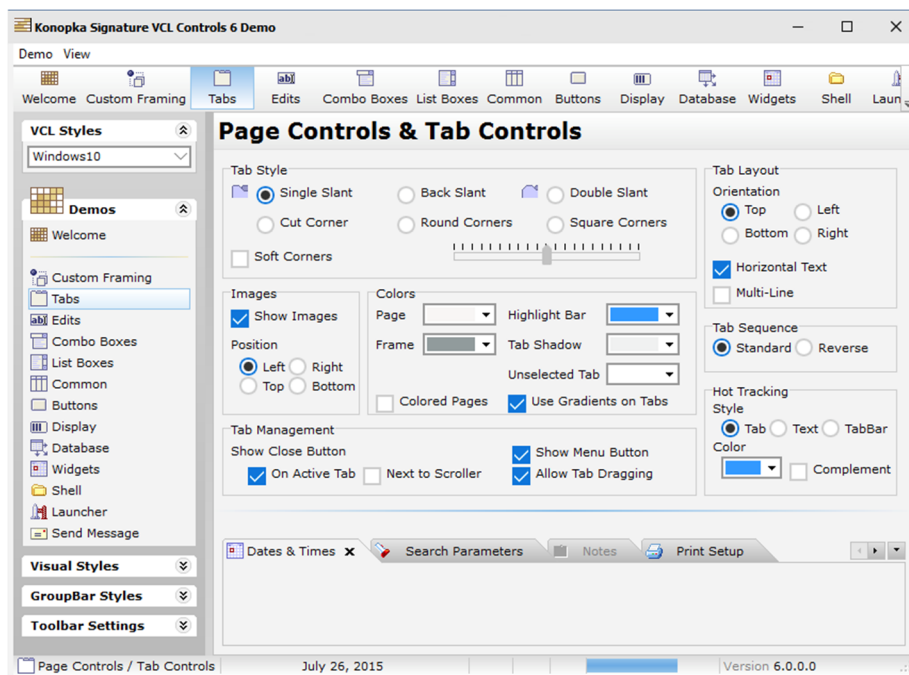


BeaconFence のコンポーネントには、いくつかのプロパティ、イベントが用意されています。例えば、OnZoneEnter イベントは、ユーザーが定義したゾーンに入った場合に発生するものです。これを用いれば、容易にユーザーの移動に対するリアクションコードを記述できます。

BeaconFence には、Beacon Fencing Map Editor が搭載されています。このビジュアルエディタを用いれば、新しいマップを作成し、オフィスフロアのレイアウト画像を読み込み、BeaconFence で利用するマップとして使用することも可能です。工場、オフィス、ショッピングモールなどで、複数階に対して異なるマップを作成し、ビーコンの配置やゾーンを定義することも可能です。

## Konopka's Signature VCL Controls

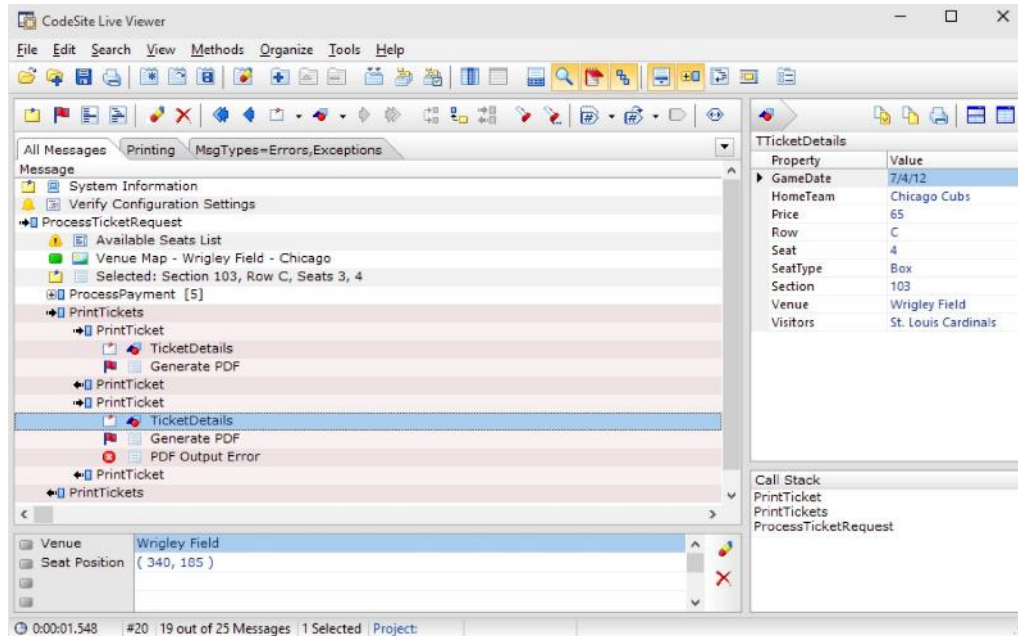
Konopka Signature VCL Controls を用いれば、Windows アプリケーション向けに、最新の洗練されたユーザーインターフェイスを素早く設計できます。125 以上の Windows UI コントロールは、より直感的でビジュアル的にも美しくなるように設計されており、Delphi および C++Builder 開発者は、すべての Windows アプリケーションの UX 構築において、現代化と簡素化を実現できます。



## CodeSite Studio

ローカルおよびリモートで実行されているコード内部の実行状況を、リアルタイムに確認できるログ取得システムがあれば、問題箇所を素早く把握できるようになります。CodeSite を用いれば、ブレークポイントの設置、デバッグといった従来の手法から、アプリケーションの実行ログをリアルタイムトレ

スする方法に移行でき、複数のアプリケーションが協調する複雑なシステムにおける問題箇所の検出も容易になり、開発効率を格段に高めることができます。



## RAD STUDIO 10 Seattle 追加情報

RAD Studio 10 Seattle は、Delphi 10 Seattle、C++Builder 10 Seattle、HTML5 Builder、InterBase を含む開発ツールスイート製品です。

- RAD Studio 10 Seattle 製品カタログ  
<http://www.embarcadero.com/images/jp/dm/datasheet/rad-studio-10-seattle-datasheet-jp.pdf>
- Delphi 10 Seattle 製品カタログ  
<http://www.embarcadero.com/images/jp/dm/datasheet/delphi-10-seattle-datasheet-jp.pdf>
- C++Builder 10 Seattle 製品カタログ  
<http://www.embarcadero.com/images/jp/dm/datasheet/cbuilder-10-seattle-datasheet-jp.pdf>

RAD Studio 10 Seattle の画面ショット集は以下のページから入手できます。

<http://www.embarcadero.com/jp/products/rad-studio/product-tour>

## RAD Studio 10 Seattle 製品ラインナップ

RAD Studio には、Professional、Enterprise、Ultimate、Architect の 4 つのエディションがあります。

| 機能                                   | Architect                                                                                                               | Ultimate                            | Enterprise | Professional                                      |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------|------------|---------------------------------------------------|
| マルチデバイス開発                            | Android、iOS、Windows、Mac のネイティブ開発（Delphi および C++Builder）<br>Web アプリケーションとモバイル Web アプリ開発（HTML5 Builder）                   |                                     |            |                                                   |
| DB アプリケーション<br>アーキテクチャ               | クライアント/サーバー、多層、ローカル/組み込み                                                                                                |                                     |            | ローカル/組み込み                                         |
| データベースおよび<br>クラウドサポート                | InterBase、SQL Server、Oracle、DB2、Sybase、MySQL、<br>ODBC など、Amazon / Azure クラウド、DataSnap 多層アクセ<br>ス、FireDAC データアクセスコンポーネント |                                     |            | InterBase、MySQL、<br>SQLite、Amazon /<br>Azure クラウド |
| IDE ツール                              | リファクタリング、ユニットテスト、UML モデリングのフル機<br>能を備えた先進的な IDE                                                                         |                                     |            | リファクタリング、ユ<br>ニットテスト、UML に<br>よる可視化機能を備え<br>た IDE |
| 高度なデータベース<br>モデリングツール<br>および SQL ツール | ER/Studio<br>Developer Edition                                                                                          | DB PowerStudio<br>Developer Edition | 含まれていない    |                                                   |
| 旧バージョンへの<br>アクセス                     | 旧バージョン（Delphi/C++Builder 2007 ～ XE8、Delphi 7、C++Builder 6、RadPHP XE2、<br>RadPHP XE）のライセンス取得とダウンロードが可能。                  |                                     |            |                                                   |
| 含まれている<br>IDE パーソナリティ                | Delphi、C++Builder、HTML5 Builder                                                                                         |                                     |            |                                                   |

## **RAD Studio 10 Seattle Professional**

RAD Studio 10 Seattle Professional は、Windows、Mac、iOS、Android のマルチデバイス開発機能、ローカルデータベースアクセス機能を利用できるほか、アドオン製品を追加することで、FireDAC によるエンタープライズデータベースアクセス機能を Delphi/C++ で利用することができます。

## **FireDAC Client/Server Add-On Pack for RAD Studio 10 Seattle Professional**

FireDAC Client/Server Add-On Pack は、RAD Studio 10 Seattle Professional で FireDAC を利用可能にするアドオン製品です。Oracle、SQL Server、InterBase、DB2、SQLite、MySQL、PostgreSQL、SQL Anywhere、Advantage DB、Firebird、Access、Informix、Teradata、MongoDB、DataSnap など広範なデータベースアクセスが可能になるほか、EMS を利用した開発も可能です。

## **RAD Studio 10 Seattle Enterprise**

RAD Studio 10 Seattle Enterprise は、Windows、Mac、iOS、Android のマルチデバイス開発機能を利用できるほか、主要な RDBMS に対応したマルチデバイス対応のハイパフォーマンスデータアクセスをサポート。EMS（Enterprise Mobility Services）ミドルウェアサーバーを利用した開発が可能です。

## **RAD Studio 10 Seattle Ultimate**

RAD Studio 10 Seattle Ultimate は、Windows、Mac、iOS、Android のマルチデバイス開発機能を利用できるほか、主要な RDBMS に対応したハイパフォーマンスデータアクセスをサポート。EMS（Enterprise Mobility Services）ミドルウェアサーバーを利用した開発が可能です。さらに DB PowerStudio Developer Edition も搭載。SQL 開発／チューニングをサポートします。

## **RAD Studio 10 Seattle Architect**

RAD Studio 10 Seattle Architect は、Windows、Mac、iOS、Android のマルチデバイス開発機能を利用できるほか、主要な RDBMS に対応したハイパフォーマンスデータアクセスをサポート。EMS（Enterprise Mobility Services）ミドルウェアサーバーを利用した開発が可能です。さらに ER/Studio Developer Edition も搭載。データモデリングを強力にバックアップします。

エンバカデロ・テクノロジーズは、1993 年にデータベースツールベンダーとして設立され、2008 年にボーランドの開発ツール部門「CodeGear」との合併によって、アプリケーション開発者とデータベース技術者が多様な環境でソフトウェアアプリケーションを設計、構築、実行するためのツールを提供する最大規模の独立系ツールベンダーとなりました。米国企業の総収入ランキング「フォーチュン 100」のうち 90 以上の企業と、世界で 300 万以上のユーザーが、エンバカデロの Delphi®、C++Builder®といった CodeGear™製品や ER/Studio®, DBArtisan®, RapidSQL®をはじめとする DatabaseGear™製品を採用し、生産性の向上と革新的なソフトウェア開発を実現しています。エンバカデロ・テクノロジーズは、サンフランシスコに本社を置き、世界各国に支社を展開しています。詳細は、[www.embarcadero.com/jp](http://www.embarcadero.com/jp) をご覧ください。

Embarcadero、Embarcadero Technologies ロゴならびにすべてのエンバカデロ・テクノロジーズ製品またはサービス名は、Embarcadero Technologies, Inc.の商標または登録商標です。その他の商標はその所有者に帰属します。