



RAD Studio 2010 機能評価ガイド

2009年12月

はじめに

Embarcadero RAD Studio 2010 を評価いただき、誠にありがとうございます。このガイドは、RAD Studio 2010 を評価する際に、新機能を中心にその概要を理解いただけるように記述されています。このガイドは、3部構成となっています。第1部では、RAD Studio 2010の概要、基本的な製品の特徴について説明しています。第2部では、RAD Studio 2010の主要な機能について説明し、最新バージョンではどのような点が強化されているのかを紹介しています(いわば、この部分が、バージョンアップで得られるメリットの説明になります)。第3部では、さらに詳細な製品の拡張された機能セットの紹介を行っています。

RAD STUDIO 2010 の概要

CodeGear RAD Studio 2010 は、広範な用途に使用できる RAD (Rapid Application Development) Windows アプリケーション開発ツールです。RAD Studio は、3 つの異なる製品、Delphi 2010、 C++Builder 2010、Delphi Prism 2010 から構成されています。RAD Studio の Delphi および C++Builder は、x86 OS で実行可能なネイティブ Win32 バイナリを生成します。Delphi Prism は、.NET アプリケーションとクロスプラットフォームの Mono アプリケーションを生成します。 RAD Studio 2010 は、Delphi 言語 (Object Pascal) と C++言語に加え、Object Pascal ベースの.NET 向け言語 Delphi Prism (旧 Oxygene) をサポートしています。

RAD Studio 2010 を用いれば、開発者は、スタンドアロン実行形式(EXE)、ダイナミックリンクラ イブラリ(DLL)、OCX や COM オブジェクト、タイプライブラリ、コントロールパネルアプレット、 Windows Server アプリケーション、コンソールアプリケーションなど任意の Windows バイナリ、さ らにはフル.NET サポートアプリケーションを作成できます。開発者は、高度で複雑なユーザーイン ターフェイスを持つクライアントアプリケーションも、シンプルなコマンドラインアプリケーション のいずれも作成可能です。主要な RDBMS(リレーショナルデータベースマネージメントシステム) に直接接続するデータベースクライアントアプリケーションや、中間層のアプリケーションサーバー、 Web アプリケーション、Web サイト、Web サービス、ActiveX コントロール、複雑な組み込みシス テムで稼動するマルチスレッドアプリケーションなども作成できます。つまり、RAD Studio 2010 は、 Windows アプリケーション開発のあらゆる需要に対応する開発ツールです。

RAD STUDIO 2010 の新機能

RAD Studio 2010 の主な新機能は以下のとおりです。

開発期間の短縮

コーディング作業を効率化する新機能により、RAD Studio では、さらなる開発生産性の向上を実現。 強力な IDE によって、開発時間を短縮し、より生産性を高めることができます。

- IDE インサイト シンプルなキー操作と検索方法で、ファイル、コンポーネント、機能、設定な どをすばやく検索可能。
- コード整形 最小の手間で、コーディングスタイルを統一できます。
- C++クラスエクスプローラ プロジェクトで使用するすべてのクラスライブラリの階層ビューに よって、宣言と実装部へのすばやいナビゲーションが可能。
- デバッガビジュアライザ 定義可能な形式で任意のデータをビジュアル表示でき、デバッグ作業を容易にします。

- デバッガのスレッド制御 デバッグ中にアプリケーション内の各スレッドを停止、再開、分離 可能。スレッドに関連する問題をより早く分析できます。
- 新しいデバッガオプション ビューまで新しいイベントをスクロール、非ユーザーブレークポイントの無視
- エントリーレベルのコード検査・測定 Professional 版でもコード検査・測定のエントリーレベルの機能を利用可能。コードの状態を掌握できます。

キーボードレスアプリケーションを開発可能

柔軟なタッチ対応フレームワークにより、タブレット、タッチパット、POS 向けのアプリケーションをすばやく開発可能。わずかなコーディングだけで既存のアプリケーションもタッチ/ジェスチャに対応させることができます。

- プラグイン型のジェスチャーエンジンアーキテクチャ
- Windows 2000、XP、Vista、Windows 7 をシームレスにサポート
- タッチ対応ハードウェアに加え、ペン入力、マウス操作などさまざまな入力スタイルをサポート
- VCL をベースとしたタッチ/マルチタッチインターフェイスのサポート
- 30 以上の既定のジェスチャ、パン、ズーム、回転などのインタラクティブなマルチタッチ
- ジェスチャーデザイナによるカスタムジェスチャーのビジュアル設計
- タッチキーボード バーチャルタッチ、マルチロケール、マルチ言語のスクリーンキーボードによる入力サポート

データコネクティビティの強化

RAD Studio 2010 を用いれば、より多くの種類のデータ、Web サービス、アプリケーションアーキ テクチャに接続することができます。

- Firebird 2.1 と 1.5 を dbExpress で新たにサポート
- InterBase 2009、Microsoft SQL Server 2008、Oracle 11g、MySQL 5.1 など主要なデータベース ドライバをアップデート
- 新しい SOAP 1.2 クライアントサポートにより、Amazon や他の Web サービス機能を活用可能
- DataSnap Windows アプリケーションと.NET 多層アプリケーション向けの HTTP コネクション とインプロセスコネクティビティ
- オブジェクトギャラリーに追加された新しい DataSnap ウィザードにより容易にサーバーを作成 可能
- サーバーのクライアント通信を可能にする DataSnap コールバックを利用可能
- DataSnap サーバーとクライアント間のデータストリームをフィルタリングによって完全にコントロール可能
- DataSnap では、COM、XML、SOAP に加えて、RESTful Web Services と JSON をサポート

より多くのユーザーデスクトップに対応

特定の Windows API の詳細を気にすることなく、より多くの Windows バージョンに対応可能。 Unicode サポートと強化されたローカライゼーション機能により、アプリケーションの国際化、多言 語化にも迅速に対応できます。

- VCL コントロールが XP、Vista、Windows 7 の機能/テーマを活用して最適化
- グローバルデータ/ユーザーに対応するフル Unicode サポート
- Windows XP、Vista 向けのタッチ/マウスジェスチャー対応アプリケーションを開発可能。
 Windows 7 の新しいタッチサポートも利用可能
- Delphi IDE のプラグイン開発のための Open Tools API を拡張
- 英語、ドイツ語、フランス語、日本語対応の多言語対応 IDE メニュー、ダイアログ、コンパイ ル済ユニット、リソース、ソースコード
- IDEメニュー、ダイアログなどの言語を容易に切り替え可能

言語の新機能

Delphi 2010 には、新たに RTTI サポートが搭載されているほか、さまざまな言語の機能強化があります。

- 動的呼び出しと他のメタプログラミングアプローチを可能するメソッド、フィールド、プロパティ公開のための RTTI サポート
- オブジェクト指向ファイルとディレクトリ IO クラス
- 型、フィールド、プロパティ、メソッド、パラメータといった多くのコード要素のためのカスタム属性のサポート
- 文字列の結合を簡単かつ高速に行う TStringBuilder の強化
- ジェネリクスリストおよびコレクションの RTL サポートの強化
- ローカライズリソースのサポート強化
- バックグラウンドスレッドでのコンパイル

C++Builder 2010 では、C++0x の言語機能の早期サポートのほか、いくつかのコンパイラおよびラ イブラリの機能強化があります:

- FastMM を C++ランタイムライブラリの標準ヒープマネージャとして採用
- #pragma once のサポート
- ソースコードの XML 表現出力のための -Zx オプション
- _FUNCTION_のサポート
- [[deprecated]] 属性のサポート
- テンプレートクラスのための __declspec(dllimport) および __declspec(dllexport)
- Boost ライブラリ 1.39
- 標準 C++ヒープマネージャの強化

- 最適化された文字列/メモリ処理関数
- バックグラウンドスレッドでのコンパイル

RAD Studio 2010 Architect でのモデリング機能の強化

RAD Studio 2010 Architect には、新しい ER/Studio 8.0 Developer Edition (英語版) が含まれており、 データの掌握、ドキュメント化、再利用を促進する強力なリバースエンジニアリング、分析、最適化 ツールを利用できます。ER/Studio の新機能は以下のとおりです。

- ビジュアルデータリネージ コードを調べることなく、組織のデータフローの状態をビジュア ル分析しドキュメント化
- 属性レベルサブモデリング サブモデルのエンティティ/テーブルに含まれる属性/列を選択し、新しい [Definition] タブでサブモデルを描画。そして、その定義での検索のためのクエリーを生成
- オブジェクトレベルの比較フラグ 比較ウィザードが無視するモデルの比較で、意図的に矛盾 を指定可能
- HTML 形式のレポートの出力
- Microsoft SQL Server 2008 のサポート

Delphi Prism の先進の言語機能

Delphi Prism では、Delphi 開発者と.NET 開発者が、.NET アプリケーションをよりよく開発できる言 語機能を提供しています。Delphi 開発者は、なじみのある言語構文を利用でき、.NET 開発者は、他 の.NET プログラミング言語ではまだ利用できていない先進の言語機能を使用できます。

Delphi Prism には、以下のような新しい言語機能が搭載されています。

- Delphi 言語との互換性強化
- AOP (Aspect Oriented Programming)
- ダイナミックタイプのサポート(.NET 4.0)

Delphi PrismのAOPサポートにより、クラスへのアスペクトと呼ばれる特殊な属性によって、コードの動作を変更したり、フィールド、プロパティ、イベント、メソッドなどを追加/削除できます。

アスペクトは、Prism で記述し、別個のライブラリにコンパイルされるので、異なるプロジェクトで 再利用可能です。

アスペクト属性は、コンパイラによってコンパイル時にロードされインスタント化されます。これに より、コンパイラが生成した強力な機能をコード上で利用することができます。

Delphi Prism の主なコンパイラ機能

- RemObjects Cirrus: AOP for Oxygene
- RemObjects Cirrus 向けの標準アスペクトライブラリ (プレビューリリース)

- Unmanaged Exports
- Generic Type Variance
- Volatile fields
- CLSCompliantAttribute サポートとコンパイラ警告
- 新しい LINQ クエリー式演算子 Skip, While, Take and Take While

Delphi Prism のその他のコンパイラ機能

- Unquote 式サポート(Cirrus を主にサポート)
- \$DELPHICOMPATIBILITY コンパイラ制御とプロジェクトオプション
- ランタイムレンジチェック
- Range 列挙型("type Ten = 1..10;")
- リードオンリークラス
- Array 型における negative low bounds のサポート

Delphi Prism の他の新機能

- UserData における OxygeneInterface、OxygenePartial、OxygeneEmpty タグサポートのための CodeDom の強化
- ビルド前/後イベント
- デバッグオプション リモートマシン
- Mono アセンブリリファレンス追加のためのカスタム「リファレンスの追加」ペイン
- Delphi Prism とともに提供される新しい Monobjc テンプレートと Monobjc ライブラリ
- Mono 2.4 インストールのためのセットアップのアップデート
- インターネットパックの追加
- wiki ドキュメントのオフラインコピーをベースとした F1 ヘルプの統合
- ASP.NET Web プロジェクト向けテンプレートの追加

データ/アプリケーションとの接続性

Delphi Prism 2010 を用いれば、HTTP プロトコルを用いたサーバーとの通信を含む DataSnap 2010 の新機能を用いた、.NET DataSnap クライアントアプリケーションを開発できます。DataSnap の機能は、Delphi Prism Enterprise、Embarcadero RAD Studio Enterprise、Embarcadero RAD Studio Architect で利用できます。

DELPHI PRISM に関する補足

Delphi Prism 製品は .NET のみを対象にしているため、同製品については、この機能評価ガイドではなく別個のガイドで扱われる予定です。基礎となる IDE は、Eclipse IDE フレームワークに似た

Microsoft の Visual Studio Shell (VSS) プログラムをベースにしています。Delphi Prism で使用され るインターフェイスおよびアプローチは異なるので、RAD Studio のその部分については別個のドキ ュメントにするほうが良いと思われます。将来的には、このドキュメントに付け加えられる可能性が あります。Delphi 製品と C++Builder 製品はどちらも同じ内部 IDE フレームワークをベースに構築さ れ、もっぱらネイティブ アプリケーションを対象にしているため、このドキュメントではこの両製 品を中心に扱います。

前提条件

RAD Studio 2010 をインストールするには、以下の前提条件がインストールされていなければなりません。

- Microsoft .NET Framework 2.0
- Microsoft Direct Access Objects 2.8
- Microsoft Internet Explorer v6.0 SP1
- Microsoft XML Core Services (MSXML) v4.0 SP2
- Microsoft Visual J# .NET v2.0 Redistributable

これらがシステムにインストールされていないと、RAD Studio インストーラがインストールを実行 します。

.NET Framework は IDE によって使用されますが、Delphi/C++Builder によって作成されるアプリケーションはネイティブアプリケーションなので、.NET Framework には依存しません。

動作環境

RAD Studio 2010 を実行するには、以下の環境が必要です。

- Intel Pentium またはその互換機 1.4GHz 以上(2GHz 以上を推奨)
- 1GB RAM (2GB 以上を推奨)
- 3GB 以上のディスクスペース(Delphi インストール用)
- 750MB 以上のディスクスペース(前提条件用)
- DVD-ROM ドライブ
- 1024x768 以上の高解像度モニタ
- マウスなどのポインティングデバイス

また RAD Studio 2010 では、以下の OS へのインストールをサポートしています。

- Microsoft Windows XP Home or Professional (SP2 以上)
- Microsoft Windows Vista SP1 (管理者権限が必要)
- Microsoft Windows Server 2003 (SP1)
- Microsoft Windows Server 2008
- Microsoft Windows 7 (管理者権限が必要)

ノート: RAD Studio 2010の英語版は、バージョン 6.0 以前の Internet Explorer を 6.0 SP1 にアップ デートします。英語以外の OS を使っている場合には、正しいローカライズバージョンの Internet Explorer を確実に取得するために、Windows アップデートを実行してください。

国際化

RAD Studio 2010 は、以下の言語にフルローカライズされています。

- 英語
- フランス語
- ドイツ語
- 日本語

エディション

RAD Studio 2010 には、Professional、Enterprise、Architect という 3 つのエディションがあります。

PROFESSIONAL

RAD Studio 2010 Professional は、RAD 開発の基本的な機能に加え、ローカルデータベースアクセス 機能を必要とする開発者向けの製品です。Professional 版には、IDE、Delphi 言語、VCL ソースコー ドを含む VCL コンポーネントのフル機能が搭載されています。データベースアクセス機能について は、InterBase、MySQL、Blackfish SQLのローカルアクセスに限定された機能を利用できます。

Professional 版には、5 接続限定のスタンドアロンサーバーに限定された、VCL for the Web の機能限 定版も含まれます。さらに、リバースエンジニアリング可能なクラス図やコードの検査・測定のため の静的分析機能などの UML サポートの限定機能も含まれます。

Professional 版は、ISV、プロフェッショナル開発者など、リモートデータベースアクセス機能を必要としていない、Windows アプリケーション開発者向けの製品です。

ENTERPRISE

RAD Studio 2010 Enterprise は、RDBMS へのアクセスを必要とする開発者向けの製品です。 Enterprise 版は、以下の複数のデータベースへローカル/リモート接続可能です。

- Firebird 2.1 および 1.5
- InterBase 2007 および 2009
- Blackfish SQL for .NET/Java
- Oracle 10g および 11g
- Microsoft SQL Server 2000, 2005, 2008
- DB2 UDB 8.X
- MySQL 4.1, 5.0, 5.1
- Informix 9x

- Sybase Adaptive Server Enterprise 12.5
- Sybase SQL Anywhere 9

さらに、Enterprise版には、フル機能の VCL for the Web が搭載されており、接続数無制限のスタンドアロンおよび ISAPI/Apache ベースのアプリケーションを開発できます。

Enterprise 版には、UML(Unified Modeling Language)による高度なモデリング機能が搭載されてい ます。クラス図のリバースエンジニアリング(Delphi 開発者は、コードとモデル図の双方向開発を利 用できます)により、容易にプロジェクト全体を鳥瞰できるビューを獲得できます。作成した図を含 むドキュメントも簡単に自動生成できるので、プロジェクトの情報共有が容易になります。

DataSnap 技術は、分散コンピューティングの必要性が増すにつれ発展し続けています。DataSnap を支える技術は、Microsoft の COM/DCOM を通じたデータ リモーティングのアプローチを超えて、 TCP/IP に基づいたよりオープンな通信アプローチに移行しました。この発展により、DataSnap 技 術では、完全なミドルウェア技術を包含するように機能を拡張できました。この技術を支える主要な 特長の1つは速いこと、つまり、ビルドが速い、配置が速い、実稼働時の実行が速いということです。

Enterprise 版は、企業の情報システム部門、データベースへのアクセス機能を利用する ISV、VAR を はじめとするエンタープライズデベロッパー向けの製品です。

ARCHITECT

RAD Studio 2010 Architect には、Enterprise 版のすべての機能に加え、データモデリング/設計機能 を提供する Embarcadero ER/Studio Developer Edition が搭載されています。Embarcadero ER/Studio は、実績あるデータモデリングツールで、データ設計の品質向上、最適化、再利用、メン テナンスなどを支援します。ラウンドトリップデータベースサポートにより、データベースアーキテ クトは、容易に既存のデータベースをリバースエンジニアリングして、分析、最適化できます。 ER/Studio の強力なコラボレーション機能を用いれば、チームの生産性、品質の向上を強化できます。

Architect 版は、Enterprise 版と同様の機能を必要としている開発者で、データベースの設計や管理について高い生産性を必要としているユーザー向けの製品です。

インストール

RAD Studio 2010 のインストールは、DVD または Web ダウンロードのいずれかで行うことができま す。ダウンロードインストールは、小さいインストールラウンチャのみをダウンロードして実行しま す。また、登録ユーザーは、<u>http://cc.embarcadero.com/reg/rad_studio</u> から追加のインストール項目 をダウンロードできます。

DVD によるインストールは、従来どおり、コンピュータに DVD を挿入して、インストーラを実行し てインストールを行います。DVD は、ISO イメージをダウンロードしていただくか、DVD 付パッケ ージの購入ないしはメディアキットの追加購入にて入手可能です。

統合開発環境(IDE - INTEGRATED DEVELOPMENT ENVIRONMENT)

RAD Studio 2010 を実行すると、統合開発環境(IDE)が表示されます。IDE は、アプリケーション 開発に必要とされるすべての機能を単一環境で提供するツールです。エディタ、フォームデザイナ、 プロジェクトマネージャ、デバッガをはじめとするさまざまな機能を利用でき、アプリケーションを 迅速かつ容易に開発できます。

IDE について

IDE は、Windows ユーザーになじみのあるユーザーインターフェイスを提供します。基本機能は、 ドロップダウンメニューやカスタマイズ可能なツールバーからアクセスできます。IDE の多くのウィ ンドウはドッキング可能で、作業環境をフルカスタマイズできます。また、デスクトップのレイアウ トは保存することができます。レイアウトは、「デバッグ用」といった特定の目的用に設定できます。 作成したアプリケーションは、IDE 内で実行、デバッグできます。IDE では、その環境やアプリケー ション、プロジェクトなど、さまざまな設定を変更できます。IDE 全体として、カスタマイズ可能な 設計になっており、迅速かつ効率的な開発が可能になっています。

スピードが鍵 – 新しい IDE インサイト

RAD Studio の新しい IDE インサイト機能でパラダイム シフトが起ころうとしています。この機能に より、開発者は IDE 内のあらゆる要素をプロジェクト、コンポーネント、テンプレート、構成設定 などからすばやく見つけることができます。F6 キーを押すだけで IDE インサイトが起動されるので、 開発者は実行したい機能の名称などを入力していきます。該当する可能性のある項目の一覧が表示さ れます。開発者がその中から目的の機能を選択すると、それが実行されるか、対応するダイアログが 表示されます。IDE インサイトは IDE 全体で利用でき、いつでもキーを押すだけで機能します。

🐵 IDE Insight 🛛 🔀
Qrvd∣ 🛛
Commands (1)
🐻 New VCL Component
New Items (4)
🐔 VCL for the Web Application Wizard (Delphi) - VCL for the Web Application Wizarc
🖿 VCL Forms Application (Delphi)
🖳 DataSnap Server (Delphi) - Creates a new DataSnap Server Console, VCL Forms,
Preferences (14)
峛 VCL Designer page (Environment Options)
🏂 Auto create forms & data modules - Module creation options (Environment Option
🏂 Display grid - Grid options (Environment Options VCL Designer)
🏂 Embedded designer - Options (Environment Options VCL Designer)
🏂 New forms as text - Module creation options (Environment Options VCL Designe
⅔ Show component captions - Options (Environment Options VCL Designer)
5/3 Show designer hints - Options (Environment Options VCL Designer)
🏂 Show extended control hints - Options (Environment Options VCL Designer)
< · · · · · · · · · · · · · · · · · · ·
Show all OK Cancel

図 1 – IDE インサイト

キーボードによる操作を多用する開発者向けに、IDE インサイトは Ctrl + . (ピリオド)を押すことで起動することもできます。

•	F	01	111	18																														L			>	<
	11	11	1		1	1	1	1			÷,	ļ,	1	i.	1			1	į.	į.			ļ,	į.	1		ī,	ļ,	ļ,	į.			ī,	į,	ļ,	1		
																							÷	÷				÷							÷			
	1.1			• •					•										1	۰.			1	1				1		۰.				1		•		
												-																	-	•								
											1		1	1					1	1			1	1				1	1	1				1	1	1		
	11	11	1	11	1	1	1	1	1.1		1	1	1	1				1	1	1		1	1	1	1.1		1	1	2	1	1.1		1	1	1	1.		
	11	11																															1	1	1	1		
				le.	4	44																																
					u	u	£.,																															
				-	-					-						-																						
		• •		• •					• •						•																				•	•	• •	
									• •																											•	• •	
				• •																														1				
																																				1		
	11	1.1		11		1	1	÷.	1.1		1	1	1	1																			1	1	1	1		
	11					1	1	÷.	1		1		1	2	2			1																	1	1		
			÷.										÷.																					-				
				_	_	_	_	_	_		_																											
								-	- 1																										÷			
		•			-	su	Ju	.01			1																									•		
	• •	• •		-	_	_	_	_	_	_	-			•					-	_			_	_		_	_	_	_	_	-							
													1																							1		
														1					1	1				1									1					
	1.1	1.1	1	11			1	1	1		1	1	1	1				1	1	1			1	1			1	1	1	1	1			1	1	1		
				. 1	1	1						1						1			. 1		1				1	1						1	1			
																												1										
					۰.																																	
																									•													
		• •		• •	•				• •	•				•	•		•		•	•				۰.		• •				۰.	•	• •				•	• •	•
	1.1	• •		• •			1		• •				1						1	•			1	1	•			1		1	•			1				
																							1	1						•		• •	1					•
• •														1					1	1				1	11			1	1	1								
	2.2																																					_
	÷	: :	1	: :	1	1	1	1	1		1	1	1	1	1			1	1	1		1	1	1	2.1		1	1	2	1	1.1		1	1	1	1		

図 2 - サンプル アプリケーション

IDE インサイトの使い方をざっと示すと、以下のようになるでしょう。

- 1. IDE を起動します。
- 2. F6 を押します。[IDE インサイト] が起動します。

a. 「vcl フォーム」と入力します(新規作成の VCL フォームアプリケーション (C++Builder)を選択します)。

- 3. F6 を押します。「Tedit」と入力し Enter を押します。
- 4. F6 を押します。「Tlistbox」と入力し Enter を押します。
- 5. F6 を押します。「Button」と入力し Enter を押します。
- 6. フォームでのコンポーネントの配置を決めます。
- 7. ボタンをダブルクリックして、ソース コードを開きます。
 - a. F6 を押し、([コード テンプレート] をダブルクリックして)テンプレートの一覧を 表示します。
 - b. 操作をキャンセルする場合は、Esc キーを押します。
- 3. コードエディタでソースコードを入力します。
 a. ListBox1->Items->Add(Edit1->Text);
- 9. ツールバーの 본 をクリックして、アプリケーションを実行します。

上記の例では基本的に、Button1 がクリックされたとき、Edit1 テキスト ボックスの内容を取得して ListBox1 の項目に追加します。

アプリケーションの作成スピードは速く、しかも IDE 内のほぼあらゆるものを見つけることができ るので開発者にとって大いに助けになります。そのうえ、ベースとなる IDE が拡張されたときに IDE インサイトを拡張できることも有用性を高めています。

伝統的な開発アプローチ

一部の開発者は、以前の Delphi 7 や C++Builder 6 で採用されていた IDE レイアウト アプローチをとても気にいっていました。これらの旧製品ではシングル ドキュメント インターフェイス (Single Document Interface: SDI)のみサポートしており、それはモニタが複数ある場合にうまく機能しました。これらの製品のリリース後ずっと、RAD Studio はマルチ ドキュメント インターフェイス (Multiple Document Interface: MDI)に基づいており、限られたクラシック モードはサポートされていましたが、設計画面を完全に分離することはできませんでした。現在 RAD Studio 2010 では、SDIの完全なサポートが有効になりました。

File Edit Search \ Standard Additional Win:	New Refactor Project Run (2 System Wn 3.1 Dialogs Data Acces	Component Tools Window s Data Controls dbExpress Datasn	Help 🥥 🖬 🖬	Indecked 🐱 💁 🐴	0 (0 7 5 +
DTRAC	▞▐▏┉⋉∊⋷⋷∊⋼	i) () T	Anne Alimantinai.	a sector factores	1
roject9.dproj - P	🕽 Form8		_ 🗆 🔀	- ox	Object Inspector ×
tete i∎•	EMP_NO DBEdet	CDREFE	FBCONNECTION	(1) (1)	Properties Events
	PROJ_ID D0Edd2 Summary Information: PINST_JAANE D0Edd3 LAST_YAANE D0Edd4 SALARY D0Edd5	R: C	DetaSetTroucter ClearDataSet1 DetaSetTroucter1 ClearDataSet1 DetaSource1 PataSource1 PataSource1 PataSource1 PataSource1 PataSource1 PataSource1 PataSource1 PataSource2 ClearDataSet2 PataSource2 Pat	lasses, Graj Lerbase, FM Jer, SglExp;	Action ActiveCool ActiveCool ActiveCool ActiveCool ActiveCool ActiveCool ActiveCool ActiveCool ActiveCool ActiveCool ActiveCool BioMode Dd.HTT 6500 BioMode Dd.HTT 6500 BioMode Dd.HTT 6500 BioMode Dd.HTT 6500 BioMode Dd.HTT 6500 BioMode Dd.HTT 6500 Cool
Interbase {IBC Label1 Label2	IBCON	NECTION: TSQLConner YEE: TSQLDataSet;	ction;	>	Caption

図 3 - SDI アプローチを用いた Delphi

コードエディタによるコーディング作業

RAD Studio 2010 のコードエディタでは、開発者の生産性を飛躍的に高めるコード入力支援機能を利用できます。エディタはタブ付きウィンドウになっており、一度に複数のファイルを開くことができます。

```
🔐 Welcome Page 👩 nxCSSClassesTests 🐒 nxCSSBaseClasses 🐲 nxCSSBorderClasses 🐲 nxCSSFontClasses
  186
      3
   E constructor TnxCSSFontStyle.Create (aFontStyle: TnxCSSFontStyleEnum);
     begin
  1.90
       inherited Create(sFontStyle);
       FFontStyle := aFontStyle;
     end:
   • Infunction TnxCSSFontStyle.GetStyleValue: string;
      begin
       Result := nxCSSFontStyleStrings[FontStyle];
      end;
   · procedure TnxCSSFontStyle.SetFontStyle(Value: TnxCSSFontStyleEnum);
  200 begin
        if FFontStyle <> Value then
        begin
             ....
```

図4-コードエディタ

コードエディタには、以下のような支援機能が搭載されています。

- 構文強調表示:コメント、文字列、識別子、キーワード、予約語などを色分けして表示し、容易
 に識別できるようにします。
- コード入力補完:コード入力中に利用可能な識別子のリストを表示し、入力の手間を軽減します。

- ライブテンプレート:わずかなキーストロークで、コードブロックを呼び出し、必要部分を埋め てすばやくコードを完成させることができます。ライブテンプレートはシンプルな XML ファイ ルとして定義されているので、容易に独自のテンプレートを作成できます。
- クラス補完:クラス宣言で自動的に実装スタブを生成します。
- リファクタリング:動作を変更することなく安全、確実に、コードを読みやすく、よりよく構成 された内容に変更します。
- CodeInsight: 呼び出す関数などのパラメータ情報をポップアップ表示します。
- ErrorInsight: 編集中のコードの構文エラーを自動検出し、波下線で示します。
- HelpInsight: コード内の識別子についての基本的なドキュメントと宣言をポップアップウィン ドウで表示します。
- ブロック補完:コードブロックの対を正しく記述できるようにします。例えば、エディタ内で、 begin と入力して Enter キーを押すと、対となる end が自動的に追加されます。コーディン グ作業を中断することなく、正しくフォーマットされたブロックを記述できます。
- 文脈依存型ナビゲーション: Ctrl キーを押しながら識別子をクリックすると、その識別子の宣言 部または実装部にナビゲートできます。複数のファイルでコードをナビゲートしても、ブラウザ の操作のように、すばやく前のコードに戻ることができます。単純なキーストロークで、開発者 は、クラスメソッドの宣言と実装を往復できます。IntelliMouse もサポートしており、大きなフ ァイルのナビゲートも容易です。
- 行番号とブックマーク:行番号により、ファイル内のコードの場所をすばやく特定できます。また、ブックマークを設定することで、アプリケーションコード内の特定の場所にすばやく戻ることができます。
- 同期編集:複数行のコードを選択すると、同期編集アイコンが余白に表示されます。同期編集モードに入ると、ハイライトされたブロックのテキストを、簡単に検索して変更することができます。同期編集は、限定されたブロック内の識別子を変更するのに使います。プロジェクト全体にわたって文脈を考慮して識別子を変換するには、名前変更リファクタリングを使います。
- マクロ:共通の操作をマクロとして記録して再生できます。
- Delphi にも C++ にも、新しいコード整形が追加されました。



図 5 - コード フォーマッタ

[編集] メニューの [ソースの整形] を選択するか Ctrl + D を押すだけで、[ツール | オプション | フ オーマッタ] ダイアログで設定されている仕様に従ってコードが自動的に整形されます。

全般的に、コードエディタは、コーディング作業を簡単かつ洗練された方法で、効率的に行えるよう になっています。

フォームデザイナによるユーザーインターフェイスのビジュアル設計

コーディング以外の時間の多くは、ユーザーインターフェイスの設計のために、コンポーネントを使ったフォームのレイアウトに割くことになります。IDEは、このために強力なフォームデザイナを提供しています。RAD Studio 2010のフォームデザイナは、設計時にフォームのWYSIWG(What-You-See-Is-What-You-Get)レイアウトを可能にするウィンドウそのものです。開発者は、このウィンドウにツールパレットからコンポーネントをドラッグ&ドロップして、フォーム上の適切な位置に配置できます。コンポーネントは、ビジュアルガイドライン機能を使って、余白設定や位置合わせを簡単に行えます。コンポーネントをフォーム上でドラッグすると、他のコンポーネントとの間に色のついた線が表示され、コンポーネント間の幅や位置合わせのタイミングを確認できます。

yle Settings	
ont Text	Borders Position Misc
ont-size:	Length: Suffix:
	▼ 1 ÷
ont-weight:	
	_

図 6-ビジュアルガイドラインを使って簡単にコントロールの位置合わせができるフォームデザイナ

コンポーネントをフォームに配置したら、オブジェクトインスペクタでプロパティを設定できます。 すべてのコンポーネントのプロパティとイベントは、オブジェクトインスペクタにリストされます。 ここで、コンポーネントの外観や動作、コンポーネントに対してなされた操作に応答するコードを指 定することができます。

Tool Palette	X
B 🗸 🔓 🕈	
🗉 Standard	^
🗄 Additional	
🗆 Win32	
TTabControl	_
TPageControl	
🗊 TImageList	
TRichEdit	
TTrackBar	
🚥 TProgressBar	
TUpDown	
THotKey	~

図7-ツールパレット

例えば、オブジェクトインスペクタを使って、開発者は、コンポーネントの位置やサイズを設定できます。ツールバーにボタンを追加し、編集ボックスのテキストを変更し、フォームの背景色を変更するといった操作が可能です。さらに、OnClick、OnMouseOver、OnKeyDown などのイベントが発生したときに実行されるコードを記述できます。

ツールパレットには、IDE にインストールされたすべての VCL コンポーネントが登録されています。 デフォルトでは、RAD Studio 2010 出荷時に含まれるすべての標準コンポーネントが含まれます。こ こには、追加でサードパーティコンポーネントなどを登録できます。

ツールパレットは完全にカスタマイズ可能で、コンポーネントを自由にカテゴリ分けできます。フィ ルター検索機能で目的のコンポーネントをすばやく発見し、マウス操作とキーボード操作のどちらで もすばやくフォームに配置できます。

プロジェクトマネージャによるアプリケーション構成要素の管理

アプリケーションはすぐに多数のフォームやコードファイルによって、複雑化してしまいます。多く のアプリケーションは、任意の数の異なるバイナリやプロジェクトによって構成されます。RAD Studio 2010 のプロジェクトマネージャは、IDE でのプロジェクト管理を担当します。プロジェクト マネージャは、プロジェクトに含まれるファイルやフォームを管理し、複数のプロジェクトをプロジ ェクトグループに含めることができます。開発者は、新しいフォームやファイルを作成したり、既存 のフォームやファイルをプロジェクトに追加でき、新しいプロジェクトをプロジェクトグループに追 加できます。プロジェクトは、プロジェクトマネージャからコンパイル・ビルドでき、ファイルとフ ォームは IDE で開くことができます。プロジェクトは、必要に応じて再構成できます。プロジェク トマネージャでは、アプリケーション開発に必要なファイルやフォームの管理に関するすべての操作 をハンドルしています。



図 2 – プロジェクトマネージャ

また、プロジェクトマネージャは、プロジェクトごとに複数のビルド設定を管理できます。特定のオ プションセットを管理し、保存、再利用することもでき、これらの設定の管理がこれまでになく簡単 になりました。

そのうえ、プロジェクトの開き直しを制御できる機能が大幅に強化されました。[ツール | オプション… | [開き直す] メニュー] で、プロジェクト数や表示されるファイルを完全に制御できるようになりました。

Environment Options	Capacity	
Object Inspector	Number of projects 10	
Tool Palette		
Colors	Number of files: 15	-
Environment Variables		
Reonen Menu	Reopen items	
Explorer	Projects	^
Delphi Options	Cil MuDacumente\ BAD, Studio\ Brojecte\ enn. bia. avample\ BidEvample.charej	-
Type Library	C:\mybocuments\R4D_Studio\rrojects\Cpp_Dig_example\BigExample.coproj	-
Library - Win32	C:\MyDocuments\RAD Studio\Projects\2010Example1(Projects/d0010	
Library - Translated	C:\MyDocuments\RAD Studio\Projects\2010_temp_example\ProjectGroup3 aroupproj	
VCL Designer	C:\Documents and Settings\All Users\Documents\RAD Studio\7.0\Demos\DelphiWin3.	=
Editor Options	C:\MyDocuments\RAD Studio\Projects\project8\Project8.cbproj	
- Source Options	C:\MyDocuments\RAD Studio\Projects\DSCPP_example\DSCPPServer.cbproj	
Color	C:\Documents and Settings\All Users\Documents\RAD Studio\7.0\Demos\CPP\Apps\I.	
Display	C:\MyDocuments\RAD Studio\Projects\rad2010_example_intro_101\stocks.dproj	
Key Mappings	C:\MyDocuments\RAD Studio\Projects\ProjectGroup2.groupproj	
Code Insight		
HIML Options	Files	
WohEnan	C:\Program Files\Embarcadero\RAD Studio\7.0\source\Win32\Visualizers\DateTime	
Formatter	C:\Program Files\Embarcadero\RAD Studio\7.0\source\Win32\db\DB.pas	
C++	C:\Program Files\Embarcadero\RAD Studio\7.0\source\Win32\vcl\Gestures.pas	
Indentation	C:\Program Files\Embarcadero\RAD Studio\7.0\source\Win32\vcl\GestureMgr.pas	
Spaces	C:\Program Files\Embarcadero\RAD Studio\7.0\source\Win32\vcl\GestureCtrls.pas	~
Line breaks	Doloto Nenovictort Eilos Doloto Claz	
🖨 Delphi		
a copin		

図 9 - [開き直す] メニューの構成ページ

ビジュアルデバッガ

問題に遭遇することは、開発者にとって不可避です。バグを発見したり、アプリケーションの予期しない動作を見つけたときには、アプリケーション内部に立ち入って、実行プログラムで何が起こっているのかを見る必要があります。RAD Studio 2010 では、IDE に統合された強力なデバッガを提供しており、実行アプリケーションの内部を詳細に確認できます。

	1.1	end else
	180	begin
&	1.1	FileSaveAs1.Execute;
۲	1.1	TempFilename := FileSaveAs1.Dialog.FileName;
۵	1.1	if TempFilename FileSaveAs1.Dialog.FileName 'C:\junk\junk.txt'
	1.1	begin
٠	-	<pre>FFilename.Filename := TempFilename;</pre>
۲	1.1	WriteTheFile;
•	-	FFilename.Filename := TempFilename; WriteTheFile;

図 10 - ビジュアルデバッガ (ブレークポイントで停止)

デバッガ ビジュアライザは 2010 で新たに導入された機能で、判読しにくいデータをわかりやすく表示する特別なビューアです。判読しにくいデータとは何でしょうか。例えば TDate オブジェクト、 TTime オブジェクト、TDateTime オブジェクトのようなものです。これらは通常、長い表現で格納 されます。この新しいビジュアライザを使用すると、人にも読みやすい形式でデータを表現できます。 これは、バイナリ形式では通常は理解できないカスタム オブジェクトまたはデータを扱う多数の開 発者の役に立ちます。

TStrings	Strings Visualizer for SL						
Index	Value						
[0] [1] [2]	Hello world Who's on first? Delphi is cool						

図 11 - デバッガ ビジュアライザ

アプリケーションを IDE 内で実行すると、デバッガはアプリケーションの実行を制御し、実行プロ セス全体にわたって、詳細な情報を検査できるようにします。開発者は、ブレークポイントをコード 中の任意の場所に設定して、実行を停止させることができます。ブレークポイントはカスタマイズ可 能で、任意のトリガーを設定できます。例えば、任意の回数だけ実行されたり、変数や関数を評価し て特定の値だった場合などに、アプリケーションを停止させることができます。例外が生成された場 合には、暗黙的なブレークポイントとして働きます。実行が停止すると、デバッガは、スコープ内の すべての情報にアクセスできるようにします。開発者は、監視式を設定して任意の変数の値をトラッ クしたり、コードをステップ実行して、1行ずつ動作を確認していくこともできます。デバッガは、 現在の呼び出し履歴を表示します。また、すべてのロード済のモジュール、アプリケーションに関連 するすべてのスレッドの状態も表示します。より詳細な情報が必要な場合には、デバッガは CPU レ ベルのビューも表示できます。これにより、実行されているアセンブリコードを確認できます。

個々のスレッドを凍結または凍結解除できる機能も、2010 で新たに導入されたものです。これは、 大多数のアプリケーションがマルチスレッド対応になって新しいマルチコア プロセッサを活用する ようになるにつれて、ますます重要になってきています。

Watches					
🍓 Local Variables	4				View Source Ctrl+V
Unit8.TForm8.Button1Cl	lick(Go to Source Ctrl+S
Name Value					Make Current
Self ([csl Sander E217	nherit				<u>F</u> reeze
Selider E217	1 Val				Freeze All Other Threads
					T <u>h</u> aw
	<				Thaw <u>All</u> Threads
		30: 1 In	sert Modified Code Desig		Process Properties
Thread Status					
Thread Id	State	Status	Location		<u>S</u> tay on Top
Project8.exe (5980)				~	Dockable
5324	Stopped	Breakpoint	C:\MyDocuments\RAD Studio\Projec	is\	
Pa 2952	Stopped	Unknown	\$7C90E514		
(m/h					

🖺 Event Log 糝 Breakpoint List ڰ Thread Status

図 12 - デバッグ時のスレッド凍結/凍結解除インターフェイス

このインターフェイスではまた、開発者が個々のスレッドにブレークポイントを設定することもできます。

データエクスプローラによるデータアクセス

多くのアプリケーションでは、データアクセスを必要とします。IDE に統合されたデータエクスプロ ーラは、データベースデータにアクセスする強力なツールです。データエクスプローラで開発者は、 DelphiのデータベースアクセステクノロジーdbExpress がサポートするデータベース接続を作成でき ます。データベース接続を作成すると、データベースのデータやメタデータを表示できます。データ ベースに接続し、データを閲覧できるようになったら、データエクスプローラからフォームデザイナ に、接続やテーブルをドラッグすることで、データアクセスコンポーネントをフォーム上に作成でき ます。

これにより、Firebird などの新しいデータベースもデータエクスプローラでサポートできるようになり、それと共に機能豊富なツール群もすべて提供されます。これらのテーブルをデータ エクスプロ ーラからドラッグして、フォームまたはデータモジュールに配置することができ、これらが接続およ びテーブル コンポーネントを自動的に用意するので、データベースにたやすくアクセスできるよう になります。



図 13 – データエクスプローラ

さらに、データエクスプローラでは、ビジュアルクエリービルダーを使って、データを検査したりクエリーを作成できます。

VCL-ビジュアルコンポーネントライブラリ

RAD Studio 2010 では、任意のアプリケーションを構築できますが、特にその威力を発揮するのは、 Windows クライアントアプリケーションやスタンドアロンデスクトップアプリケーションの開発に おいてです。このために、RAD Studio 2010 では、VCL(Visual Component Library)と呼ばれるア プリケーション開発フレームワークを提供しています。VCL は、広範な Win32 API をカプセル化し たクラスライブラリで、Windows アプリケーションの構築をコンポーネント化します。開発者は、 設計時にフォームデザイナ上で扱うことのできるコンポーネントを使って、ビジュアルにアプリケー ションを開発できます。

anxCSSStylePropertyEditorForm	
Style Settings	Style Text:
font-size: Length: Suffix:	- font-weight: bold
font-weight:	
	Style Preview:
Label8	
	Choose Tag Type Custom Tag: Custom Tag:

図 14 - 多数の VCL コントロールを配置したフォーム

VCLは、RAD Studio 2010 での Windows アプリケーション開発の基盤となるものです。Windows ベ ースのウィンドウを TForm クラスとしてカプセル化しています(従来からの慣例で、Object Pascal のクラスには、はじめに'T'を付けます)。IDE のフォームデザイナは、ユーザーインターフェイス を設計するための「キャンバス」を開発者に提供します。VCL は、標準 Windows UI コントロールを、 TButton、TEdit、TLabel、TCheckbox のようにコンポーネントにラップしています。フレーム ワークは、単純に継承することで拡張でき、独自のカスタムコンポーネントとして、容易に IDE に 追加して利用できます(結果として、大変豊富なサードパーティ開発者コミュニティが形成されてお り、商用、無償、オープンソースのコンポーネントが流通しています。開発者は、幅広い高度な機能 を備えたコンポーネントを活用できます)。

VCL は、実績あるコンポーネントアーキテクチャとして、永らく使用されており、いくつもの異な るプラットフォームに移植されています。VCL は、16-bit Windows に始まり、すぐに Windows 95 のサポートにより 32-bit に対応しました。また、Linux や.NET にも実装されました。

将来を見据えたタッチ機能

タッチ ベースの GUI、タブレット、タッチパッド、キオスク アプリケーションなどを迅速に作成、 または既存アプリケーションの UI を追加のコーディングをほとんどあるいはまったくせずに容易に アップグレードできます。

- プラグイン可能なジェスチャ エンジン アーキテクチャ
- サポートされているすべての Windows バージョン(2000、XP、Vista、Windows 7)で動作
- タッチ対応ハードウェアを使用、あるいは手持ちのデバイス(たとえばマウス)で動作 •
- ベースとなる VCL でタッチ インターフェイスとマルチタッチ インターフェイスを統合サポ ート
- 30 以上の標準ジェスチャ(パン、ズーム、回転など)
- カスタム ジェスチャ デザイナによる独自ジェスチャの作成
- タッチ キーボード キーボードを使用しないインターフェイスでアプリケーションとのやり 取りを促進するための完全な仮想キーボード(複数のロケールおよび言語をサポート)

DELPHI 2010 と C++BUILDER 2010 に新たに導入されたデータベース機能およびタッチ機能(例) 以下に示すのは、ジェスチャ サポートを用いてマルチデータベース アプリケーションを作成するた めの段階的なプロジェクトです。この例のために、お使いのマシンに Firebird と InterBase を両方共 ロードしておく必要があります。InterBase 2009 Developer Edition はパッケージに付属しており、 Firebird は Firebird の Web サイト (http://www.firebirdsql.org/) からダウンロードできます。

メモ: Firebird クライアント ライブラリ(fbclient.dll)がシステムパス内か RAD Studio の bin デ ィレクトリ内にあることを確かめてください。

- 1. F6 を押して、VCL フォーム アプリケーションを新規作成します(上記手順に従います)。
- 2. 「データ エクスプローラ」 タブに移動します。
 - a. Firebird 接続を開いて、テーブルなどの要素を表示します。
 - b. InterBase 接続を開いて、テーブルなどの要素を表示します。
 - c. EMPLOYEE PROJECT を Firebird 接続からフォームへドラッグしドロップします。
 - i. これで、フォームに以下の2つのコンポーネントが配置 されます。 1. FBConnnection PROJ_ID 2. EMPLOYEE PROJECT ([ツール パレット] の [Data Access] カテゴリから)フォー ムに [TDataSetProvider] をドラッグします。 i. [オブジェクト インスペクタ] の [プロパティ] タブで、 DataSet プロパティを EMPLOYEE PROJECT に設定 します。
 - b. ([ツール パレット]の [Data Access] カテゴリから) フォー
 - ムに [TClientDataSet] をドラッグします。
 - i. ProviderName プロパティを DatSetProvider1 に 設定します。
 - ii. ClientDataSet1 コンポーネントを右クリックし、[項目の設定...] を選択しま す。
 - iii. 右クリックし、[すべての項目の追加] を選択します。
 - iv. EMP_NOと PROJ_ID をフォームにドラッグします。DataSource1 が追加 されることに注意してください。
 - ([ツール パレット] の [Data Controls] カテゴリから) フォームに [TDBNavigator] を C. ドラッグします。
 - i. 【オブジェクト インスペクタ】の [プロパティ] タブで DataSource1 に接続し ます。



図 15 - フィールド エディタ

a.



図 16 - 現段階での画面レイアウト

- 10. [データ エクスプローラ] で InterBase を開き、EMPLOYEE テーブルをフォームにドラッグ します。
 - a. これで、以下の2つのコンポーネントがドロップされます。
 - i. IBConnection
 - ii. EMPLOYEE
 - b. ([ツール パレット] の [Data Access] カテゴリから [TDataSetProvider] をドラッグして)フォームに DataSetProvider2 を追加します。
 - i. DataSet プロパティを EMPLOYEE に設定します。
 - c. ([ツール パレット] の [Data Access] カテゴリから [TClientDataSet] をドラッグし
 - て)フォームに ClientDataSet2 を追加します。
 - i. ProviderName プロパティを DataSetProvider2 に設定します。
 - MasterSource プロパティを DataSource1 (Firebird データベース) に設定し ます。
 - iii. MasterField プロパティを EMP_NO (= EMP_NO) に設定します。

Field Link Designer	×
D <u>e</u> tail Fields	<u>M</u> aster Fields
FIRST_NAME LAST_NAME PHONE_EXT HIRE_DATE DEPT_NO	PROJ_ID
EMP_NO -> EMP_NO	Delete
ОК	Cancel Help

図 17 - マスタ/詳細関係の設定

- d. ([ツール パレット] の [Standard] カテゴリから)フォームに [TGroupBox] をドラッ グします。
- e. ClientDataSet2 を右クリックし、[項目の設定...]を選択します。
 - i. すべての項目を追加します。
 - ii. 以下のフィールドのみフォームにドラッグします。
 - 1. FIRST_NAME
 - 2. LAST_NAME
 - 3. SALARY

EMP NO		reka : P
DBEdit1		FBCONNEC
PROJ_ID		
DBEdit2		EMPLOYEE I
		-
Summary Informatio	n:	<u></u>
FIRST_NAME		DataSetPro
DBEdit3		· · · · · · · · · · · · · · · · · · ·
LAST_NAME		ClientData
DBEdit4		
SALARY		<u></u> ⊐⊰
DBEdit5		DataSour
		рвх
	• • • • • • • • • • • • • • • • • • •	
		TROOMIEG
		IBCONNEC

図 18 - マスタ/詳細関係の設定

- iii. EMPLOYEE_PROJECT と EMPLOYEE の Active プロパティを True に設定 します。
- iv. ClientDataSet1 と ClientDataSet2 の Active プロパティを True に設定します。 1. 実際のデータが設計画面に表示されるはずです。
- v. すべてを保存します([ファイル | すべて保存])。
- vi. ツールバーの 🕨 をクリックして、アプリケーションを実行します。

ジェスチャ/タッチ

さて、上記のアプリケーションにジェスチャ サポートを追加しましょう。タッチ機能もジェスチャ 機能も今では VCL に組み込まれているので、VCL アプリケーションではすべて、この素晴らしい機 能を活用できます。タッチ スクリーン コンピュータのないユーザーでも、マウスを通じてジェスチャ サポートを利用することができます。

- 11. ジェスチャの追加
 - a. ([ツール パレット] の [Additional] カテゴリから [TActionManager] をドラッグして) フォームに ActionManager1 を追加します。
 - i. ActionManager1 を右クリックし、[カスタマイズ...] を選択します。

(All Actions)			t2 · ₩ + 4	
Categories:	Agtions:			
No Categor	y)			
	New Action New Standard Actio	Ins n Ctrl+Ins		
	 ♠ Move Up ♦ Move Down 	Ctrl+Up Ctrl+Down		
	Cut Copy Paste W Delete Select Al	Ctrl+X Ctrl+C Ctrl+V Del		
Description				
To add actio	ons to your application sim	ply drag and drop from either	Categories or Actions onto an existing	

図 19 - TActionManager インスタンスのカスタマイズ ツール

- ii. [標準アクションの新規追加...]を選択します。
- iii. [データセット] アクション リストに移動します。



図 20 - ジェスチャでサポートするアクション

- iv. TDataSetFirst(先頭)、TDataSetPrior(前)、TDataSetNext(次)、 TDataSetLast(最後)、TDataSetInsert(挿入)を選択します。
- v. [OK] をクリックします。
- vi. ActionManager1 のカスタマイズ ツールで [データセット] カテゴリを選択して、アクションを表示します。
- vii. [閉じる] ボタンをクリックします。
- b. ジェスチャを使用できる場所について検討します(個々のコントロールかフォームか)。
 - i. フォームの Touch プロパティを強調表示します。
 - ii. 編集ボックスの Touch を強調表示します。
- c. ジェスチャ マネージャ コンポーネント(GestureManager1)をフォームにドラッグ しドロップします。

- d. フォームの Touch プロパティを Gesture Manager1 に設定します。
 - i. GestureManager1 コンポーネントを右クリックし、[カスタム ジェスチャ...] を選択します。
- e. フォームと Touch プロパティに戻ります。
 - i. [Gestures | Standard] ノード下のデフォルト ジェスチャを表示します。
 - ii. 以下のように、DataSetNext1 と DataSetPrior1 のみ設定します。
 - LeftRight = DataSetNext1
 - 2. RightLeft = DataSetPrior1
- f. すべてを保存します([ファイル | すべて保存])。
- g. 実行します。

アプリケーションが開始されると、左マウス ボタンを押したまま画面の左から右へドラッグすることで次のレコードに移動できるはずです。これは、DBNavigator1 コンポーネントの [>] ボタンを押 すのと同じです。同様に、左マウス ボタンを押したまま画面の右から左へドラッグすることで前の レコードに移動できるはずです。

C++BUILDER の新しいクラス エクスプローラ機能

C++Builder 2010 では、新しいクラス エクスプローラも特長の 1 つになっています。クラス エクス プローラはクラス モデリング機能に基づいて作成されており、プロジェクト全体に含まれているク ラス構造を表示できるようになっています。開発者は、クラスをトップダウン表示するかボトムアッ プ表示するかを選択できます。さらに、クラス エクスプローラを使用して、既存のクラス ライブラ リにクラス、メソッド、手続きを追加することもできます。そのうえ、信じられないほど高速です。



図 21 - 新しい C++ クラス エクスプローラのインターフェイス

新しい多層データベースアーキテクチャ – DATASNAP 2010

Delphi は、多層データベース開発フレームワークを搭載した最初の開発ツールのひとつです。 DataSnap と呼ばれるこのフレームワークにより、開発者は、アプリケーションでデータの提供とビジネスルールの管理を行う中間層のアプリケーションサーバーを構築できます。中間層は、データベ ースの中間に立ち、データへのアクセス性を提供し、データの処理や更新時のビジネスルールを規定 します。DataSnap では、クライアントでのデータ管理や操作のための強力なインメモリデータセッ トをはじめとする、中間層へのアクセスのための強力なクライアントソリューションを提供していま す。

RAD Studio 2010 には、DataSnap アーキテクチャのメジャーアップデートが搭載されています。従 来バージョンの DataSnap は、COM テクノロジーをベースとしていました。RAD Studio 2010 では、 この依存性を排除し、Server Methods と呼ばれる強力かつ軽量な実装によって置き換えました。 Server Methods により、開発者は中間層に属するメソッドを記述できます。これらのメソッドは、 クライアントに対してシームレスな方法で呼び出し可能にします。開発者は、サーバーメソッドをあ たかもクライアントのバイナリの中にあるかのように呼び出すことができます。Server Methods で は、string、integer、dataset、datareader、connection、OLEVariant といった、dbExpress の型シス テムに含まれる任意の型をパラメータとして渡すことができます。これは、クライアントと中間層の 間でのデータ転送における強力な手段となります。

新しい DataSnap では、従来の DataSnap サーバーもサポートしています。

New DataSnap Server	New DataSnap WebBroker Application
You may select from one of the following applications types	You may select from one of the following types of World Wide Web server applications.
VCL Forms Application	ISAPI/NSAPI Dynamic Link Library GGI Stand-alone executable
○ <u>C</u> onsole Application	
<u>Service Application</u>	Class Name:
You may select one or more communication protocols	
▲ ICb/Ib	You may choose from the following DataSnap options
<u>Н</u> ТТР	Add Server Methods Class
Authentication	Ancestor: TDSServerModule
	Include sample methods
Add Server Methods Class	TDSServer
	TDSServerClass
Ancestor: TDSServerModule	
☑ Include sample methods	TDSHTTPServiceAuthe
OK Cancel <u>H</u> elp	TDSTCPServerTransport

図 22 - DataSnap 向けの新しいツール

新しい DataSnap では、HTTP/HTTPS でのアクセス、RESTful な Web サービスへのアクセス、認証 と暗号化を追加できる機能をサポートしています。そのうえ、IIS サーバーへのインプロセス配置も 可能です。

UMLとQA 検査/測定の完全な統合

モデル ビューを直接プロジェクトからリバース エンジニアリングすることができます。これにより、 プロジェクト内のクラスがツリー形式で表示される場合に階層的なモデル ビューが作成されます。 また、UML クラス図で表現することもできます。このモデルとダイアグラムは、プロジェクトのド キュメントを自動生成するのに使用することができます。



図 23 - モデル ビューから生成された UML クラス図

UML 統合では、コード ベースのモデルとコード ベースでないモデルの両方をサポートしています。 つまり、開発者は特定の言語を想定せずにモデルを作成でき、RAD Studio ではそれらの非言語モデ ルからソース コードを生成することができます。パッケージにはまた、ソース コードの完全な QA 測定/検査も含まれており、それを用いて、オブジェクト コードの健全度をチェックでき、質の低い コーディング慣習を見つけることができます。

新しい RTTI および RTL のサポート

実行時型情報(RTTI)は、型に関する情報を実行時に取得できるプログラミングパラダイムです。 RTTI生成が有効な場合、結果として出力されるバイナリには、型に関する情報(たとえば、クラスの分類階層、宣言済みフィールド、注釈付き属性など)が記載された特別なメタデータが含まれています。 RTTIユニットで提供される機能を用いれば、この情報を実行時に取得できます。最終的には、 RTTIを公開するあらゆる型を操作できる、より抽象的で一般化されたフレームワークを作成できるようになります。

検索コマンドの変更点

さまざまな検索コマンドの機能が以下のとおり強化および拡張されました。

[検索 | 検索...]コマンド(Ctrl + F)は設計変更され、ダイアログボックスとしてではなく、コードエディタウィンドウの下端にタスクバーとして表示されるようになりました([検索]も参照)。

🗙 🕂 🖓 🖉 🗌 Case sensitive 🗌 Whole words 🗌 Regular expression 🗌 Search selection

- IDE では、検索一致箇所がすべて強調表示されます。最初に見つかった一致箇所とそれ以外の一 致箇所は異なる色で強調表示されます。
 - 使用される 2 つの色は、IDE の配色であらかじめ定義されています。
 - [ツール | オプション... | エディタ設定]の[色]ページで、[追加の検索一致部分の強調表示] 要素の前景色と背景色を選択することにより、これらの色をカスタマイズすることができます。
 - 検索一致箇所をすべて強調表示しないようにするには、[ツール | オプション… | エディ タ設定]の[一致する検索結果をすべて表示] をオフにします。
- インクリメンタル サーチ にも のような新しい検索バーが 用意されています。検索文字列フィールドに入力するか、以前使用した検索文字列のリストから 該当するものを選択すればよいだけです。
- [ファイル内の検索]ダイアログボックス([検索 | ファイル検索...])には、*.pas や*.cpp などの ワイルドカードによる指定が可能な新しいフィールド([ファイルマスク])があります。[ディレ クトリ]フィールドでディレクトリ名をセミコロンで区切るか新しい[フォルダとグループ]ボタ ンをクリックして[ディレクトリの選択]ダイアログボックスを開くことで、複数のディレクト リを指定することもできます。[ディレクトリの選択]では、ディレクトリリストとディレクトリ グループを作成できます。詳細については、以下を参照してください。
 - [ファイル内の検索] ダイアログ ボックス
 - ディレクトリ グループの検索
 - ディレクトリのリストの検索
 - [ディレクトリの選択] ダイアログ ボックス

RAD STUDIO 2009 の新機能

新バージョンへの移行を検討して製品評価をしている方は、新機能に一番興味があると思います。このセクションでは、RAD Studio 2009 で導入され、さらに多くは RAD Studio 2010 でも拡張された 新機能を紹介します。

データベースの設計とモデリング

RAD Studio 2010 Architect Edition には、データベース設計とモデリングのためのソリューションが 同梱されています。開発者は、ER/Studio Developer Edition を使って、エンティティリレーションシ ップやデータベースの物理モデルを作成することができます。作成したモデルは、ER/Studio がサポ ートする任意のデータベースにエクスポートできます。

ER/Studio Developer Edition については、EDN の以下の記事で、オンラインデモをご覧いただけます。

http://windemo1.codegear.com/Tiburon/LaunchReplays/ERStudio/ERStudio.html

DELPHIの新しい言語機能

ジェネリックス

ジェネリックスを使えば、特定の型を指定することなく、その型を安全に参照できるコードを記述で きます。ジェネリックスは、パラメータ化された型ともいわれ、特定の型に依らない処理を行う「ジ ェネリック」クラスを記述できます。ジェネリックスを使用するクラスにリストがあります。リスト を記述するときには、リストに含まれる項目の型は必要ありません。

10	
1.1	type
1.1	
- 1 E	TControlReporter <t: constructor="" tcontrol,=""> = class</t:>
I	private
-	FInternalType: T;
1.1	public
1.1	constructor Create;
1.1	procedure ProcessComponent;
1.1	<pre>procedure SetType(aT: T);</pre>
20	end;

図 24 – Delphi で制約を宣言したジェネリッククラス

RAD Studio 2010 では、ジェネリックメソッド、ジェネリックスの制約を含み、ジェネリックスをフ ルサポートしています。制約は、ジェネリック型を特定の機能セットに限定する機能です。例えば、 制約によって、ジェネリック型を、コンストラクタを持つものや特定のインターフェイスを実装した ものに限定することができます。

RAD Studio 2010 では、ジェネリックなリスト、コレクション、スタック、キューなどを新しいラン タイムライブラリでサポートしています。

無名メソッド

無名メソッドは、パラメータとしてコードブロックを渡すことができる機能です。それは、特定の名 前を付けていない関数やプロシージャといえます。無名メソッドは、変数に割り当てられるか、メソ ッドへのパラメータとして使われるエンティティとして、コードブロックを扱います。さらに、無名 メソッドは変数で参照可能で、メソッドが定義されるコンテキスト内で、その値を変数にバインドで きます。つまり、Delphiの無名メソッドは、コードブロックを通るときに状態をキャプチャできるフ ル機能のクロージャータイプです。

```
type
            // method reference
           TProc = reference to procedure(x: Integer);
        procedure Call(const proc: TProc);
         begin
          proc(42);
        end;
     20 procedure DoSimpleAnonymousMethod;
         var
          proc: TProc;
         begin
           // anonymous method
           proc := procedure(a: Integer)
           begin
             Writeln(a);
_
           end;
usN
           Call(proc);
         end;
nd S
```

図 25 - 無名メソッドの使い方を示すコード

VCL の新機能

リボンコントロール

RAD Studio 2010 には、Office 2007 のユーザインターフェイスコントロール、いわゆる「リボンコ ントロール」のフル実装が含まれます。既存の VCL アーキテクチャを用いて Object Pascal によって 構築されており、開発者は、リボンコントロールを使って、モダンで強力、そして容易に操作できる GUI アプリケーションを、特別なコーディングを必要とせずに作成できます。Delphi の強力な TActionManager テクノロジーを使って構築されているため、既存のアプリケーションも、極めて簡 単にこの新しいインターフェイスデザインパラダイムに移行できます。

r Microsoft Win	ndows - RibbonDemoMainF	orm							
ject <u>R</u> un <u>C</u> ompo	onent <u>T</u> ools <u>W</u> indow <u>H</u> elp	Default Layout	 P P						
2 🙆 🙆 🗄	🕨 🗸 🚺 🔲 🗎 🗧	🔶 + + + + 🥔							
9 8 🔓	P 🛛 🔐 Welcome Page 📾 RibbonDemoMainForm								
	•								
<u>^</u>	Ribbon Demo								
	🔊 🖌 🗧 🔊) Ŧ	Document1 - CodeGear Ribbo						
	Home Insert	Page Layout Control Typ	oes References Style						
	🖰 🔏 Cut	Calibri (Body)							
	Paste Copy								
	✓ ✓ Format Painter	R Σ Π ↓ spe X ⁵							
≡	Clipboard 🐄	Font							
			a						
		PopupActionBar 1	ilChart ScreenTipsManager1						
		<u>a</u>	정폐						

図 26 – フォームデザイナ上でリボンコントロールを使った Delphi フォーム

新しいコンポーネント

TPNGImage

RAD Studio 2010 では、新たに PNG(Portable Network Graphics)イメージフォーマットをフルサポートしています。 TImage と TImageList コンポーネントは、いずれも PNG イメ ージに対応しています。開発者は、PNG イメージをフォーム やツールバー、メニュー、ボタンなどに表示できます。

TCategoryPanelGroup

TCategoryPanelGroup は、任意の数の折りたたみ可能なパネ ルを複数持つことのできるコンポーネントです。「Outlook ツ ールバー」と同様、TCategoryPanelGroup は任意の数のパネ ルを持つことができ、そのパネルには他のコンポーネントを 任意の数配置可能です。

図 27 – TCategoryPanelGroup には、3 つの CategoryPanel が あり、2 つがたたまれています



TBallonHint

RAD Studio 2010 では、柔軟かつカスタマイズ可能なコントロールヒントシステムをサポートしてい ます。カスタムヒントは、TCustomHint クラスを継承して、容易に作成できます。



図 28 -- バルーンヒントを表示する VCL コントロール

RAD Studio 2010 では、デフォルト実装として TBalloonHint を提供しています。すべての VCL コン ポーネントは、CustomHint と呼ばれるプロパティを持ち、TCustomHint を継承したコンポーネント を、独自のヒント用途に指定できます。

TBalloonHintは、タイトルとメインテキストを持ちます。ヒント内にはイメージも表示できます。

TButtonedEdit

TButtonedEdit は、内部にイメージを持つエディットボックスです。イメージは、エディットボック スの右端と左端に表示されます。それぞれのイメージは、クリックイベントがあり、それぞれのグリ フをクリックしたときに実行されるコードを指定できます。

-TButtonedEdit TButtondEdit can have a TButtonedEdit button on both the left and the right. Both buttons will The 'X' deletes reside within the edit iself. In this case, the right button will clear any text in the control.

図 29 – 検索用に使用した TButtonedEdit。赤い X ボタンはテキストのクリアが可能

既存コンポーネントのアップデート

既存の VCL コンポーネントにも、いくつものアップデートがあります。

- TImageListは、TImageによってサポートされるすべてのタイプのイメージを含めることができるようになりました。
- TButton は、TImageList を保持できるようになり、ボタンにイメージを表示できるように なり ました。イメージは、任意の位置に合わせて配置できます。また、Vista の機能であるコマンド リンクやスプリットボタンをサポートする新しいスタイルも含まれます。
- TTreeView では、項目が開かれているか、たたまれているかによって、異なるイメージを表示 できるようになりました。
- TListView では、Vista のグループをサポートするようになりました。

- TRichEdit は、Windows RichEdit 2.0 仕様をサポートします。
- TProgressBarは、ポーズ、エラーを含む Vista スタイルをサポートしています。また、Vista の 機能であるマーキースクロールにも対応しています。
- TEdit では、パスワード入力文字のカスタマイズができるようになりました。また、TextTip プロパティを使って、TEdit にフォーカスがなく何も入力されていないときに表示されるヒントを 指定できるようになりました。

VCLの新機能については、EDNの以下の記事で、オンラインデモをご覧いただけます。

http://windemo1.codegear.com/Tiburon/LaunchReplays/DelphiVCL/DelphiVCL.html

IDE の新機能

リソースマネージャ

多くの Windows アプリケーションには、ビットマップ、カーソル、フォント、その他のデータといったコンパイル済のバイナリに入れることのできる Windows リソースが含まれています。RAD Studio 2010 では、新たにリソースマネージャが搭載され、プロジェクトへのリソースの追加、管理 が容易になりました。開発者は、すべての標準 Windows リソースタイプを追加でき、名前を指定して、TResourceStream クラスを使ってコードで利用できます。

esource files			Properties
File name	Type	Identifier	Resource <u>I</u> dentifier:
💼 Autumn Leaves.jpg	RCData	LEAVES	CHEMICAL
folder_closed_64_d.bmp CHEMICAL.ICO	Bitmap Icon	CLOSEDF CHEMICAL	Resource Type:
			Icon
	<u>A</u> dd	<u>R</u> emove	
		ОК	Cancel Help

図 30 – リソースマネージャ

ビルド設定

多くの Delphi プロジェクトには、単一のアプリケーションの作成のための特定のプロジェクトが複数個含まれて鋳ます。開発者は、単一ソリューションを DLL、EXE、パッケージなど複数のファイル で構成します。すべての異なるプロジェクトやコードを管理するのは困難です。特に、ビルド作業は 大変です。

 Delphi Compiler Compiling 	Build Configuration 🥜 Release		Apply Options	
Hints and Warnings	Code generation			
	€ Code inlining control	On		
Directories and Conditionals	Code page	0		
	Emit runtime type information	False		
Version Info	E Minimum enum size	Byte		
Packages	Optimization	True		
Debugger		False		
		Quad word	ad word	
Environment Block		False	Apply Option Set	
Build Events	E String format checking	On		
	Debugging		Option Set File	
	> Assertions	False	C:\temp\TestOptionSet.optset	
	Value from "Base"	True		
	☑ Debug information	False	Action	
	Local symbols	False	ACOOT	
	 Symbol reference info 	None	Add as reference	
		False	Modify this configuration	
	■ Use imported data references	True		
	Other options		Priority: Replace all values	
	Additional switches to pass to the	con		
	Generate XML documentation	False	OK Cancel Help	
	I ook for 8.3 filenames also	False		

図 31 -- プロジェクトオプションダイアログで、オプションセットを継承して適用

プロジェクトは、しばしば異なる目的のために異なるビルド方法を用いる必要があります。テスト用、 デバッグ用、フィールドテスト用、リリース用といったビルドが必要になります。それぞれの異なる ビルドでは、異なる設定と異なるコンパイラオプションセットが必要です。RAD Studio 2010 では、 柔軟なビルド設定管理システムを導入し、IDE とコマンドラインで同一の設定を利用できるようにな っています。

さらに、RAD Studio 2010 では、開発者はプロジ ェクトオプションセットを作成し、ビルドオプシ ョンを構成・管理できます。オプションセットは、 他の設定を「継承」することができ、必要に応じ て、設定を引き継いだり、上書きすることができ ます。オプションセットは、ファイルに保存して、 特定のプロジェクトや他の設定に適用できます。

クラスエクスプローラ

RAD Studio 2010 には、クラスエクスプローラと いう新機能も搭載されています。クラスエクスプ ローラは、Delphiのクラスモデリング機能の上に 構築されており、プロジェクト全体にわたってク ラスの構造を見ることができます。開発者は、ト ップダウンおよびボトムアップビューでクラスを 選択して見ることができます。さらに、クラスエ クスプローラでは、既存のクラスライブラリにク ラス、メソッド、プロシージャなどを追加するこ ともできます。



図 32 – クラスエクスプローラ

翻訳ツール

国際化は、RAD Studio 2010 の重要なテーマです。新しい国際化機能を活用すれば、他の言語にアプリケーションを翻訳する作業を効率化できます。RAD Studio 2010 には、ITE(Integrated Translation Environment)とETM(External Translation Manager)のアップデートバージョンが搭載されています。

ITE は、アプリケーションの翻訳バージョンの作成を支援する IDE 組み込みのツールです。ITE は、 プロジェクトを複数言語対応可能にします。プロジェクトをスキャンし、すべての文字列、キャプシ ョン、テキストなどを抽出し、翻訳可能にします。開発者や翻訳者は、IDE に搭載されたツールを使 って、これらの文字列を他の言語に翻訳できます。翻訳は、プロジェクト専用のリソース DLL に配 置されます。リソース DLL があれば、アプリケーションは実行時に翻訳言語で表示されるようにな ります。

ETM は、ITE のアプトプットとともに、翻訳者や翻訳サービスに再配布されるツールです。これにより、アプリケーション翻訳のアウトソーシングの管理が容易になります。

ビルド設定とリソースマネージャの新機能については、EDNの以下の記事で、オンラインデモをご 覧いただけます。

http://windemo1.codegear.com/Tiburon/LaunchReplays/RAD Studio 2010IDE/RAD Studio 2010IDE.html

ITE と ETM については、EDN の以下の記事で、オンラインデモをご覧いただけます。

http://windemo1.codegear.com/Tiburon/LaunchReplays/ASCIInew/ASCIInew.html

COM/ACTIVEX サポートのアップデートと強化

RAD Studio 2010 には、完全にアーキテクチャを変更した COM および ActiveX 開発フレームワーク が搭載されています。この新しいフレームワークは、Microsoft IDL 仕様のサブセットである新しい 「Reduced IDL」言語をベースにしています。それは、Delphi で COM オブジェクトを定義するため

に特に設計されています。RIDL ファイルと関 連する PAS ファイルの組み合わせは、COM と ActiveX オブジェクトの定義を、純粋にテ キストファイルのみで行えるということを意 味しています。この2つのファイルは、タイ プライブラリファイル(*.TLB)にコンパイル され、プロジェクト外で利用されます。TLB ファイルは、プロジェクトのコンパイル結果 であり、プロジェクトの一部ではありません。 この方法により、Delphi COM と ActiveX オブ ジェクトは、適切に管理、マージされ、ソー スコード管理ツールなどにストアできます。 COM/ActiveX プロジェクトに関する情報を保 存するのに、もはや TLB ファイルを必要とし なくなりました。



図 33 – *.RIDL ファイルを編集するタイプライブラリエディタ

RAD Studio の COM および ActiveX の新機能ついては、EDN の以下の記事で、オンラインデモをご 覧いただけます。

http://windemo1.codegear.com/Tiburon/LaunchReplays/COM/COM.html

DELPHI 2007 で追加された新機能

以下の機能は、Delphi 2007 で追加された新機能です。これらは、いずれも RAD Studio 2010 でも利用できます。

BLACKFISHTM SQL

Delphi 2007 には、新たに Blackfish SQL Delphi Edition が搭載されました。Blackfish SQL は、SQL-92 準拠のマネージドコード RDBMS です。Blackfish は、大変柔軟なデータベースです。スタンドア ロンアプリケーションの組み込みデータベースエンジンとして使用したり、エンタープライズシステ ムのデータベース機能の一部として活用することもできます。アプリケーションや Web ソリューシ ョンでインプロセスとして実行したり、サーバーあるいは Windows サービスとして実行することも できます。配布は極めてシンプルで、バイナリ、データベースファイル、ライセンスファイルを配布 するだけです。

.NET Framework を活用することで、Blackfish SQL は、RDBMS としてのフル機能を提供しながらも、 アプリケーション組み込みからエンタープライズレベルのスケールまで幅広い柔軟性とポータビリテ ィを提供しています。

Blackfish は、以下の3つの方法で実行できます。

- 1. Windows サービスとして
- 2. スタンドアロン実行プロセスとして
- 3. インプロセスアセンブリとして

アプリケーションは、Blackfish にローカルあるいはリモートで接続可能です。リモート接続の場合、 接続アプリケーションは、TCP/IP スタックを情報転送に使用します。ローカル接続の場合、接続ア プリケーションは、他のアセンブリと同様、単純に Blackfish アセンブリにバインドします。

インストール時に、Blackfish は、Windows サービスとしてインストールでき、起動時に自動的に実 行させるように設定できます。これにより、Blackfish は、開発者のマシンで共通して利用可能にな ります。Blackfish SQL は、デフォルトで、ポート 2508 を使用します。

BLACKFISH SQL に関する追加情報

- Blackfish を用いたアプリケーション開発と配布に関する情報(英語): Developing and Deploying with Blackfish SQL and Delphi
- Blackfish SQL の機能を紹介するデモビデオは<u>こちら</u>
- Blackfish SQL 開発者ガイドは、<u>こちらから</u> ダウンロードできます
- Blackfish の主任開発者 Steve Shaughnessy が書いた Blackfish に関するブログ
- Blackfish SQL の R&D 開発者 Leonel Tognioli によるプレゼンテーション: Introduction to Blackfish SQL

VISTA サポート

Vista OS のリリースにより、Vista の新機能をサポートする多くの新しい API が提供されるようにな りました。Delphi 2007 は、これらの新しい Vista API を直接活用できる Delphi で最初のツールでし た。VCL コンポーネントとそのプロパティによって、これらの新機能をラップすることで、Delphi 2007 は、開発者がすぐに Vista の多くの新機能にアクセスできるようにしました。RAD Studio 2010 では、VCL の多くのコンポーネントに新しい Vista API サポートを追加することで、さらに多くの機 能を利用できるようにしています。

もちろん、Vista サポートを組み込んだアプリケーションは、Vista 上で実行されることを期待して、 完全にその機能が有効になるように作られます。しかし、XP や Windows 2000 のマシンで実行した 場合には、Vista 固有の機能は、動作しないのではなく、XP の基本機能として「それなりに」表示さ れます。

グラス効果

Delphi 2007 では、Windows Aero インターフェイスのサポートが導入されました。VCL では、 TForm に、グラスフレームを作成するプロパティが追加されました。ネイティブ Windows 開発者は、 ネイティブアプリケーションでこの機能を活用できます。TForm.GlassFrame.Enabled プロパティ を True に設定するだけです。開発者は、フォームの外枠のグラス効果の幅を設定したり、 SheetOfGlass プロパティを True に設定して、フォーム全体をグラス状にすることもできます。

アプリケーションがグラス効果を使っていて、Vista 以外の環境で実行された場合、グラス効果は働きません。

VISTA **ダイアログ**

新しい Vista OS では、標準ダイアログも強化され、OS のコントロールセットとして提供されていま す。VCL では、これらの新しいダイアログを TFileOpenDialog、TFileSaveDialog、 TTaskDialog コンポーネントとしてカプセル化しています。これらのコンポーネントは、新しい Vista ダイアログコントロールによって提供される機能をカプセル化しており、アプリケーションで これらの機能を極めて容易に活用できるようにします。

もし、これらの新しいダイアログを使ったアプリケーションが Vista 以外の環境で実行されており、 これらの Vista ダイアログを呼び出すと、Vista ベースのシステムでのみサポートされる機能である ために、特別な例外が発生します。開発者は、これを捕捉して、実行中の OS で適切にサポートされ た動作を呼び出すことができます。

AJAX と VCL FOR THE WEB

Delphi 2007 では、VCL ファミリーに VCL for the Web という新しい機能が加わりました。

AtoZed Software の IntraWeb テクノロジーをベースとした VCL for the Web は、標準的な Delphi ク ライアントアプリケーションを開発するのと同様の手法で、Web アプリケーションを構築可能にし ます。VCL for the Web は、コンポーネントベースのテクノロジーで、VCL for Win32 と非常に類似 した機能を提供します。開発者は、通常の VCL フォームと同じように、コンポーネントをフォーム にドロップして、プロパティを設定して、Web ページを設計します。しかし、アプリケーションは、 ブラウザで実行されます。

VCL for the Web は、Web サイトではなく、Web アプリケーションの開発に特化した最初で唯一の ツールです。他のツールよりもすばやく簡単に Web アプリケーションを構築できます。強力な HTML レンダリングエンジンをベースとして、VCL for the Web は、通常の Windows アプリケーショ ンと同じ方法で、Web アプリケーションの設計を可能にします。ドラッグ&ドロップアプローチに より、開発者はコントロールをフォームに配置し、イベントを定義してプロパティを設定できます (それは、HTMLページとフォームの組み合わせとみなすことができます)。 VCL for the Web では、さらに、AJAX テクノロジーを自動的に利用できます。可能であれば、VCL for the Web は、クライアントサイドプロセスを活用するためにアプリケーション中に必要な AJAX コードを自動的に呼び出します。設計時に非同期イベントにコードを設定するだけで、開発者は、 Delphi のみのコードで、AJAX ベースのイベントを作成できます。

VCL for the Web については、EDN の以下の記事で、オンラインデモをご覧いただけます。 http://windemo1.codegear.com/Tiburon/LaunchReplays/D2009Intraweb/D2009Intraweb.html

DBEXPRESS 4

データベースサポートとコンポーネントベースのデータアクセス機能は、Delphiの機能セットの中で も当初から中心的なものでした。そうした前提を継続して、Delphi 2007 では、dbExpress 4 と呼ば れる新しいデータアクセスアーキテクチャをその基盤に導入しました。コンポーネントレベルで dbExpress 3 と完全に後方互換性を保ち、dbExpress 4 は、単一のソースで統一されたデータ操作を 行う VCL のデータベースアクセスレイヤーとして、完全にアーキテクチャの変更がなされました。

dbExpress 4 は、データベースドライバ構築のプロセスを大幅に単純化しました。これにより、VCL 開発者は、広範なデータベースサーバーへのアクセスが可能になります。100% Object Pascal で記 述され、ネイティブとマネージドコードのクロスプラットフォームに対応しています。さらに、コネ クションプーリングやコマンドトレース機能を搭載し、開発者にデリゲートモデルが公開されていま す。これにより、データアクセスプロセスをフックし、データの読み書きプロセスで追加機能を開発 することができます。dbExpress 4 では、旧バージョンと比較して、大幅なパフォーマンス向上も実 現しています。

Delphi 2007 では、サポートされる全てのデータベースのメタデータの読み書きを可能にする豊富な メタデータクラス群も提供されています。

dbExpress 4 は、ベースのアーキテクチャであり、Delphi 2007 でのアプリケーションレベルでの変 更は軽微です。データアクセスと VCL データ対応コントロールへのデータ提供を行う VCL コンポー ネントである dbExpress コンポーネントに大きな変更はありません。そのかわり、下位のレイヤー のコードは、より効率的で拡張可能になりました。多くのユーザーはその違いに気がつかないかもし れませんが、データアクセスのパフォーマンス向上については気がつくと思います。

RAD STUDIO 2009 から導入されたその他の機能

以下のセクションでは、RAD Studio 2009 に搭載されたその他の機能のうち、製品評価を進める上で 興味を持ちそうな内容を紹介します。

統合開発環境

コードエディタ

RAD Studio 2010 のコードエディタには、コーディング作業を強力に支援します。それには、適切な メソッド名をすばやく選択し、テンプレートから共通コードを再利用したり、コーディングエラーを すばやく検出するといった、コーディング作業の労力を軽減するさまざまな支援機能が含まれていま す。

ライブテンプレート

ライブテンプレートは、わずかなキーストロークで、共通のコード構造を使ってコードを完成させる ことのできるコードエディタの機能です。ライブテンプレートは、テンプレートの記述機能、インテ リジェントコード挿入、テンプレートの変数部分のインタラクティブナビゲーション機能を提供しま す。ライブテンプレートを用いれば、必要に応じてカスタマイズすることで、小さな記憶用のコード 断片を大きなコードセットに拡張することもできます。

for	I := 0 to List.Count - 1	do
begi	loop counter	
	<tab> or $<$ shift + tab> to navigate	
end;		

図 34 – for ループを作成するライブテンプレート

図 34 では、'for' とタイプしてスペースキーを押すだけで、for テンプレートを呼び出しています。 テンプレートが表示されると、テンプレートを埋めるエントリーポイントが提供されます。このエン トリーポイント間は、タブキーによって移動でき、それぞれに必要な情報を埋めていくことができま す。システムは、各エントリーポイントに何を入力するのかを示すヒントも表示します。

ライブテンプレートは、シンプルな XML ファイルによってその動作が定義されています。そのため、 開発者は、容易に、特定の目的用のテンプレートを作成することができます。ライブテンプレートは、 さまざまな機能を実行できるスクリプトエンジンを活用することもできます。

コード補完

コード補完は、現在のコードのスコープ内で宣言された識別子についての情報をはじめ、開発者がそのクラスで使っているクラスに関するすべての情報を表示するコードエディタの機能です。

var			
Var	rishleWi	ithliongName, string.	
Val	TADIEWI	ICHALONGNAME: SCIING;	
begin	1		
Vai	ria		
end;	var	VariableWithALongName: string;	
	unit	Variants;	
end.	const	VARIABLE_PITCH = 2;	
	type	Variant:	

図 35-コード補完によって、複雑な識別子をすばやく入力

どんなアプリケーションでも、多くの識別子を使っており、しばしばそれは、説明的で長いものになります。コード補完は、こうした識別子の入力をすばやく、ミスが起こらないようにします。図 34 のように、コード補完が呼び出されると、タイピング作業中に、簡単に目的の識別子を見つけることができます。目的の識別子を見つけたら、Enter キーを押すだけで、コードエディタに入力されます。

コード補完では、さらに、クラスやレコードの適切なメソッド名やフィールドをすばやく見つけることもできます。

Design for the set	- 1		
Butto	nı.		
end;	constructor	Create(AOwner: TComponent);	
	procedure	Click;	
end.	function	UseRightToLeftAlignment: Boolean;	
	property	Action: TBasicAction;	
	property	Align: TAlign;	
	property	Anchors: TAnchors;	
	property	BiDiMode: TBiDiMode;	
	property	Cancel: Boolean;	
	property	Caption: TCaption;	
	property	Constraints: TSizeConstraints;	
	property	Default: Boolean:	

図 36 – コード補完が TButton のすべてのメソッドをリスト

図 36 のように、変数名 Button1 に続いてピリオドを入力して、CRTL+スペースキーを押すと、コード補完が呼び出されます(あるいは、ピリオドを入力して一定時間すると、自動的に表示されます)。 コード補完は、TButton クラスで利用可能なすべてのメソッドとフィールドをリストします。続けてコードをタイプしていくと、タイプした文字列で始まる項目だけがフィルターされていき、そのまま入力を続けるか、表示されたリストから候補を選択します。適切な項目を見つけたら、あとはEnter キーを押すだけで、項目がコードエディタに入力されます。

ブロック補完

ブロック補完は、コードブロックが常に適切に閉じられるように、コード構造の改善を支援します。 例えば、Delphiでは、すべての begin は、end 文に対応していなければなりません。すべての case 文も、同じように end を必要とします。ブロック補完は、開発者が特に注意することなく、こ れらの構文が確実に閉じられるように補完されます。

次のように入力します。

begin<enter キーを押します>

ブロック補完は、以下のように end を追加して、カーソルを'l'文字の場所に移動させます。

begin | end;

Error Insight

Error Insight は、コードエディタで構文エラーを視覚的に伝えます。スペルチェッカの「赤い波線」 と同じように、Error Insight はコード中の問題箇所を指摘します。

```
- B procedure TForm47.FormCreate(Sender: TObject);
var
Misspelled: string;
begin
ThisVariableDoesNotExist := 42;
ThereIsNoProcedureByThisName(aBadParameterName);
Mispelled := 'Misspelled';
end;
```

図 37 – Error Insight がコーディングエラーを指摘

Help Insight

Help Insight は、識別子に対する情報を「ツールチップ」で表示する機能です。ツールチップに表示 される情報は、開発者が記述するコードで、スラッシュ3つ(///)によるコメントに、XML タグを 使って定義することができます。開発者は、コードにコメントを記述し、それを、対象とする識別子 上にマウスカーソルが置かれたときに、ツールチップとして表示させることができます。ツールチッ プのスタイル自身は、CSS(Cascading Style Sheets)を使って設定できます。



図 38 – Help Insight のツールチップに表示される情報。この内容は、画面上で記述しているコメント がフォーマットされたものです

ビジュアル設計

フォームデザイナ

RAD Studio 2010 のフォームデザイナは、ビジュアルにイベントドリブンの開発を行うインターフェ イスを提供します。フォームデザイナでは、フォーム上でコンポーネントをビジュアルに操作できま す。開発者はツールパレットからコンポーネントをフォーム上にドラッグ&ドロップし、設計時から フォームのレイアウトを確認しながら、ユーザーインターフェイスを作成できます。設計フォームの コンポーネントは、実行時と同じような表示、動作をします。オブジェクトインスペクタを使ってプ ロパティを設定し、コンポーネントを操作できます。プロパティの変更は、設計フォームで直ちに確 認できます。フォームデザイナには、グリッドやビジュアルガイドラインといった、コンポーネント を簡単にレイアウトできる支援機能が搭載されています。

ビジュアルガイドライン

多くのアプリケーションでは、数多くのコントロールを配置したフォームを作ります。そのため、多 くの開発者が、これらのコントロールの位置を適切に揃える方法に苦慮しています。Label は、恐ら く「ラベリング」の対象となるコントロールに位置合わせしたいでしょう。よいデザインのフォーム は、コントロールがちゃんとレイアウトされており、適当にちらばっていないものです。

RAD Studio 2010 のフォームデザイナでは、ビジュアルガイドラインによって、コントロール、ある いはコントロール内のテキストで、適切に位置合わせが可能です。開発者は、マウス操作で簡単に、 適切な位置にコントロールを配置することができます。



図 39 - ビジュアルガイドラインを使ってラベルとエディットボックスを位置合わせ

ビジュアルガイドラインは、3 つの異なるインジケータを表示します(図 39 を参照)。青の線は、 コンポーネントの上、下、左、右での位置合わせを行います。マゼンタの線は、コントロールのテキ ストのベースラインで位置合わせをするガイドです。そして、グレーの線は、コントロールの Margin および Padding プロパティに従って余白が設定されていることを示します。

オブジェクトインスペクタ

オブジェクトインスペクタには2つの役割があり ます。コントロールのプロパティを表示・設定す ることと、イベントとそれによって呼び出される コードを結びつけることです。

オブジェクトインスペクタは、選択した任意のコ ントロールのプロパティを動的かつ自動的に表示 します。複数のコントロールが選択されていると きには、共通のプロパティが表示されます。オブ ジェクトインスペクタでは、デフォルト値から変 更されたプロパティの値をボールドで表示します。 プロパティは、名前順またはカテゴリ別に表示で きます。また、オブジェクトインスペクタは、ド ッキングウィンドウとして表示できます。

図 40 – オブジェクトインスペクタが TButton コンポーネントのプロパティを表示



モデリングビュー

モデリングビューは、プロジェクトから直接リバースエンジニアリング可能です。モデルビューでは、 プロジェクトのクラスをツリー構造で表示し、UML 図としても表示できます。モデルおよび図から は、自動的にドキュメントを生成することができます。



図 41 – モデルビューで生成した UML クラス図

プロジェクトマネージャ

アプリケーションプロジェクトやコード、ファイル、あるいは他のリソースなどプロジェクトに関連 するものを管理することは、どの開発ツールでも重要な機能のひとつです。RAD Studio 2010 では、 強力なプロジェクトマネージャを搭載し、コードファイル、フォームファイル、リソースをはじめと する開発プロジェクトを構成する項目を管理できます。

プロジェクトマネージャは、階層構造のプロジェクトビューを提供します。ツリーのルートは、プロ ジェクトグループで、複数のプロジェクトを保持することができます。プロジェクトグループ内の複 数のプロジェクトは、その順番を自由に変更できます。プロジェクトグループ内のすべてのプロジェ クトは、プロジェクトマネージャに登録された順序でコンパ

ビルド設定

イルされます。

開発者は、複数のビルド設定をプロジェクト用に作成し、プ ロジェクトの最適化、検索パス、その他のオプションを設定 できます。ビルド設定は、ベースの設定からオプションを継 承して、特定のビルド項目を詳細に編集できます。

図 42 – プロジェクトマネージャのビルド設定



ビルド設定には、プロジェクトマネージャからアクセスでき、特定のプロジェクトのすべての設定は、 コンテキストメニューの単一の[ビルド]コマンドからビルド可能です。

プロジェクトオプションは、オプションセットとして保存できます。保存したオプションセットは、 複数のプロジェクトのビルド設定でいつでも共有することができます。

ビルド設定マネージャは、現在のプロジェクトグループのすべてのプロジェクトをリストします。設 定名を選択して、[すべてを選択]ボタンをクリックし[適用]をクリックすれば、その設定をすべ てのプロジェクトに適用できます。

1	Build Config	uration Manager	TPODET	×
		ame:		Apply
	Available project	<u>Select All</u>		
	Name	Active Configuration	Path	<u>C</u> lear All
	Puzzle	Debug Build	D:\Users\Mr	
	FindFunction	Release Build	D:\Users\Mr	
	SpaceDust	CodeGuard	D:\Users\Mr	
				Close

図 43-3つのプロジェクトがオープンされた状態のビルド設定マネージャ

デバッガ

RAD Studio 2010 には、完全なフル機能のデバッガが搭載されています。Delphiのデバッガを用いれ ば、開発者はアプリケーションの実行を完全に制御し、実行されているマシンコードに至るまで詳細 な情報をウォッチできます。任意の変数の状態をトラックし、行単位でコードを実行できます。デバ ッガがブレークポイントで停止すると、スコープ内の任意の変数を詳細に検査できます。呼び出し履 歴はいつでも参照できるので、開発者は、現在の地点に至るまで、どのようにプログラムが実行され たのかをさかのぼって確認できます。つまり、Delphiのデバッガを用いれば、アプリケーション実行 時のさまざまな問題をすばやく確認して分析することができます。

ブレークポイント

ブレークポイントの使用は、デバッグ テクニックで最も一般的で直接的な方 法です。ブレークポイントは、特定の 時点でプログラムの実行を停止するよ うにデバッガに命令するシグナルです。 実行コードがブレークポイントで停止 すると、開発者は、アプリケーション を一時停止させたまま。その状態を調 べることができます。以下に、アプリ ケーション内部の状態を調べる方法を 示します。

図 44 – デバッガがブレークポイント で停止し、デバッガツールチップに情 報が表示される



拡張可能な監視式

デバッガの監視式機能を用いると、アプリケーション内の特定の変数をウォッチすることがで、きます。デバッガでは、指定した監視式を継続的にトラックして状態を報告します。RAD Studio では、 さらに、拡張可能な監視式機能を搭載し、ドリルダウンによって、複雑な項目の監視を容易にしました。図 44 では、あるオブジェクト内部の情報をツールチップによって監視しています。

CPUビュー

デバッガの CPU ビューは、アプリケーションのマシンコードレベルの完全なビューを提供します。 開発者は、いつでもアプリケーションを、レジスターの情報や実際のアセンブリコードの実行状況、 バイナリコード、FPU の状態といったマシンレベルで確認することができます。このレベルでは、 明らかになっていない情報は一切ありません。完全な CPU ウィンドウは、図 45 に示したとおりで すが、CPU ビューは、ドッキングウィンドウとしても独立したウィンドウとしても表示できます。

🔐 Welcome Pa	ge 📄 CustomDrawTreeView	🊈 сри						\otimes	83
[\$00B96344	=\$00B458C0 Thread	#5732	EAX	00BC771	5	~	CF	0	~
004641	4E 8BF9	mov edi,ecx	EBX	0089557	5	T	PF	0	
004641	50 8BD8	mov ebx,eax	ECX	0012F3D	4	-	AF	0	-
Custor	DrawTreeView.pas.1:	20: with TV.Canvas do	EDX	00B63E5	5		ZF	0	
004641	52 8B8360030000	mov eax, [ebx+\$00000360]	ESI	0084940	з		SF	0	
004641	58 8BB054020000	mov esi,[eax+\$00000254]	EDI	0012F3D	4		TF	0	
Custon	DrawTreeView.pas.12	22: if None1 <> nil then	EBF	0012F31	2		IF	1	
004641	5E 8B83C8030000	mov eax,[ebx+\$000003c8]	ESF	0012F31	5		DF	0	
004641	64 85C0	test eax,eax	EIF	00464D6	Ξ		OF	0	
004641	66 747F	jz \$00464de7	EFI	. 0000020	2		IO	0	
Custon	DrawTreeView.pas.12	24: if Nonel.Checked then //no pic	CS	001B			NF	0	
004641	68 80783800	cmp byte ptr [eax+\$38],\$00	DS	0023			RF	0	
004641	6C 742B	jz \$00464d99	SS	0023			VM	0	
Custon	DrawTreeView.pas.12	26: Brush.Color := BkgColorDialog	ES	0023		~	AC	0	v
004641									
004641	74 8B5060	mov edx,[eax+\$60]		0012F39C	00000	004	144		
004641	77 8B4614	mov eax,[esi+\$14]		0012F398	00000	011	1.00	**	
004641	7A E871BAFBFF	call TBrush.SetColor		0012F394	00000	011	1.1	-	
Custon	DrawTreeView.pas.12	27: Brush.Style := FBrushStyle;		0012F390	00000	OAO	• • •	а. Г	
004641	7F 0FB69310040000	movzx edx, [ebx+\$00000410]		0012F38C	00020	001	1979	-	
004641	86 8B4614	mov eax,[esi+\$14]		0012F388	00010	000	• • •	-	
004641	89 E87EBBFBFF	call TBrush.SetStyle		0012F384	00000	085		•	
Custon	DrawTreeView.pas.12	28: FillRect(ARect);		0012F380	0012F	480	• • •	.ô€	
004641	8E 8BD7	mov edx,edi		0012F37C	003F5	3AC		?S	
004641	90 8BC6	mov eax,esi		0012F378	15011	E68		.h	
004641	92 E845BEFBFF	call TCanvas.FillRect		0012F374	00050	F50			
004641	97 EB4E	jmp \$00464de7		0012F370	5AD76	7E6	s z,	'gæ	ŝ
Custon	DrawTreeView.pas.13	30: if Tile1.Checked then //tile 1	o:	0012F36C	00000	000) •••		
004641	99 8B83C0030000	mov eax,[ebx+\$000003c0]		0012F368	0012F	590		Öœ	
004641	9F 80783800	cmp byte ptr [eax+\$38],\$00		0012F364	00B63	ESC	9	[>P	
004641	A3 7426	jz \$00464dcb		0012F360	0000B	C4E	1.1	HIN	
Custon	DrawTreeView.pas.13	32: Brush.Bitmap := Image1.Picture	3	0012F35C	00125	414		.0.	
004641	A5 8B83EC030000	mov eax,[ebx+\$000003ec]		0012F358	00462	E54	• •	.T	
004641	AB 8B8098010000	mov eax,[eax+\$00000198]		00121354	0012F	768		÷n	
004641	B1 E82EE1FBFF	call TPicture.GetBitmap		0012F350	00128	414		.0.	
004641	B6 8BD0	mov edx,eax		00121340	00462	A29	• •	(7)	
004641	B8 8B4614	mov eax,[esi+\$14]		00125348	00125	359		.01 	
004641	BB E804BAFBFF	call TBrush.SetBitmap		00125344	00121	414		. 0 .	
Custon	DrawTreeView.pas.13	<pre>33: FillRect(ARect);</pre>		00121340	00462	541		18K	
004641	C0 8BD7	mov edx,edi		00128330	00125	340		OII	
004641	C2 8BC6	mov eax,esi		00125338	00462	524	!	187	
004641	C4 E813BEFBFF	call TCanvas.FillRect		00121334	00456	386		.a	
004641	C9 EB1C	jmp \$00464de7		00121330	00563	LSU		1>P	
Custon	DrawTreeView.pas.13	35: StretchDraw(ARect, Image1.Pict	3	00121320	00008	0.91		+ain	
004641	CB 8B83EC030000	mov eax,[ebx+\$000003ec]		00125328	00000	1001			
00410000	64 FF 30 64 89 20	8B düüdte /		00125324	00128	325		0(
00410000	3 8D 45 F0 F8 AP F6	FF Ó FÁàzöü	8	00121320	00463	9A1		51	
00410010	F PD 45 FO FF 15 PO	ng o rao rú		00121310	00128	414		.0.	
00410010	6 00 8D 55 F0 8B C6	FR F HALFÀ		00121318	00000	PCC			
00410020	C 00 00 00 33 C0 57	50 i 317V		00121314	00863	£50		125	
00110020 1	C 00 00 00 35 C0 5A	55 1 SALI	9 🖻	00121310	00128	304	100	.00	12

図 45- CPU ビューによりアプリケーションの低レベル情報をすべて表示

統合ユニットテスト

テストドリブン開発(TDD)は、ここ数年でより一般的になってきており、ユニットテストは、その 主要な方法論のひとつです。RAD Studio 2010 では、DUnit(VCL 言語拡張により Delphi コードでユ ニットテストを実行する Delphi ベースのライブラリ)によるユニットテストの作成と実行プロセス の自動化機能を提供しています。開発者は、特定のライブラリコードを IDE で処理するように指定 できます。RAD Studio は、テストプロジェクトを作成し、選択したクラスの public および published メソッドのための空のテストケースを生成します。テストされるべき新しいメソッドが追加された場 合には、ユニットテストウィザードは、この新しいメソッドを認識して、これらのメソッドのための 新しいスタブメソッドを追加します。そのため、開発者はすばやく簡単に、コードライブラリのため のユニットテストを作成することができます。



図 46 - ユニットテストウィザードがコードライブラリのユニットテストセットを作成するために情報を収集

リファクタリング

リファクタリングは、既存のコードを、その動作を変更することなく内部構造を変更し、再構成する 機能です。プログラミングが発明された時から、マニュアル操作によるリファクタリングは行われて きました。しかし、近年になって、初めて定型化され自動化されました。RAD Studio 2010 は、コー ドのリファクタリングを自動化する強力なツールセットを提供しています。

例えば、パラメータの変更リファクタリングでは、メソッド宣言とその実装部をすばやく修正することができます。この機能は、既存のパラメータの属性を変更するほか、パラメータの追加と削除をサ ポートしています。

この機能を使うには、コードエディタでメソッドや関数、プロシージャを選択して(宣言部、実装部のいずれでもかまいません)、[リファクタリング | パラメータの変更] メニューを選択します。パラメータの変更ダイアログボックス(図 47)で、変更する情報を指定します。

🛃 Change	Parameters									
Class WindowPainClass.TWindowPain Method WaitForWindow Parameters										
Status	Parameter Name	Data Type	Default Value	Туре	Add					
Existing Existing Existing Existing Existing Existing	WindowTitle WindowClass TimeOut NumChildWindows ChildWindowsComp NumMenuItems MenuItemsCompare	string string Cardinal Integer CompareType Integer CompareType		value value value value value value value	Edit Remove Move Up Move Down					
			ок	Cancel	Help					

図 47 – パラメータの変更ダイアログ

さらに、Delphiには、たくさんのパターンに関連するリファクタリング機能があります。RAD Studioに搭載されたすべてのリファクタリング機能の概要を以下にリストします。

リファクタリング名	説明
メンバーの移動	静的なメソッドをあるクラスから別のクラスに移動する
インターフェイスの抽出	選択したあるクラスのメソッドからインターフェイスを作成し、指定し たクラスによって、そのインターフェイスの実装を宣言する
スーパークラスの抽出	選択したクラスメンバーを、そのクラスの親クラスとなる新しいクラス に移動する
メンバーのプルアップ	選択したメンバーを選択した親クラスに移動する
メンバーのプッシュダウン	選択したメンバーを選択した継承クラスに移動する
安全な削除	選択した項目をアプリケーション内で利用されてないことを確認して削 除する
変数のインライン化	一時的な変数を実変数からインライン変数に変換する
フィールドの導入	既存のローカル変数を指定したクラスのフィールドに変える
変数の導入	リテラル式に新しい変数名を割り当て変数宣言に変換します。
名前の変更	指定した識別子名をプロジェクト全体にわたって変更する
変数の宣言	コードで使われているがまだ宣言されていない変数をローカル変数とし て宣言する
フィールドの宣言	選択された宣言されていない識別子を、指定したクラスのフィールドと して宣言する
メソッド抽出	選択されたコードを新しいメソッドとして作成し、必要なパラメータを

	渡すように記述する
リソース文字列の抽出	選択した文字列から新しいリソース文字列を作成し、文字列リテラルを リソース文字列を使う形式に変更する
パラメータの変更	指定したメソッドのパラメータを追加、修正、削除する

これらの多くのリファクタリング機能を有効にするには、プロジェクトとのモデリングサポートを有効にする必要があります。

OBJECT PASCAL 言語

Object Pascal 言語は、学術的な分野をルーツとして発展してきました。当初の教育目的の言語から は大きく飛躍し、現在では、Object Pascal は、フル機能のオブジェクト指向開発言語となっていま す。Object Pascal は、最新の言語設計のトレンドと機能を取り込んできました。実際、Object Pascal は、構造化例外処理とプロパティを初めて実装したメインストリームの開発言語でした。可 読性の高い文法とシンプルな言語構造により、Delphi は、学習が容易でありながら、強力な言語機能 を利用可能な言語として定評があります。

Object Pascal には、プロシージャとオブジェクト指向のコードの双方を記述する能力があります。2 種類のコーディング手法をスムースに融合させて、開発者は両方の長所を利用できます。開発者は、 VCL を継承した独自のクラスを作成できると同時に、標準的なプロシージャ的なコードも作成でき ます。Delphi は、幅広いレンジのコーディング技法をカバーしています。

おそらく、Object Pascal 言語についてのもっともよい情報は、Marco Cantú 氏の <u>Essential Pascal</u> (PDF) でしょう。

VCL (VISUAL COMPONENT LIBRARY)

VCL(Visual Component Library)は、フル機能の拡張可能で強力なコンポーネントベースのクラス ライブラリです。その土台から最上部まですべてコンポーネントテクノロジーによって設計された VCLは、ビジュアルデザイナによる開発をサポートし、1995年から活躍してきました。その間、い くつものプラットフォームの変化を経験し、Win16からWin32へ、そして.NETにも拡張しました。

データベースアクセス

Delphiは、シンプルながら強力なデータアクセスモデルを提供します。VCL クラスをベースとした TDataSet と呼ばれるデータセットアクセスは、コードによる開発と比較して極めてシンプルで、ビ ジュアルです。Connection コンポーネントを簡単に Dataset コンポーネントと結びつけることで、 テーブルのデータは容易に VCL のデータ対応コントロール上で利用可能になります。実際、データ を表示するために、一切のコーディングを必要としません。

さらに、データフィールドは、自動的に VCL の TField を継承したクラスにバインドされるので、開発者は、次のような分かりやすいコードで、データを取り出したり変更することができます。

CustomerTableCustomerNameField.AsString := 'Gary Johnson';

この方法により、開発者は、ビジュアルでもコードでも、労力をかけずに簡単にデータにアクセスすることができます。VCL コンポーネントは、データのアクセスや操作を行う数多くの機能を提供しているので、開発者はほとんどコードを記述する必要がありません。

WEB 開発

VCL FOR THE WEB

RAD Studio 2010 で Web アプリケーションを構築するための主要なツールは VCL for the Web と呼ばれています。AtoZed Softwareの IntraWeb テクノロジーをベースとした VCL for the Web は、標準的な Delphi クライアントアプリケーションを開発するのと同様の手法で、Web アプリケーションを構築可能にします。VCL for the Web は、コンポーネントベースのテクノロジーで、VCL for Win32と非常に類似した機能を提供します。開発者は、通常の VCL フォームと同じように、コンポーネントをフォームにドロップして、プロパティを設定して、Web ページを設計します。しかし、アプリケーションは、ブラウザで実行されます。

VCL for the Web では、さらに、AJAX テクノロジーを自動的に利用できます。可能であれば、VCL for the Web は、クライアントサイドプロセスを活用するためにアプリケーション中に必要な AJAX コードを自動的に呼び出します。設計時に非同期イベントにコードを設定するだけで、開発者は、 Object Pascal のみのコードで、AJAX ベースのイベントを作成できます。

多層開発

VCL では、DataSnap と呼ばれる強力な多層アプリケーション開発アーキテクチャを提供しています。 さまざまな通信プロトコルをコンポーネントセットとして提供しているので、VCL 開発者はわずか な労力で簡単に、データを供給する中間層のアプリケーションサーバーと、このアプリケーションサ ーバーからデータを取得し更新するシンクライアントアプリケーションを構築できます。

TRemoteDataModule クラスを活用して、開発者は簡単に HTTP、DCOM、あるいはソケット通信 を使ったリモートアクセス機能を提供するデータプロバイダコントロールのインターフェイスをエク スポートできます。クライアントアプリケーションは、リモートデータへの参照を取得して、 TClientDataSet を継承したデータセットにインポートできます。一度 TClientDataSet へのイ ンポートが確立されれば、クライアントアプリケーションは、データの作成、読み込み、更新、削除 (CRUD) といったすべての操作を、TClientDataSet が捕捉しすべての変更をキャッシュして、 実行可能になります。クライアントアプリケーションは、完全にサーバーから切り離して、後で再接 続する形式でも使用できます(いわゆる「ブリーフケースモデル」)。再接続が行われると、 TClientDataSet は、すべての変更をサーバーと調整するコードを実行します。

追加情報

DELPHI で作られたアプリケーション

Delphiは、容易にネイティブアプリケーション開発が実現できるので、ISVの開発プラットフォーム として広く採用されています。以下に列挙した有名なアプリケーションパッケージも Delphi で作ら れています。

- <u>NewsGator</u>社の <u>FeedDemon</u>
- <u>Skype</u>のWindowsクライアント
- Scooter Software 社の <u>BeyondCompare</u>
- The PortableApps Project
- <u>FL Studio</u> デジタルオーディオワークステーション
- <u>AutomatedQA</u>社の <u>TestComplete</u> および <u>AQTime</u>
- VSoft Technologies 社の <u>FinalBuilder</u>

- <u>InstallAware</u> インストールツール
- <u>MediaMonkey</u> 音楽管理ツール
- <u>Copernic Desktop Search</u>

Delphi によって作られたその他のアプリケーションについての完全なリストは、<u>public Delphi Wiki</u> に掲載されています。

また、RAD Studio 2010 も Delphi 自身によって作成されています。<u>Delphi for PHP</u> や <u>Blackfish SQL</u> といった他の CodeGear ツールも Delphi によって作られています。



エンバカデロ・テクノロジーズについて

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300万以上のコミュニティが、エンバカデロの Delphi®、C++Builder®、JBuilder®といった CodeGea™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 をご覧ください。