

엠바카데로 델파이와 Windows 개발 현황



An Idera, Inc. Company

데이비드 아이 (David Intersimone "David I")

1. 소개	6
2. Windows 개발 현황	8
3. 왜 델파이인가?	10
3.1 델파이 이전	10
3.2 델파이 1과 2	11
3.3 델파이 3 ~ 10	12
3.4 델파이의 진화는 계속된다	13
4. Windows 애플리케이션 프로젝트 유형	16
5. Windows 개발도구 비교	18
5.1 엠바카데로 델파이 (델파이)	18
5.2 마이크로소프트 비주얼 스튜디오	19
5.3 젯브레인 인텔리J IDEA	19
5.4 이클립스 개발환경	19
5.5 마이크로소프트 자마린 (Xamarin)	20
5.6 비주얼 스튜디오 코드	20
5.7 도구 비교표	20
6. Windows 개발자를 위한 주요 기능과 이점	24
6.1 파이어닥 (FireDAC)	25
6.2 비주얼 라이브바인딩	27
6.3 양방향 비주얼 도구	28
6.4 스타일 및 테마	29
6.5 Windows API 및 WinRT 유닛	31
6.6 병렬 프로그래밍 라이브러리 (PPL)	31
6.7 Windows 10 컴포넌트, 고해상도 DPI 및 다중 모니터 지원	32
6.7.1 Windows 10용 컴포넌트	32
6.7.2 Windows 10 고해상도 DPI 지원	33
6.7.3 VCL 및 FMX에서 멀티 디스플레이 지원	33
6.8 RAD서버	34
6.9 애플테더링 (App Tethering)	35

6.10	블루투스, 비콘, 비콘펜싱 및 사물인터넷 (IoT)	36
6.11	인터베이스 데이터베이스	40
6.12	엔터프라이즈 커넥터	40
7.	델파이 코드 생산성 및 성능	42
7.1	생산성	42
7.2	성능	44
8.	기타 기능, 장점 및 유용한 자료	46
8.1	크로스 플랫폼	46
8.2	델파이 애플리케이션 가상화 및 배포 방법	46
8.2.1	Cybele Software Thinfinity® VirtualUI	46
8.2.2	아마존 AppStream 2.0	47
8.2.3	델파이 배포 관리자	47
8.2.4	써드파티 설치 도구	48
8.2.5	온라인 스토어에 애플리케이션 제출	48
8.3	기술 파트너 생태계	49
8.4	교육 – 강의, 도움말(Help), 코드 샘플	51
8.4.1	LearnDelphi.org	51
8.4.2	엠바카데로 아카데미	51
8.4.3	개발자 스킬 스프린트	52
8.4.4	유튜브 – 웨비나 다시 보기 및 기타 비디오	52
8.4.5	How-To 비디오	52
8.4.6	엠바카데로 개발자 블로그	52
8.4.7	엠바카데로 소셜미디어	52
8.4.8	델파이로 만들어진 프로그램 사례	53
8.4.9	코드 예제, 튜토리얼 및 샘플	53
8.4.10	RAD스튜디오 개발환경 도구 API (Tools API)	53
8.4.11	갯잇 패키지 매니저	54
8.4.12	델파이용 깃허브 리포지토리	54
8.4.13	스택오버플로우 Q&A	54
8.4.14	플랫폼 지원 확인 페이지	54
9.	비디오 – Windows 10과 VCL 관련	56
9.1	Windows 10 관련 앱 현대화 비디오 목록과 링크	56
9.2	델파이 개발환경 기능과 생산성 관련 비디오 목록과 링크	57

10. 델파이, 25년 혁신의 기록	59
델파이 1 – 1995.2.15	59
델파이 2 – 1996.2.10	59
델파이 3 – 1997.8.5	60
델파이 4 – 1998.6.17	60
델파이 5 – 1999.8.10	60
델파이 6 – 2001.5.21	60
델파이 7 – 2002.8.9	61
델파이 8 – 2003.12.22	61
델파이 2005 – 2004.10.12	61
델파이 2006 – 2005.11.23	61
델파이 2007 – 2007.3.16	62
델파이 2009 – 2008.8.25	62
델파이 2010 – 2009.8.15	62
델파이 XE – 2010.8.30	63
델파이 XE2 – 2011.9.2	63
델파이 XE3 – 2012.9.3	64
델파이 XE4 – 2013.4.22	64
델파이 XE5 – 2013.9.11	64
델파이 XE6 – 2014.4.15	65
델파이 XE7 – 2014.9.2	65
델파이 XE8 – 2015.4.7	66
델파이 10 시애틀 – 2015.8.31	66
델파이 10.1 베를린 – 2016.4.16	66
델파이 10.2 도쿄 – 2017.3.22	67
델파이 10.3 리오 – 2018.11.21	67
델파이 10.3.1 리오 – 2019.2.14	67
델파이 10.3.2 리오 – 2019.7.18	68
델파이 10.3.3 리오 – 2019.11.21	68
델파이 10.4 시드니 – 2020.5.26	68
저자 소개	70



소개

1. 소개

마이크로소프트 Windows가 여전히 데스크톱 컴퓨팅을 주도하고 있는 운영체제라는 점은 각종 분석 보고서와 컴퓨터 사용 추적 회사들이 증명하고 있다. 2020년 1월 [StatCounter.com](https://www.statcounter.com) 보고서에 따르면, 데스크톱 운영체제 시장 점유율에서 마이크로소프트 Windows는 77.7%를 차지하고 있다. Windows가 세상에 나오고 나서 지금까지 수백만 개의 Windows용 응용 프로그램이 개발되었다. 개발에 사용된 도구와 프로그래밍 언어 또한 매우 다양하다. 이 백서는 Windows용 개발 언어와 도구를 사용해본 적이 있는 개발자를 위해 작성되었다. 델파이 현재 버전 또는 이전 버전을 사용하고 있는 개발자 뿐만 아니라 Windows 개발을 처음 접하는 개발자에게도 도움이 될 것이다.

Windows 개발 현황

2.Windows 개발 현황

Windows 1 초기 데모 버전을 내가 처음 본 것은 빌 게이츠가 참석한 회의장에서였다. 당시 나는 메리셀(Merisel)의 전신인 소프트셀(Softsel Computer Products)에 재직 중이었다. 내가 본 Windows 1 초기 데모 버전은 특이하게도 사용되지 않는 화면 영역에서 애플리케이션이 열렸는데 (제록스의 스타, 애플의 리사, 매킨토시와는 달리) 마치 타일처럼 보였다. 그래서 나는 Windows 제품 책임자에게 이 타일 방식에 대한 질문을 했더니, 빌 게이츠가 나서서 각 윈도우 창 1개당 애플리케이션이 1개씩 작동하고 있다고 말해주었던 기억이 난다. (*역자 주, 이 새 운영체제에서는 애플리케이션이 윈도우 안에서 작동하고, 이런 윈도우가 여러개이므로 윈도우의 영문 복수형인 Windows가 이름으로 사용됨)

Windows 1과 2가 출시되는 동안에도, 개발자들은 여전히 DOS와 텍스트 콘솔용 애플리케이션을 만들고 있었다. IBM이 마이크로소프트와 공동으로 DOS를 대체할 GUI 기반의 운영체제인 OS/2를 만들던 시기였다. OS/2가 출시되던 중에도, 마이크로소프트는 Windows 운영체제 개발을 계속하였고 Windows 3.0을 발표했는데, Windows 개발자들이 “쓸만한” 첫 버전이었다고 기억한다. 초창기의 Windows 애플리케이션 개발은 C, 파스칼, 베이직 등등의 언어를 사용하여 Windows API를 직접 호출하는 방식이었다. 개발자 중에는 당시에 간단한 Windows 애플리케이션 하나를 만들 때 사용된 C 코드 100줄을 아직도 기억하는 사람이 있을 것이다.



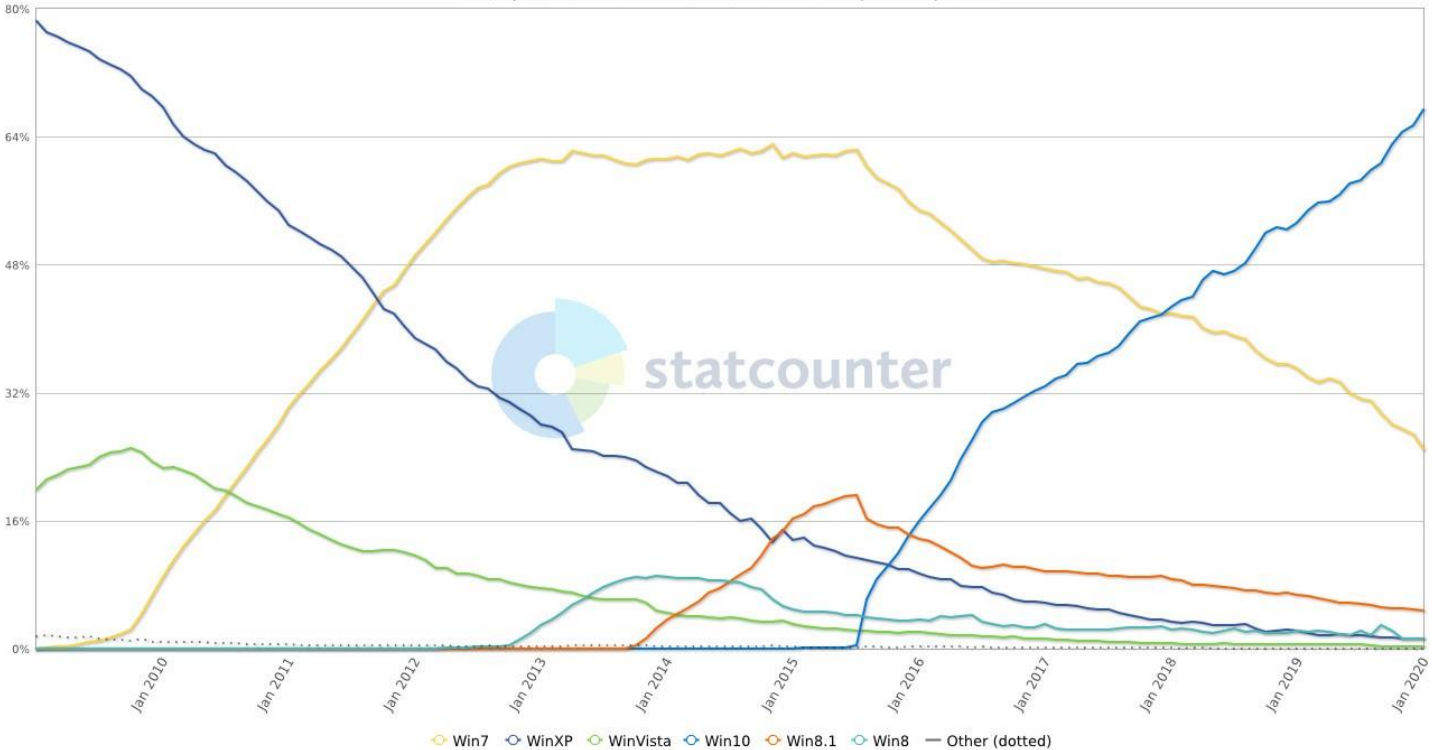
이후 C++, 오브젝트 파스칼, 액터(Actor) 등 객체 지향 언어가 두각을 나타내면서 Windows 애플리케이션 개발에도 새 물결이 일어나고 Windows 프로그래밍은 보다 간단해졌다. 개발자 대부분이 여전히 Windows API를 직접 사용하고 있었지만, Windows API를 캡슐화하고 추상화 수준을 더 높여서 여러 Windows 버전을 지원할 수 있도록 하는 새 클래스와 컴포넌트 라이브러리가 부각되기 시작했다.

데스크톱 컴퓨터에서 Windows가 더욱 널리 사용되면서, 개발자들은 자신들이 만든 애플리케이션이 Windows 여러 버전에서 작동되도록 만들어야 했다. 시간이 흐르면서 마이크로소프트는 오래된 버전 지원을 공식적으로 중단해 나가기 시작했다. 가장 최근에 “공식적으로” 지원이 중단된 버전은 2020년 1월 14일에 중단된 Windows 7이다. 하지만 여전히 개발자들은 Windows 7, 8.x와 10 등 널리 사용되는 몇 가지 버전을 지원해야만 한다.

StatCounter.com 의 [전세계 데스크톱 Windows 버전 시장 점유율](#) 차트는 (2009년 1월에서 2020년 1월까지) 지난 11년간의 변화를 보여준다. 이 차트에 따르면 현재 Windows 7과 Windows 10이 가장 많이 사용되고 있다.



왜 델파이인가 ?



3. 왜 델파이인가?

애플리케이션을 만들 때 사용하는 플랫폼, 개발도구, 프로그래밍 언어, 라이브러리, 아키텍처 등에 대한 선택의 폭은 넓다. 그런데 굳이 개발자가 델파이를 사용해야 하는 이유는 무엇일까? 델파이로 어떤 종류의 애플리케이션을 만들 수 있을까? 사용자와 고객의 까다로운 요구 사항과 일정을 충족시킬 수 있을까? 지금껏 델파이가 이룩한 주요 기능과 기술 혁신을 살펴보면, 델파이가 어떻게 비즈니스와 고객 요구를 성공적으로 반영하고 첨단 기술과 최신 소프트웨어 엔지니어링을 어떻게 수용해가는지를 이해할 수 있다.

3.1 델파이 이전

델파이 이전에는 비객체 언어와 객체 지향 프로그래밍 언어로 Windows 애플리케이션을 다양하게 개발했었다. 엠바카데로 개발도구와 라이브러리들은 DOS와 Windows 애플리케이션 개발이 API를 직접 사용하던 방식에서 클래스 라이브러리를 사용하는 방식으로 전환될 수 있도록 지원했다.

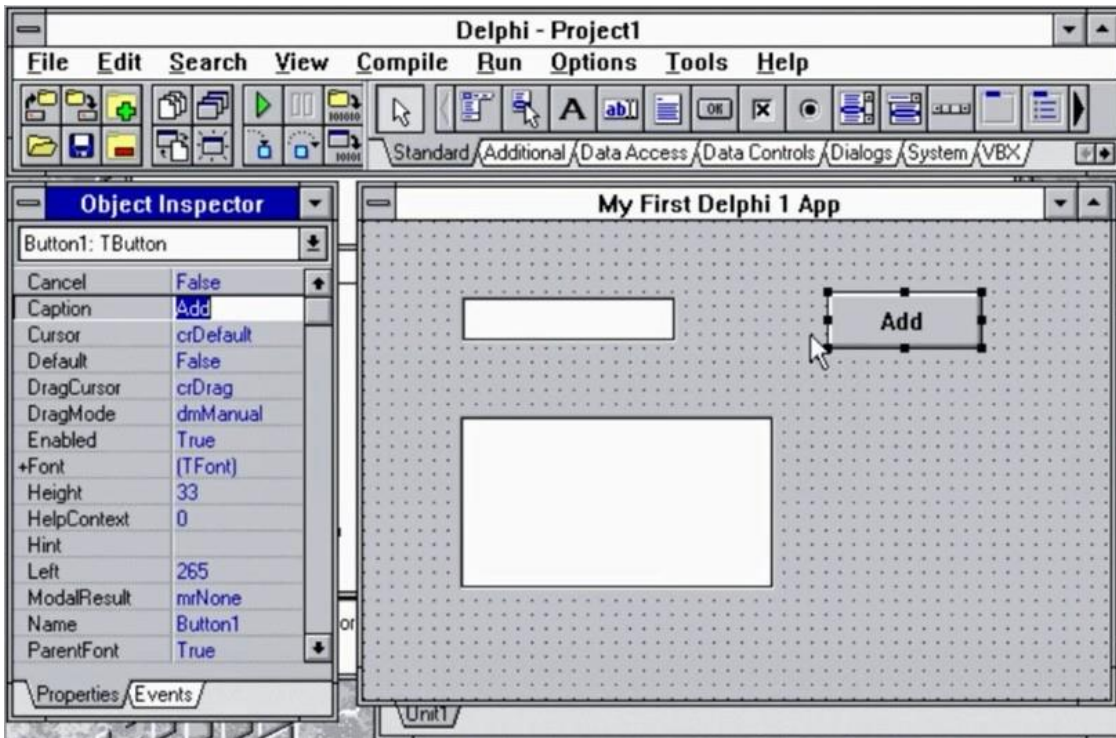
- DOS용 터보 파스칼과 터보 비전 (Turbo Vision)
- Windows용 터보 파스칼과 오브젝트 Windows 라이브러리 (OWL)

이 개발 솔루션들이 개발자의 생산성을 향상시켰지만, 마이크로소프트 Windows를 더 빠르게 성공시키기 위해서는 차세대 기술과 구현이 필요했다. 그 결과 비주얼 프로그래밍, 컴파일러 기술, 데이터베이스 프로그래밍 모두가 자연스럽게 개발 과정에 통합된 최첨단 Windows 애플리케이션 개발 솔루션이 나오게 되었다.

3.2 델파이 1과 2

1995년 2월 14일 샌프란시스코의 모스콘 컨벤션 센터에서 열린 소프트웨어 개발 컨퍼런스에서 2,000명이 넘는 개발자와 PC 산업 전문가들에게 델파이 1이 공개되었다. 그날 저녁 델파이 수석 아키텍트인 앤더스 헤즐스버그(Anders Hejlsberg)는 시연을 마치고 참석자들로부터 기립박수를 받았다.

앤더스 헤즐스버그가 당시에 한 말: “델파이는 비주얼 프로그래밍, 컴파일러 기술, 데이터베이스 프로그래밍을 상자 하나에 모두 담은 최초의 제품입니다. 델파이의 기본 사상은 개방성과 확장성입니다. 예를 들어, 델파이에는 엑스퍼트 기능이 들어있는데, 미리 만들어놓은 엑스퍼트들의 세트를 단순히 추가하기만 한 것이 아니라 여러분이 엑스퍼트를 직접 만들어서 설치할 수도 있도록 했습니다. 템플릿 역시 표준 템플릿 몇 가지만 추가한 것이 아니라 여러분이 직접 만든 템플릿들을 설치할 수 있도록 해 놓았습니다. 델파이는 32비트까지 커버할 수 있도록 기초가 설계되어 있습니다. 여러분이 작성한 소스 코드를 고치지 않고도 32비트로 포팅된다는 점에 대해 우리는 이미 내부 검증을 마쳤습니다. 우리가 앞으로 끌어안아야 하는 새 기술들 역시 준비하고 있습니다. 예를 들어 OLE2, OCX 컨트롤, Windows 95에 들어가게 될 모든 새 API가 델파이에도 반영될 것입니다. 델파이를 사용하여 개발하는 것은 재미있습니다. 이 점이 가장 신나는 점이죠. 우리도 델파이를 사용하는 것이 정말 재미있습니다. 우리는 델파이를 가지고 델파이를 만들었습니다. 즉, 지난 2년 내내 델파이를 사용했습니다. 그동안 우리는 여러분이 겪게 될 수도 있었던 모든 문제를 미리 경험했고 또 해결해 놓았습니다. 전문가 수준의 화면을 가진 애플리케이션을 이제 얼마나 빨리 만들 수 있는지를 확인하면 놀랄 것입니다. 컨트롤 몇 개를 폼에 올리고 나서 코드 몇 줄을 작성하면 끝입니다.”



델파이 1은 Windows 3 16비트 API를 지원했으며 단일 통합개발환경, 16비트 네이티브 코드 컴파일러, 비주얼 컴포넌트 라이브러리 (VCL, Windows GUI 인터페이스를 다루는 네이티브 코드 오브젝트들이 모여있다), 디버깅, 볼랜드 데이터베이스 엔진 (BDE, 공통 컴포넌트 세트를 통해 로컬 데이터베이스와 SQL 데이터베이스 클라이언트/서버 액세스를 지원한다)이 개발환경 안에서 모두 제공되었다.

개발자는 델파이 통합개발환경으로 비주얼 컴포넌트 뿐만 아니라 외형이 없는 기능 컴포넌트를 폼 위에 올려놓고, 올려둔 컴포넌트의 프로퍼티를 오브젝트 인스펙터에서 설정한 후, 컴포넌트의 이벤트 중 원하는 것을 선택하고 그곳에 알고리즘과 비즈니스 로직을 구현하는 코드를 추가했다. 델파이 통합개발환경에서 컴포넌트와 프로퍼티를 가지고 할 수 있는 것이라면 무엇이든 코드를 사용해서도 구현할 수 있었다. 폼 디자이너에서는 애플리케이션의 UI 화면 디자인의 실제 모습을 볼 수 있었고, 텍스트 모드에는 해당 코드가 표시되었다. 비주얼 화면이나 코드 중 어느 곳에서도 디자인을 변경할 수 있었다. 델파이를 통해 개발자는 개발 화면에서 난생 처음으로 실제 데이터를 볼 수 있게 되었다. (파라독스, 디베이스 등) 로컬 데이터베이스와 다양한 SQL 클라이언트 서버 데이터베이스가 연결되었고 실제 데이터가 개발환경 안으로 전달되었다.

델파이 1이 개발되는 동안, 마이크로소프트에서는 Windows의 차기 버전을 준비하고 있었다. 위에서 앤더스가 이야기했듯이 델파이 팀은 16비트와 32비트 Windows 지원을 동시에 테스트했는데, 16비트 Windows용으로 작성된 코드가 32비트 Windows용으로 컴파일 될 때에도 코드를 수정할 필요가 없도록 하기 위해서였다. 나는 마이크로소프트 회사 잔디밭에서 Windows 95가 출시될 때, 그곳에서 델파이 2 버전의 초기 데모 버전을 시연했다.

지금까지도 여전히 VCL은 컴포넌트 기반 아키텍처를 유지하고 있다. 또한 개발자가 자신의 코드 자산을 버리지 않고 지키면서도 Windows 후속 버전들을 함께 지원하도록 하는 접근 방식 역시 유지하고 있다. 그러기 위해 델파이는 끊임없이 진화해 왔다.

3.3 델파이 3 ~ 10

델파이 3과 4에서는 새 언어 기능, 멀티-티어 개발, Windows 98 지원이 추가되었다. 이때 추가된 새 언어 기능은 인터페이스, 메소드 오버로딩, 동적 배열, 컴포넌트 패키지 등이다.

델파이 5, 6, 7에서는 인터넷과 웹 기능이 소개되었고 프레임, XML 지원, SOAP 웹서비스, UML, Windows XP 지원이 추가되었다.

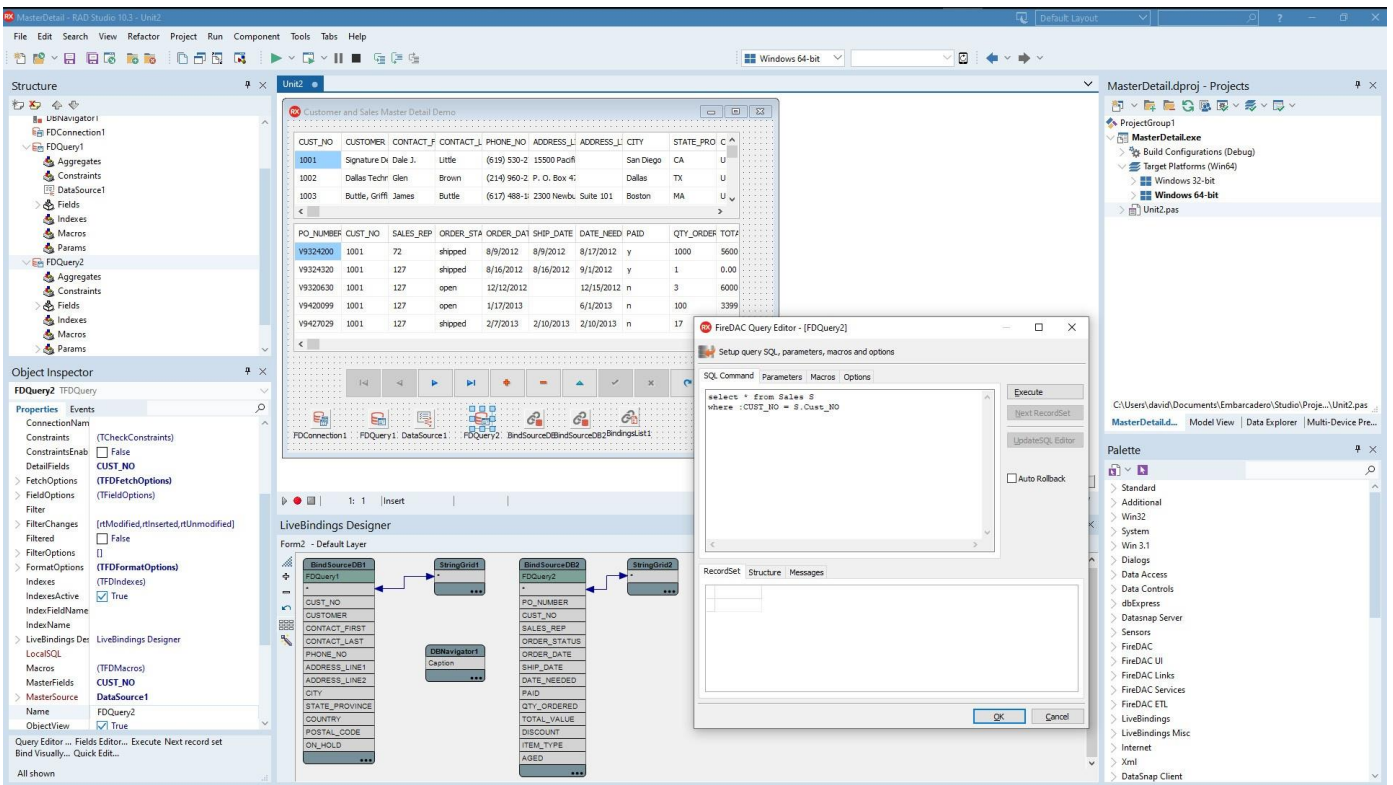
델파이 2005에서 2010까지는 인라인, FOR..IN, 제네릭스, 익명 메소드, 애트리뷰트, 터치/제스처, 다중 유닛 네임 스페이스, 연산자 오버로딩, 정적 메소드, 정적 프로퍼티, 코드 에디터 강화, FastMM 메모리 관리자, 유니코드 지원, 기타 Windows 지원에 필요한 여러 기능들이 추가되었다.

델파이 XE에서 XE8까지는 크로스 플랫폼 개발 (Windows, 맥OS, iOS, 안드로이드)을 위한 파이어몽키 (FMX), 파이어닥(FireDAC), REST 지원, 라이브바인딩, 스타일, 앱테더링(App Tethering), 클라우드/BaaS 지원, 병렬 프로그래밍 라이브러리(PPL), RAD서버 등이 추가되었다.

델파이 10.0 에서 10.3 까지는 인라인 변수 선언, 타입 유추(type inferencing), MongoDB 지원, 블루투스 클래식/LE, 비콘, 사물인터넷(IoT) 컴포넌트, Windows 10 스타일, 64비트 플랫폼 지원 (맥OS, iOS, 안드로이드, 리눅스), Windows 고해상도 DPI 지원, RAD서버 강화 등이 추가되었다.

3.4 델파이의 진화는 계속된다

델파이와 VCL은 진화를 거듭하며 Windows 운영체제와 Windows API의 새 기능과 변경 사항을 지원해오고 있다. Windows에 적용되는 많은 기술들은 (iOS, 맥OS, 안드로이드, 리눅스와 같은) Windows가 아닌 플랫폼에도 그대로 활용된다.



델파이의 오브젝트 파스칼 언어 역시 개발환경과 함께 지속 발전하면서 RAD스튜디오에 반영되었고, SQL 데이터베이스와 비SQL 데이터베이스, (웹, 클라우드, 멀티-티어, 마이크로 서비스 등) 새로운 아키텍처, 블루투스, 사물인터넷 (IoT) 장비 등에 대한 지원도 계속 진화하고 있다.

델파이에는 파이어몽키 (FMX) 컴포넌트 라이브러리도 포함되어 있다. FMX를 사용하여 크로스 플랫폼 GUI 애플리케이션을 만들면 Win32, Win64, 맥OS, iOS, 안드로이드, 리눅스 등 각 플랫폼별로 네이티브 앱을 배포할 수 있다. 델파이에는 VCL, FMX, 런타임 라이브러리의 소스 코드가 모두 들어있다. 델파이 소스 코드는 컴포넌트와 라이브러리가 어떻게 만들어졌는지를 이해하는데 도움이 될 뿐만 아니라 가장 근본적인 도움말이 되어준다.

“VCL과 FMX 중 어느 컴포넌트 라이브러리를 사용해야 하나요?”라는 질문은 수년동안 반복되고 있다. 정답은 개발하는 애플리케이션이 작동할 운영체제가 무엇인가 그리고 향후에는 어떤 운영체제로 갈

것인가에 따라 달라진다. 만약 Windows 만을 목표로 하는 애플리케이션이라면 VCL이 최고의 선택이다. Windows, 맥OS, 모바일 장비를 지금이나 가까운 미래에 지원할 필요가 있다면 FMX가 최고의 선택이다. 짐 맥키트(Jim McKeeth)가 **블로그 글**에 다음과 같이 잘 설명했다 “VCL은 여러분이 Windows 만을 목표로 하면서 3D, 시각 효과나 파이어몽키의 유연성이 필요 없을 때 사용하세요. FMX는 멀티-플랫폼으로 가려고 하거나 또는 파이어몽키의 (특히 그래픽 처리 관련) 유연성이 필요한 경우 사용하면 됩니다. 두가지 프레임워크 모두 오래동안 유지될 것입니다. 두가지 모두 사용할 수 있다면 각 상황에 맞게 사용할 수 있기 때문에 마음이 한결 편안해질 것입니다.” 만약 VCL로 애플리케이션을 만들었는데 FMX로 옮겨가야 한다면 기술 파트너의 도구인 Mida Converter를 활용해 VCL 애플리케이션을 FMX로 변환하는 것을 고려하는 것도 좋다.

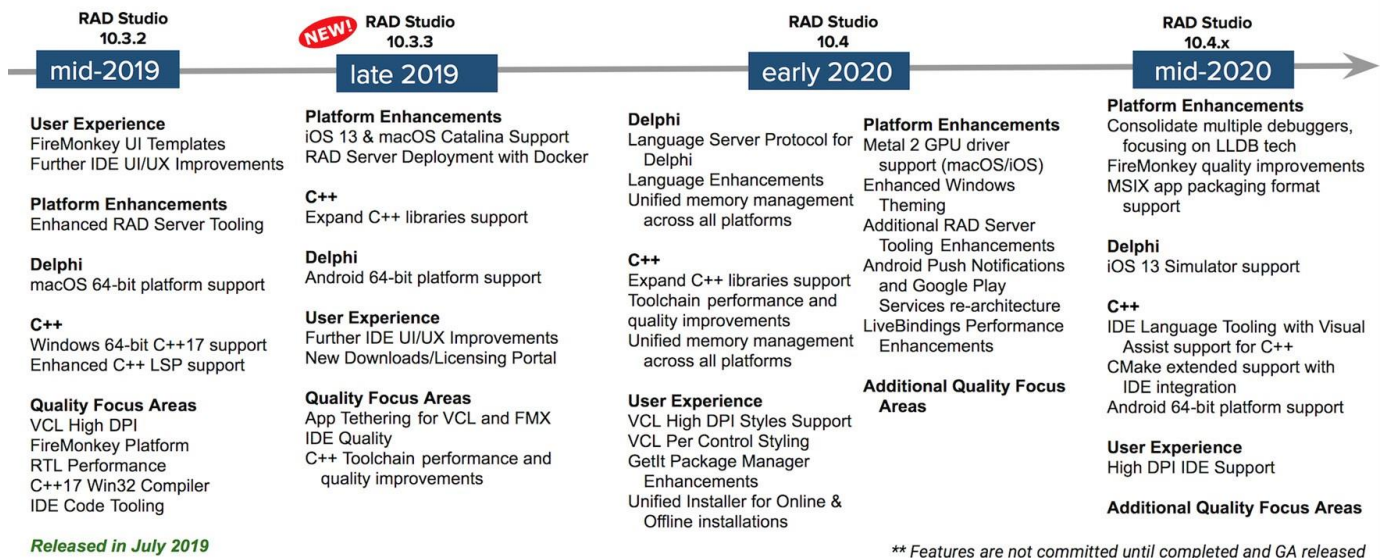
델파이 각 버전별로 주요 기능 추가 사항과 혁신에 대한 목록은 이 백서의 맨 뒤에 있다. 향후 추가 계획은 온라인 **RAD스튜디오 로드맵**에 게시된다.

Updated - August 2019



RAD Studio Roadmap**

RAD Studio CY 2019 / CY 2020 Timeline**



Copyright © Embarcadero Technologies, Inc. 2019



Windows 애플리케이션 프로젝트 유형

Windows 개발도구 비교

5. Windows 개발도구 비교

Windows 애플리케이션을 개발하는 도구는 여러 가지가 있다. 그 중 몇 가지는 다양한 프로그래밍 언어를 지원하는 완전한 통합개발환경이다. 어떤 개발자들은 코드 작성을 위한 프로그래밍 편집기와 애플리케이션 생성용 커맨드 라인 컴파일러 인터페이스를 활용하는 빌드 시스템을 따로 사용하기도 한다. 델파이는 에디터와 빌드 시스템이 통합된 환경을 선호하는 개발자를 위한 통합개발환경 뿐만 아니라 커맨드 라인 컴파일러 역시 별도로 제공한다. 이제부터 설명할 도구 중에는 Windows용 네이티브 코드 애플리케이션을 생성하는 도구도 있고, 소스 코드를 중간 형식 언어로 컴파일하는 도구 또는 JIT 컴파일러를 사용하여 실행 중에 소스 코드가 해석(INTERPRET)되도록 하는 도구도 있다.

현재 개발도구 대부분이 디버깅, 팀 개발, 테스트 자동화, 리팩토링, 배포, 코드 분석, 코드 품질 측정, 프로파일링 등을 통합 지원한다. Windows, 맥 OS, iOS, 안드로이드 등 멀티 플랫폼에서 개발할 수 있는 도구도 일부 있다. 이들 도구들은 각 플랫폼의 네이티브 API에 액세스하거나, 네이티브 API 위에 있는 프레임워크를 사용하거나, 지원 플랫폼 상의 일부 또는 모든 코드를 재사용한다.

크로스 플랫폼 개발도구를 비교할 때 각 도구의 차별성을 판단 기준으로 개발자는 다음 2가지를 알고 싶어한다. '(대체로 UI 와 OS 인터페이스와 관련하여) 플랫폼 네이티브가 얼마나 잘 지원되는가?', 그리고 '(각 플랫폼의 프로세서별로 알맞은 컴파일과 관련하여) 네이티브 머신 코드는 어떻게 생성되는가?' 이다.

다음은 Windows 애플리케이션을 개발할 때 많이 사용되는 개발도구들이다. 참고로 (자바스크립트나 타입스크립트와 같은) 몇몇 언어들은 웹 기반 애플리케이션, 즉 브라우저 또는 Windows 상의 브라우저 컨테이너 안에서 작동하는 애플리케이션을 만들 때 사용하기 좋다.

5.1 엠바카데로 델파이 (델파이)

델파이는 단품으로 구입할 수도 있고, 델파이(오브젝트 파스칼)와 C++ 빌더(C++)가 함께 들어있는 RAD스튜디오로 구입할 수도 있다. 비주얼한 방식으로 Windows용 콘솔 애플리케이션, GUI 데스크톱 애플리케이션, 클라이언트 서버 애플리케이션, 멀티-티어 프로젝트를 설계하고 개발할 수 있다. 개발자는 VCL 컴포넌트와 런타임 라이브러리 (RTL)를 사용한다. 델파이와 C++빌더 모두 동일한 소스 코드를 사용하여 다양한 플랫폼별로 네이티브로 컴파일된 애플리케이션을 생성할 수도 있다. 이 경우에는 파이어몽키 (FMX) 컴포넌트와 RTL을 사용하여 Windows, 맥OS, iOS, 안드로이드용 앱을 개발한다. 델파이로는 인텔 프로세서 기반의 64비트 리눅스용 네이티브 애플리케이션 또한 콘솔용, 데스크톱용, 서버용으로 만들 수 있다.

델파이에는 **네 가지 에디션**이 있다.

- 델파이 커뮤니티(무료) – 프로페셔널과 기능 동일, 하지만 비전문가 및 취미 개발자만 사용 가능
- 델파이 프로페셔널 – 모바일, 데스크톱, 로컬 데이터베이스
- 델파이 엔터프라이즈 – 원격 데이터베이스, 클라이언트 서버, 멀티-티어 솔루션, 리눅스 플랫폼 지원
- 델파이 아키텍트 – 엔터프라이즈 에디션에 아키텍트용 도구들 추가 제공

제품 홈페이지 –

<https://devgear.co.kr/archives/products/delphi> (한글페이지)

5.2 마이크로소프트 비주얼 스튜디오

비주얼 스튜디오는 C++, C#, 비주얼 베이직, 자바스크립트/ 타입 스크립트, 파이썬, F# 언어로 Windows 애플리케이션을 작성하는 도구이다. 네이티브 컴파일러는 C++ 언어용으로만 제공된다. 비주얼 스튜디오는 주로 Windows API, C++의 MFC 프레임워크, 윈폼(WinForms)을 사용하여 콘솔용, 데스크톱용, 서버용 애플리케이션을 만드는 데 사용된다. 비주얼 스튜디오의 C# 언어는 크로스 플랫폼 애플리케이션 개발도 지원한다. Windows, iOS, 안드로이드용 애플리케이션을 만들 수 있으며 닷넷, 윈폼/자마린(Xamarin) 폼을 플랫폼 네이티브 API 인터페이스와 함께 사용한다.

비주얼 스튜디오 에디션(<https://visualstudio.microsoft.com/vs/compare/>)은 다음과 같다.

- 비주얼 스튜디오 커뮤니티 (무료)
- 비주얼 스튜디오 프로페셔널
- 비주얼 스튜디오 엔터프라이즈

제품 홈페이지 – <https://visualstudio.microsoft.com/>

5.3 젯브레인 인텔리J IDEA

인텔리J IDEA는 자바, 코틀린(Kotlin), 자바스크립트/타입 스크립트, 스칼라(Scala), 그루비(Groovy), C/C++(<https://www.jetbrains.com/cpp/>) 기타 프로그래밍 언어를 사용하여 데스크톱, 서버, 웹, 크로스 플랫폼 애플리케이션을 만들 수 있다. 인텔리J IDEA는 자바 개발용으로 주로 사용되어왔다. 멀티-플랫폼에서 작동하는 JVM을 사용하는 애플리케이션을 개발할 때, 자바, 코틀린, 스칼라, 그루비 언어를 사용할 수 있다. 개발환경은 구글의 안드로이드 스튜디오에서 제공되는 것과 동일한 안드로이드 코딩과 내장 툴셋을 담고 있다.

인텔리J IDEA 에디션 (https://www.jetbrains.com/idea/features/editions_comparison_matrix.html):

- 인텔리J IDEA 커뮤니티 에디션 – JVM 및 안드로이드 개발용
- 인텔리J IDEA 얼티미트 – 웹 및 엔터프라이즈용

제품 홈페이지 – <https://www.jetbrains.com/idea/>

5.4 이클립스 개발환경

이클립스 재단에서 제공하는 오픈 소스 통합개발환경이다. 자바, C++, 포트란(FORTRAN), 자바스크립트 언어를 지원한다. 이클립스 통합 환경은 오픈 API를 가지고 있어서 개발도구, 빌드 도구, 라이브러리 등을 통합할 수 있다. 이클립스는 처음 나올 때부터 JVM과 자바 런타임이 있는 장비에서 작동하는 클라이언트와 서버 쪽의 자바 애플리케이션 개발에 중점을 두었다. C++과 포트란 언어 통합이 개선되면서, 네이티브 코드 컴파일러와 디버거가 제공되고 있다.

제품 홈페이지 – <https://www.eclipse.org/eclipseide/>

5.5 마이크로소프트 자마린 (Xamarin)

자마린(Xamarin)은 오픈 소스 크로스 플랫폼 개발도구이다. C#과 오픈 소스, 닷넷 플랫폼을 사용한다. 닷넷 개발 플랫폼을 기반으로 Windows, 안드로이드, iOS, 맥OS, tvOS, watchOS 애플리케이션을 만들 수 있는 도구와 라이브러리를 제공한다.

크로스 플랫폼 애플리케이션을 개발할 때, 개발자는 공유 C# 프로그램/비즈니스 로직, 공유 C# UI 코드, 특정 플랫폼용 C# 코드를 사용한다. 마이크로소프트/자마린은 C# 코드 중 최소 75%가 플랫폼을 넘나들며 공유될 수 있다고 말한다. Xamarin.Forms GUI 애플리케이션은 iOS, 안드로이드, Windows 10 유니버설 플랫폼용 애플리케이션을 개발할 수 있다.

제품 홈페이지 - <https://dotnet.microsoft.com/apps/xamarin>

5.6 비주얼 스튜디오 코드

비주얼 스튜디오 코드는 C++, C#, 파이썬 및 기타 언어를 지원하는 오픈 소스 프로그래머 환경이다. 개발환경은 Windows, 맥OS, 리눅스에서 작동된다. 개발은 프로그래머 에디터가 중심이 된다. 빌드, 디버깅 환경이 제공된다. 오픈 소스 기여자는 특정 프로그래밍 언어와 특정 플랫폼용 비주얼 디자이너를 만들 수 있다.

제품 홈페이지 - <https://code.visualstudio.com/>

5.7 도구 비교표

앞에서 언급한 Windows 개발환경별 기능 요약/비교표이다. 보다 자세한 사항과 최신 버전 정보 등은 각 개발환경의 기능표를 확인하기 바란다.

기능/ 개발도구	RAD 스튜디오	비주얼 스튜디오	인텔리J IDEA	이클립스	자마린	비주얼 스튜디오 코드
에디션	아키텍트, 엔터프라이즈, 프로페셔널, 커뮤니티(무료)	엔터프라이즈, 프로페셔널, 커뮤니티 (무료)	얼티미트, 커뮤니티 (무료)	무료 (오픈소스)	무료 (오픈소스)	무료 (오픈소스)
개발환경	통합개발환경	통합개발환경	통합개발환경	통합개발환경	통합개발환경	소스코드 에디터
Windows 프로젝트 타겟	Win32, Win64	Win32, Win64	JVM	JVM	닷넷	다양함

기능/ 개발도구	RAD 스튜디오	비주얼 스튜디오	인텔리J IDEA	이클립스	자마린	비주얼 스튜디오 코드
언어	오브젝트 파스칼, C++	C++, C#, VB, 타입스크립트/ 자바스크립트, 파이썬, F#	자바, 코틀린, 자바스크립트/ 타입스크립트, 스칼라, 그루비, C/C++, 루비 등	자바, C++, 포트란, 자바스크립트 등	C#	C++, C#, 파이썬, 자바, 고(Go), Rust, VB
네이티브 머신 코드 컴파일러	Y (오브젝트 파스칼과 C++)	Y (C++)	Y (C++)	Y (C++ 과 포트란)	닷넷 바이트 코드	Y (C++)
Windows UI 프레임워크 및 컴포넌트	Y	C++ (MFC)	N	N	N	N
크로스 플랫폼 UI 프레임워크 및 컴포넌트	Y	C# (자마린 품)	N	N	N	N
통합 비주얼 UI 디자이너	Y	N	N	Windows 빌더 프로 플러그인	Y	디자이너 용 개발환경 API
데스크톱용 컴포넌트. SQL과 비SQL 데이터베이스	Y	N	N	N	N	N
웹 서버 애플리케이션 구축	IIS, 아파치	IIS, 애저(Azure) 앱 서비스	아파치 Tomcat, WebLogic, WebSphere	아파치 Tomcat	IIS, ASP. NET	아파치용 플러그인, Jetty용 플러그인
클라우드 통합	애저(Azure), AWS	애저(Azure)	Cloud Foundry, 구글 앱 엔진, OpenShift, AWS	Cloud Foundry, AWS, 아파치	애저(Azure)	애저(Azure)
컴포넌트 및 런타임 라이브러리 소스 코드 제공	Y	Y	Y	Y	닷넷 코어	Y
팀 개발	Subversion, Mercurial, Git	Subversion, Mercurial, Git, VS Team Services (VSTS)	Subversion, Mercurial, Git, Perforce, Team Foundation Server (플러그인)	Subversion, Mercurial, Git	Subversion, Mercurial, Git	Subversion, Mercurial, Git

기능/ 개발도구	RAD 스튜디오	비주얼 스튜디오	인텔리J IDEA	이클립스	자마린	비주얼 스튜디오 코드
정적 코드 분석 및 코드 매트릭스	Y	Y	Y	Y	플러그인	플러그인
유닛 테스트 통합	Y	Y	Y	Y	Y	플러그인
지역화된 개발환경	영어, 프랑스어, 독일어, 일어	영어, 프랑스어, 독일어, 중국어, 일어, 체코어, 이탈리아어, 한국어, 폴란드어, 포르투갈어 (브라질), 러시아어, 스페인어	N	N	N	플러그인 언어 팩
마이크로 서비스 턴키 프레임워크	RAD서버	애저 서비스 Fabric, 닷넷 마이크로서비스	Micronaut, Quarkus, Helidon	플러그인	닷넷 마이크로서비스	N



Windows 개발자를 위한 주요 기능과 이점

6. Windows 개발자를 위한 주요 기능과 이점

처음 나올때부터 지금까지, 델파이는 최고의 컴포넌트 기반 비주얼 개발환경을 제공하는 것에 주력해왔다. 그 중심에는 네이티브 기계어 코드 컴파일, 데이터베이스 액세스, Windows 플랫폼을 위한 광범위한 API와 기술 제공이 있다. 이 장에서는 델파이가 Windows 개발 솔루션들 중 가장 높은 평가를 받고 있는 이유인 주요 기능과 이점들을 살펴본다. 델파이와 C++ 빌더 **중합 기능표**는 엠바카데로 웹사이트에 있다.

델파이는 32비트 및 64비트에 최적화된 컴파일러와 툴체인을 제공한다. 그리고 Windows 7, 8.x 및 10을 포함한 여러 Windows 버전을 지원한다. 소스 코드 호환성이 매우 높기 때문에, 동일한 델파이 코드는 Windows 여러 버전과 인텔 프로세서 여러 버전에서 재사용할 수 있다.

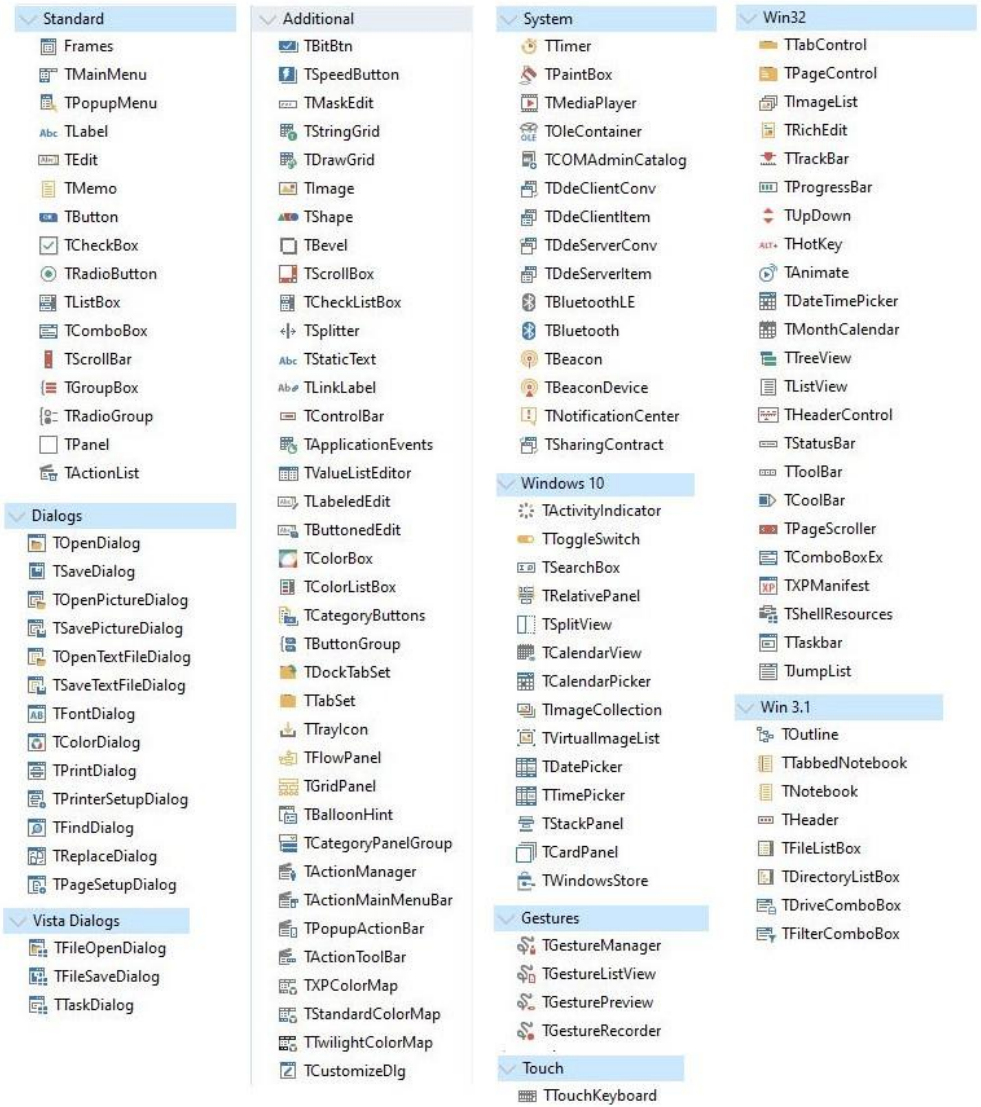
시각적으로 화면에 표현되는 컴포넌트 라이브러리인 VCL (Visual Component Library)과 형태가 없는 기능 컴포넌트 모두 다양한 Windows 버전을 지원하고 32비트, 64비트 프로세서에서 작동한다.

컴포넌트 기반 개발 방식으로 UI, 데이터베이스 액세스, 아키텍처, 기타 비즈니스 로직을 작성한다. 즉 폼 위에 원하는 컴포넌트를 올린 뒤 프로퍼티를 설정하고 꼭 필요한 최소한의 코드만 직접 작성하는 개발 방식이다.

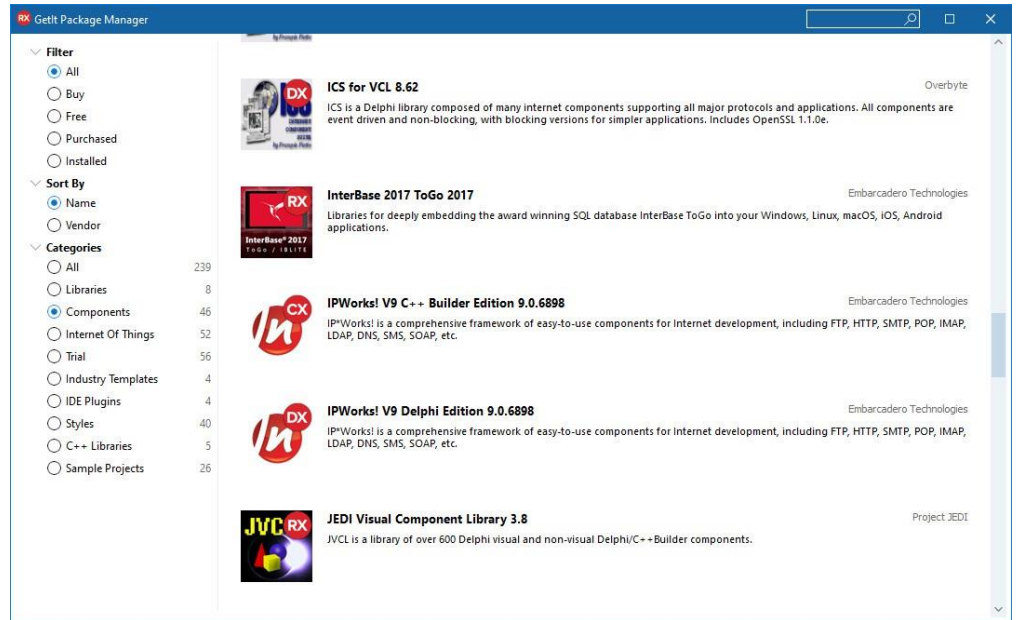
컴포넌트는 개발 시점에 사용할 수 있으며, 실행 중에 생성할 수도 있다. 프로퍼티 설정 역시 개발 시점과 실행 중 모두 가능하다. 컴포넌트는 상속을 통해 확장할 수 있고, 기반 클래스를 바탕으로 새 컴포넌트를 만들 수도 있다. 또한 폭넓은 델파이 개발 생태계에서 제공하는 다양한 유/무료

컴포넌트와 도구들을 쉽게 활용할 수 있어, 비즈니스와 애플리케이션의 요구사항을 충족시킬 수 있다.

델파이에는 UI와 Windows 시스템 기능들을 다루고 구축할 수 있는 수많은 컴포넌트가 들어있다.



개발환경 안에서 깃잇 패키지 매니저 (GetIt Package Manager)를 통해 더 많은 컴포넌트들을 다운로드하고 설치할 수 있다.

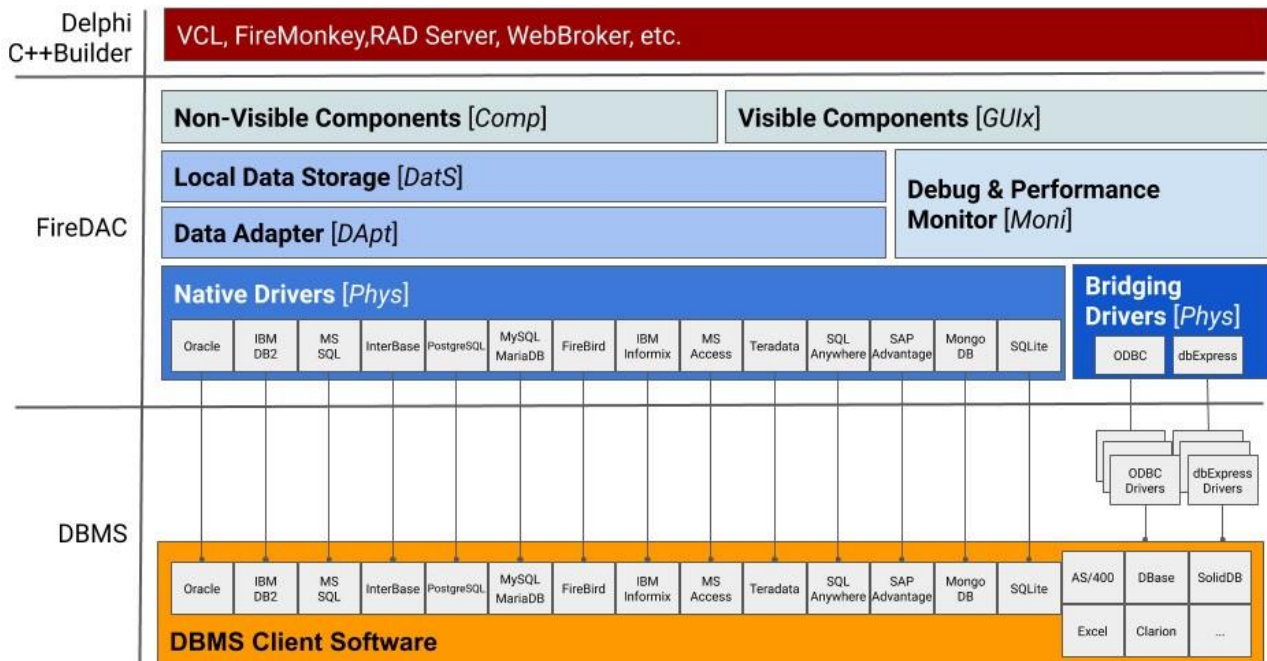


VCL에 대한 더 많은 정보는 엠바카데로 DocWiki에 있다. (아래 링크 참조)
[http://docwiki.embarcadero.com/RADStudio/en/Introducing_the_Visual_Component_Library_\(VCL\)](http://docwiki.embarcadero.com/RADStudio/en/Introducing_the_Visual_Component_Library_(VCL))

6.1 파이어닥 (FireDAC)

파이어닥은 데이터베이스 애플리케이션을 개발할 때 사용된다. Windows와 멀티-디바이스를 모두 지원하는 델파이용 데이터 액세스 컴포넌트 세트가 파이어닥에 포함되어 있다. 파이어닥은 델파이와 데이터베이스를 네이티브 방식으로 그리고 빠른 속도로 연결한다. 인터베이스, SQLite, MySQL, SQL서버, 오라클, PostgreSQL, IBM DB2, SQL Anywhere, 액세스(Access), 파이어버드, 인포믹스, MongoDB 등등을 지원한다.

파이어닥은 아래 다이어그램과 같이 유연하고 강력하며 확장성이 높은 아키텍처로 되어있다.

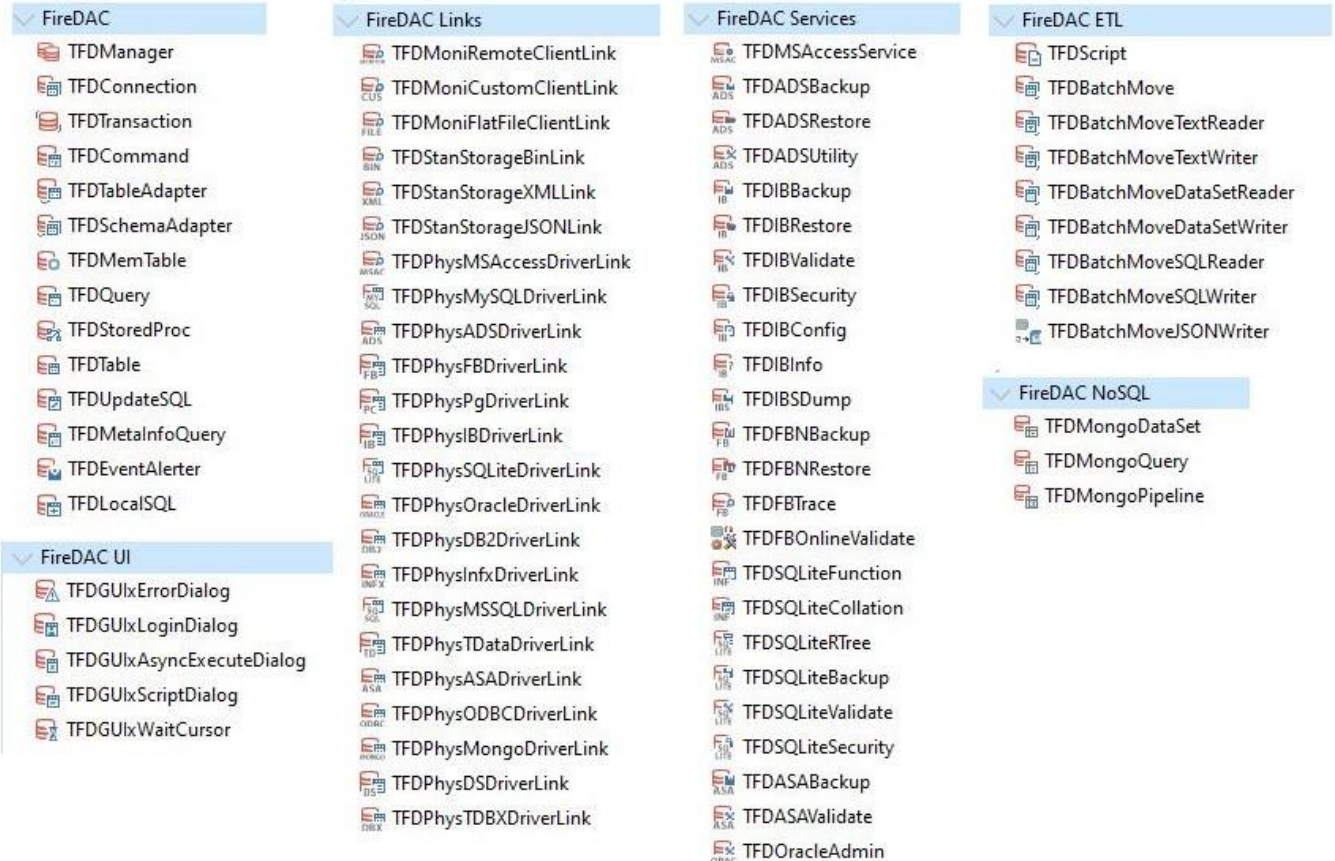


파이어닥이 지원하는 모든 데이터베이스 기본 기능과 각각의 고유 기능들을 (직접 연결 또는 브릿지 드라이버를 통한 방식으로) 활용할 때, 기존에 구현한 소스 코드와 컴포넌트는 계속 그대로 사용할 수 있다. 파이어닥에는 No-SQL 데이터베이스인 MongoDB를 다루는 컴포넌트도 포함되어 있다. 파이어닥의 디버깅과 성능 모니터링 인터페이스를 사용하면 연결된 데이터베이스와 애플리케이션 사이의 상호 작용을 모니터링하고 추적할 수 있다.

파이어닥(FireDAC)의 장점은 다음과 같다:

- 파이어닥은 데이터베이스 애플리케이션 실전 개발 경험 10년의 결실이다.
- 개발자는 프로그램 개발에 집중할 수 있다. 복잡한 데이터베이스 상호 작용은 파이어닥이 처리한다.
- 파이어닥은 주요 RDBMS 환경을 모두 커버한다. 또한 각 RDBMS별 고유 기능을 사용할 수 있도록 풍부한 최신 컴포넌트 세트를 제공한다.
- 파이어닥은 단일 컴포넌트 세트만으로 모든 지원 RDBMS를 다룰 수 있다.
- 파이어닥은 총 소유 비용(TCO)을 낮춘다.
 - 고가의 라이브러리를 별도로 구입하지 않기 때문에 초기 비용이 작다.
 - 독창적이고 강력한 라이브러리를 제공하면서도 개발자가 익숙하게 사용하기까지 걸리는 시간이 짧다.
 - 다른 데이터베이스로 대상을 변경해도, 애플리케이션 설계와 개발을 바로 진행할 수 있다.

FireDAC 컴포넌트들은 다음과 같다:



6.2 비주얼 라이브바인딩

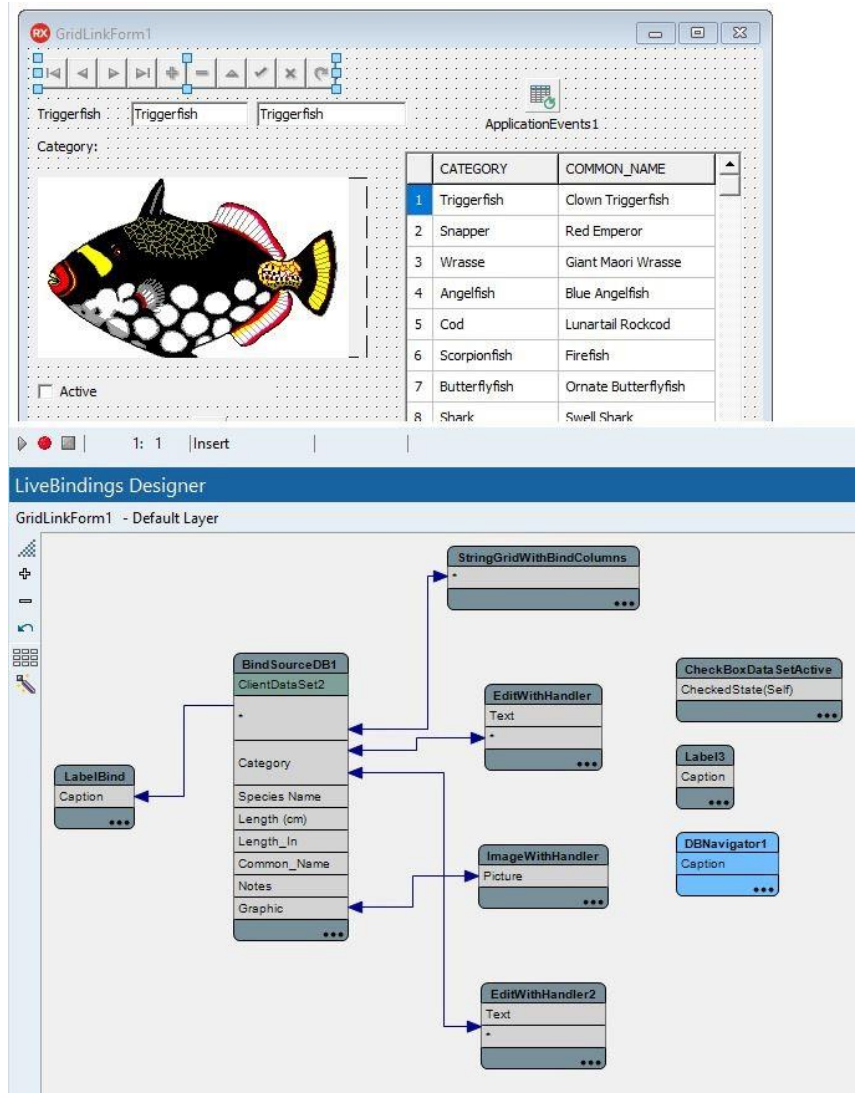
라이브바인딩은 표현식을 기반으로 하는 프레임워크이다. VCL과 파이어몽키 모두 데이터 바인딩 기능을 지원한다. 라이브바인딩은 바인딩 표현식이라고하는 관계식을 기반으로 단방향 또는 양방향으로 데이터를 바인딩한다. 바인딩 표현식을 사용하면 어떤 오브젝트든 다른 어떤 오브젝트에 바인딩할 수 있다. 서로 바인딩되는 오브젝트의 (하나 또는 복수의) 프로퍼티에 데이터가 어떻게 연결될 것인지를 바인딩 표현식으로 정의하면 된다.

예를 들어, 에디트 컨트롤(TEdit)을 레이블(TLabel)에 바인딩하면, 에디트 박스의 텍스트가 변경될 때 바인딩 표현식에 정의된 바에 따라, 레이블의 텍스트가 자동으로 변경된다.

다른 예를 들면, 트랙 바 컨트롤을 진행 바(progress bar)에 바인딩하면, 사용자가 트랙 바를 움직일 때마다 진행 바의 진척도가 증가하거나 감소하도록 만들 수 있다. 동일한 방식으로, 데이터 셋에 있는 특정 필드를 여러 오브젝트에 연결할 수 있고, 연결된 각 오브젝트에 있는 여러 프로퍼티를 변경하도록 만들 수도 있다. 라이브바인딩은 전파되는 특징이 있는데, 컨트롤 오브젝트에 바인딩된 오브젝트를 또다시 다른 오브젝트에 연결하는 식으로 이 특징을 이용하면, 원하는 오브젝트의 프로퍼티까지 연결하여 변경할 수 있다.

라이브바인딩 디자이너에서 컴포넌트 프로퍼티와 데이터를 눈으로 보면서 연결하면 코드를 작성할 필요가 없고, 개발 속도는 더 빨라진다. (개발 초기에는 DB팀에서 데이터베이스 설계나 데이터가 미처 제공하지 못하는 경우가 종종 있는데 이와 같이) 실제 데이터베이스가 없는 경우에는 라이브바인딩으로 프로토타입 데이터를 생성해 개발을 진행할 수 있다. 실제 데이터가 준비되면, 마우스 클릭 한 번으로 프로토타입 데이터 대신 실제 데이터를 연결할 수 있다.

라이브바인딩 디자이너 뷰가 복잡해지면 디자인 화면을 줌인 또는 줌아웃해서 확인할 수 있고, 화면의 현재 이미지를 파일로 저장할 수도 있다. 라이브바인딩 디자이너에서는 원하는 컴포넌트를 숨기거나 표시할 수 있어서 바인딩된 요소들만 보이도록 할 수 있다.



자세한 내용은 "VCL 응용 프로그램에서 라이브바인딩 사용" 튜토리얼에 설명되어 있다. (아래 링크 참조)
http://docwiki.embarcadero.com/RADStudio/en/Tutorial:_Using_LiveBinding_in_VCL_Applications

6.3 양방향 비주얼 도구

개발환경의 폼 디자이너에서 실시간 스냅-힌트와 레이아웃 안내선을 활용하면 네이티브 Windows 애플리케이션 화면을 눈으로 보면서 구성하는데 도움이 된다. 폼 디자이너의 시각적 양방향 기능과 컴포넌트, 프로퍼티, 이벤트를 사용하면 Windows 애플리케이션의 UI 작성 속도는 빨라지고 코드 작성량은 줄어든다.

애플리케이션 UI는 오브젝트 인스펙터에서 프로퍼티를 변경하여 바꿀 수 있다. 또한, 폼을 텍스트 모드로 열어서 UI가 정의된 텍스트를 직접 수정하는 방식으로도 UI를 바꿀 수 있다. (폼을 표현하는 텍스트에는 폼에 정의된 컴포넌트, 프로퍼티, 이벤트와 지정된 내용이 텍스트로 서술되어 있다). 소스 코드가 변경되는 경우에도 폼, 컴포넌트, 프로퍼티, 이벤트에 해당 변경 사항이 반영된다.

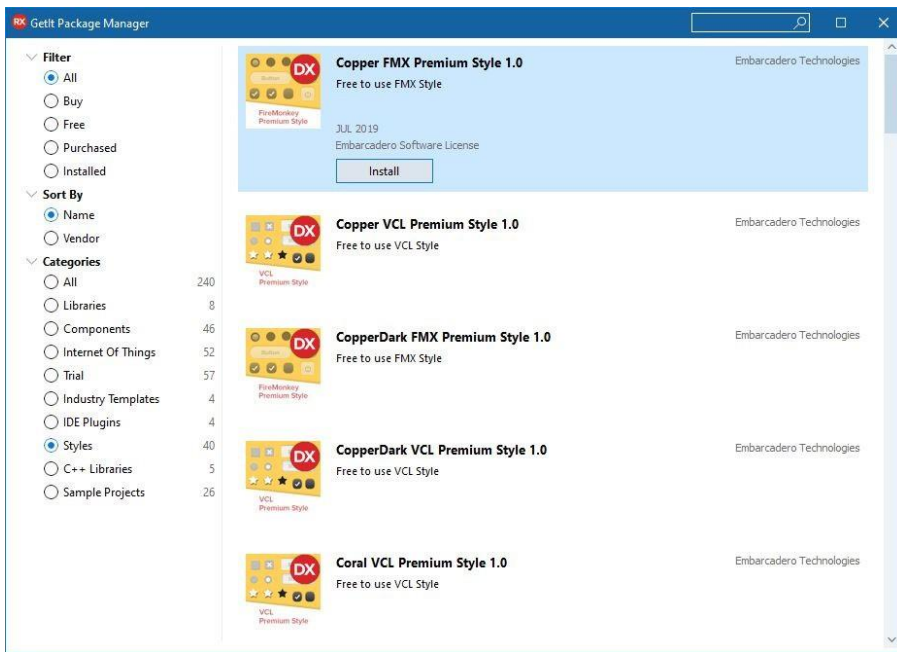
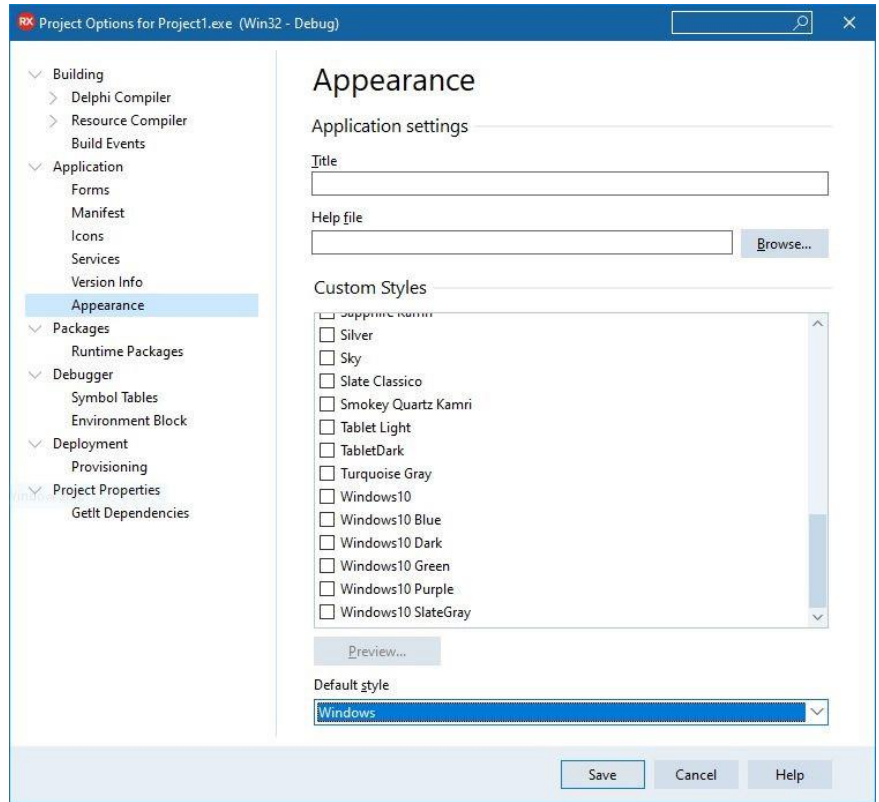
아래 그림은 고객별 판매 현황을 단 한 줄의 코드도 작성하지 않고 완성하는 개발 화면이다. 폼 디자이너에서 실제 데이터베이스에서 마스터/디테일 관계인 고객과 판매 데이터를 연결하고 라이브바인딩을 사용하여 구현한 것이다.

The screenshot shows the RAD Studio IDE with a live-binding application. The main window displays a 'Customer and Sales Master Detail Demo' with two tables. The top table lists customer information, and the bottom table lists sales orders. The LiveBindings Designer at the bottom shows the data flow between a DBNavigator, a BindSource, and a StringGrid.

CUST_NO	CUSTOMER	CONTACT_F	CONTACT_L	PHONE_NO	ADDRESS_L1	ADDRESS_L2	CITY	STATE_PRO	C
1001	Signature Dr	Dale J.	Little	(619) 530-2	15500 Pacifi		San Diego	CA	U
1002	Dallas Techn	Glen	Brown	(214) 960-2	P. O. Box 4		Dallas	TX	U
1003	Buttle, Griff	James	Buttle	(617) 488-1	2300 Newbu	Suite 101	Boston	MA	U

PO_NUMBER	CUST_NO	SALES_REP	ORDER_STA	ORDER_DAT1	SHIP_DATE	DATE_NEED	PAID	QTY_ORDER	TOTL
V9324200	1001	72	shipped	8/9/2012	8/9/2012	8/17/2012	y	1000	5600
V9324320	1001	127	shipped	8/16/2012	8/16/2012	9/1/2012	y	1	0.00
V9320630	1001	127	open	12/12/2012		12/15/2012	n	3	6000
V9420099	1001	127	open	1/17/2013		6/1/2013	n	100	3399
V9427029	1001	127	shipped	2/7/2013	2/10/2013	2/10/2013	n	17	4222

VCL의 경우, VCL 스타일 디자이너에서 사용자 지정 스타일을 만들 수 있다. 공통 대화 상자, TWebBrowser 컴포넌트, TGrid의 자손 등의 스타일도 이곳에서 지정할 수 있다. 스타일에 정의된 각 VCL 컨트롤의 요소와 상태 역시 개발자가 변경할 수 있다. VCL 컨트롤은 파트와 상태로 이루어져 있다. VCL 스타일은 이 파트와 상태 각각의 값들을 하나의 세트로 모아 놓은 것이다. 예를 들어 스크롤 바에 있는 각 파트는 프레임, 슬라이드, 각 방향으로 이동할 수 있는 두 개의 측면 버튼이다. 측면 버튼의 상태로는 눌림, 비활성화, 핫, 일반 상태가 있다.



파이어몽키의 경우, 컨트롤 클래스들은 각 지원 플랫폼별로 알맞은 스타일이 기본으로 지정되어 있다. 컨트롤 팝업 메뉴의 Edit Default Style을 이용하면 하드 코딩된 스타일의 사본이 생성된다. TStyleBook 컴포넌트에는 기본 스타일 정보뿐만 아니라 사용자 정의 스타일 정보도 들어있다. 파이어몽키 스타일 디자이너 창을 통해 기본 스타일을 편집하거나 새 스타일을 만들 수 있다. 이렇게 작업한 스타일 정보는 TStyleBook 컴포넌트에 저장된다.

특정 컨트롤 컴포넌트의 스타일을 변경하려면, 스타일 디자이너 창의 스트럭처 뷰 화면에서 원하는 컨트롤 컴포넌트를 선택하여 수정하면 된다. 필요한 수정 작업이 끝나면 저장하고 화면을 종료하면, StyleBook에는 모든 변경 사항이 적용된다. 편집 내용을 확인하고 싶다면 StyleContainer를 더블 클릭하면 된다.

6.5 Windows API 및 WinRT 유닛

델파이에는 델파이 소스 코드도 함께 제공된다. 델파이의 런타임 라이브러리(RTL) 소스 코드 역시 제공되는데, 그 중에는 Windows API와 WinRT API를 래핑해놓은 유닛들의 코드도 들어있다. Windows 애플리케이션을 개발하는 개발자 대부분은 프로퍼티, 메소드, 이벤트에 접근할 때 컴포넌트를 사용한다. 하지만 혹시라도 Windows API를 직접 호출하는 코드를 작성해야 하는 경우가 있다면, 이 유닛들의 소스 코드가 매우 유용할 것이다.

Windows API와 WinRT API 유닛의 소스 코드는 설치 디렉토리 안 아래 경로에 있다.

- C:\Program Files (x86)\Embarcadero\Studio\20.0\source\rtl\win
- C:\Program Files (x86)\Embarcadero\Studio\20.0\source\rtl\win\winrt

(참고: 위 경로 중 20.0은 10.3 Rio 버전의 지정 폴더이다. 이 숫자가 작을수록 더 오래전 버전이 설치된 것이다. 예를 들어 10.2 버전의 폴더 경로 숫자는 19.0 이다.)

6.6 병렬 프로그래밍 라이브러리 (PPL)

런타임 라이브러리(RTL)에는 병렬 프로그래밍 라이브러리(PPL)가 포함되어있어서, 개발자는 애플리케이션 안에서 여러 작업(Task)이 병렬로 실행되도록 구현할 수 있다. PPL에는 병렬 처리를 위한 여러 고급 기능이 들어 있다. 예를 들어, 원하는 처리가, 특정 작업(Task)에서 실행, 특정 작업에 참여, 또는 특정 작업 그룹에서 대기하도록 구현하는 코드를 매우 간결하게 작성할 수 있다. PPL에는(CPU 부하를 바탕으로) 자동으로 조정되는 스레드풀(Thread Pool)이 들어있어서, 스레드 생성과 관리를 따로 신경 쓰지 않아도 된다.

PPL을 사용하고 싶으면, 애플리케이션에 System.Threading 유닛을 적용하면 된다. 이 유닛에 있는 여러 기능은 새 프로젝트 뿐만 아니라 기존 프로젝트에서도 사용할 수 있다. 이 유닛에는 오버로드 되는 인수들이 많아서 델파이 뿐만 아니라 C++에서도 사용하기 좋다. PPL은 Windows(32비트, 64비트), 맥OS, 안드로이드, iOS 장치에서 작동한다.

PPL을 사용하면, 애플리케이션 코드에서 다음 작업을 쉽게 구현할 수 있다.

- Parallel.For를 사용해 더 루핑(반복문) 수행 속도를 향상
튜토리얼: 병렬 프로그래밍 라이브러리에서 For 루프 사용
- TTask와 ITask를 사용해 여러 작업을 병렬로 실행
튜토리얼: 병렬 프로그래밍 라이브러리에서 작업(Task) 사용
- 처리 중인 작업이 계속 진행되도록 내버려둔 채로, 원하는 다른 작업(Task)을 진행하고 필요한 시점이 되면 앞의 작업이 실행된 결과를 받아 오기. (IFuture는 애플리케이션이 실행 중인 코드 블록의 우선 순위를 정하고 우선 순위가 밀린 작업이라 할 지라도 계속 참고하고 있다가 해당 결과가 필요할 때 반환해준다)
튜토리얼: 병렬 프로그래밍 라이브러리 장점 사용

델파이에는 **콘웨이의 라이프 게임** (Conways Game of Life) 병렬 버전을 구현하는 VCL 및 FMX 샘플 애플리케이션이 포함되어 있다.

추가 정보 및 튜토리얼은 다음 온라인 엠바카데로 DocWiki에 있다 .

http://docwiki.embarcadero.com/RADStudio/en/Using_the_Parallel_Programming_Library

6.7 Windows 10 컴포넌트, 고해상도 DPI 및 다중 모니터 지원

델파이는 새로운 Windows 10 기능 지원 등 최신 Windows 지원을 위해 계속해서 발전하고 있다. 델파이가 제공하는 새로운 운영체제와 Windows 10에서 활용 가능한 주요 핵심 기능들을 정리해 놓았다.

6.7.1 Windows 10용 컴포넌트

컴포넌트 팔레트에는 다양한 Windows 10 컴포넌트가 있다. 이 델파이 Windows 10 컴포넌트들은 Windows 7 및 8.x에서도 실행될 수 있다.

TWindowsStore 컴포넌트는 개발한 Windows 애플리케이션을 Windows 스토어에 연동해서 사용자가 보유한 앱, 사용 가능한 추가 기능들, 구입한 추가 기능들을 확인할 수 있는 기능과 평가판 모드 처리 기능을 제공한다.

TWindowsStore 컴포넌트는 Windows 플랫폼만 지원하며, VCL용과 파이어몽키용이 따로 있다.

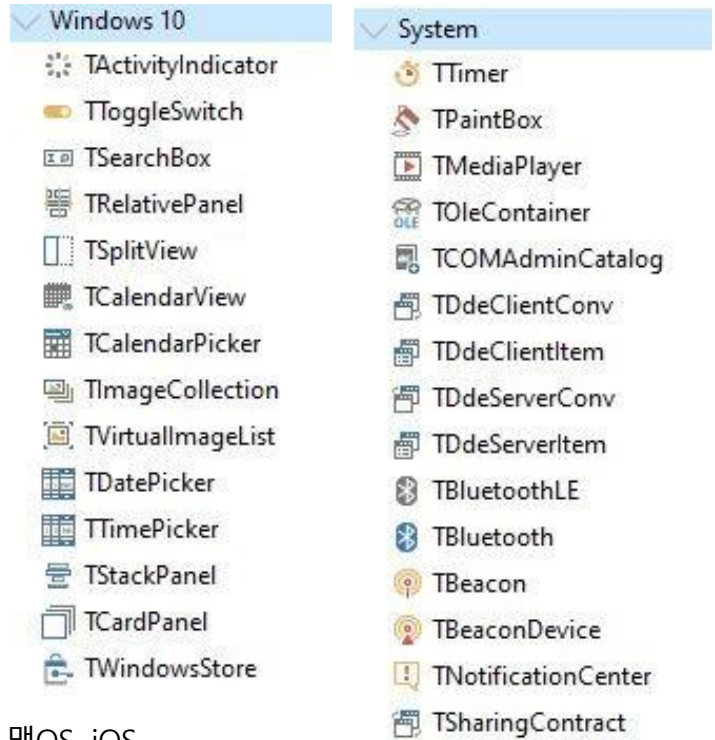
TNotificationCenter 컴포넌트로는 멀티-디바이스 알림을 관리할 수 있다. 애플리케이션은 알림 센터를 통해 사용자에게 알림 메시지를 보낼 수 있다. 알림 메시지는 각 지원 플랫폼마다 지정된 알림 영역에 표시된다. Windows에서 알림 기능은 VCL과 FMX 모두에서 사용할 수 있다.

TNotificationCenter 컴포넌트는 Windows 8.x, 10, 맥OS, iOS, 안드로이드에서 작동한다.

TSharingContract를 사용하면 Windows 공유 계약 기능을 통해 다른 Windows 애플리케이션과 콘텐츠를 공유할 수 있다. 델파이에 공유 계약 API가 구현되어 있으므로, 개발자는 애플리케이션을 공유 소스로 설정할 수 있다. 공유 대상은 네이티브 Windows 애플리케이션이어야 한다. 공유 계약 기능을 적용하려면 TSharingContract 컴포넌트를 그대로 사용해도 되고, TCustomSharingContract 컴포넌트를 구현해도 된다.

TSharingContract 컴포넌트의 프로퍼티는 다음과 같다.

- DataTitle과 Description: 사이드 바에 나타나는 타이틀과 타이틀에 대한 설명을 지정
- ImageFile: 공유할 이미지 지정 (개발자가 OnTransferImage 이벤트의 이벤트 핸들러를 직접 작성할 수도 있다. 이 방식은 개발자가 작성한 코드에 따라 애플리케이션에 있는 특정 스트림을 다른 애플리케이션으로 전달할 때 사용된다.)
- 대상 응용 프로그램과 훨씬 더 많은 정보를 공유할 수 있는 다양한 프로퍼티가 들어있다. 공유 계약에 대한 자세한 내용은 [Windows 개발자 센터 - 콘텐츠 공유 지침](#) 및 Windows 개발자 데이터패키지 클래스를 참고하면 된다.



6.7.2 Windows 10 고해상도 DPI 지원

엠바카데로 개발도구 담당자들 짐 맥키트, 마르코 칸투, 데이비드 밀링턴이 Windows 10의 VCL 앱 고해상도 지원에 대해 블로그에 작성했는데, 당시 짐 맥키트는 다음과 같이 요약했다. “고해상도 (96 DPI 보다 큰) 모니터를 사용할 경우, Windows는 배율을 조정해 글꼴과 이미지를 선명하게 표현합니다. 하지만 폰트 크기가 작아지기 때문에 읽기가 어려워집니다.



VCL은 크기 조정을 대부분 자동으로 처리합니다. 하지만, 개발자가 고해상도를 보다 세밀하게 다룰 수 있도록 하는

GetSystemMetricsForWindow 함수도 VCL.Classes 유닛에 새로 추가되었습니다. 이 함수는 Windows의 GetSystemMetricsForDPI(구.GetSystemMetrics를 대체) API를 활용하여 프로그램이 실행되는 모니터를 기준으로 프로그램의 각 창에 대한 해상도, DPI, 측정값을 알려줍니다.”

델파이 10.3 리오에서 제공하는 고해상도 지원을 위한 기능:

- **이미지 목록 및 이미지 컬렉션** 지원
- **모니터별 V2** 지원 (Project | Options | Application | Manifest 메뉴)
- 고해상도용 새로운 스타일 제공
- 고해상도용으로 업데이트 된 Windows API를 지원

더 많은 정보:

- <https://community.idera.com/developer-tools/b/blog/posts/high-dpi-on-windows-10>
- <https://community.idera.com/developer-tools/b/blog/posts/windows-10-and-modern-high-dpi-display-support-in-delphi-and-c-builder>

6.7.3 VCL 및 FMX에서 멀티 디스플레이 지원

VCL은 **여러 대의 모니터**를 다룰 수 있는 방법을 제공한다. 프로젝트를 시작할 때 TScreen 타입의 전역 변수 Screen이 생성된다. Screen은 프로그램이 실행중인 화면의 상태를 캡슐화한다. Screen이 하는 일반적인 작업에는 다음과 같은 것들이 있다:

- 응용 프로그램에서 액세스 할 수 있는 여러 모니터의 목록 및 매개 변수 처리
- 응용 프로그램이 실행되는 창의 크기
- 화면 장치에서 사용 가능한 글꼴 목록
- 커서 모양

VCL 애플리케이션을 여러 모니터에서 띄우면, VCL은 이 모니터들에 표시되는 애플리케이션 UI 레이아웃을 효과적으로 관리한다. 예를 들어 메뉴, 대화 상자, 기타 팝업 컨트롤은 부모 폼과 동일한 모니터에서 열린다. Screen 변수를 사용하면 모니터와 그 크기값까지 확인할 수 있다. 프로퍼티 중에서 Monitors를 사용하면 데스크톱을 구성하는 모든 모니터에 액세스 할 수 있고, MonitorCount로는 모니터 개수 정보를 확인할 수 있다. 모든 조정값은 메인 모니터의 값을 기준으로 설정된다.

FMX(파이어몽키)는 여러 모니터에서 애플리케이션이 실행되도록 자동 지원한다. 여러 화면에서 파이어몽키 애플리케이션을 실행하기 위해 별도의 프로그래밍을 구현할 필요가 없다. 파이어몽키 역시 VCL과 마찬가지로 여러 화면에서 실행되는 애플리케이션의 UI 레이아웃을 효과적으로 관리해준다. 메뉴, 대화 상자, 팝업 컨트롤들은 부모 폼과 동일한 화면에서 실행된다. 파이어몽키의 다중 디스플레이 지원은 VCL의 다중 모니터 지원과 유사하다. TScreen 클래스에는 여러 디스플레이의 매개 변수에 액세스 할 수 있는 메소드와 프로퍼티들이 있다. 애플리케이션은 자동으로 TScreen 타입의 Screen 전역 변수를 생성하게 되는데, 이 Screen 변수는 애플리케이션이 실행되는 화면의 상태를 캡슐화한 것이다. Screen을 통해 다음의 정보들을 확인할 수 있다:

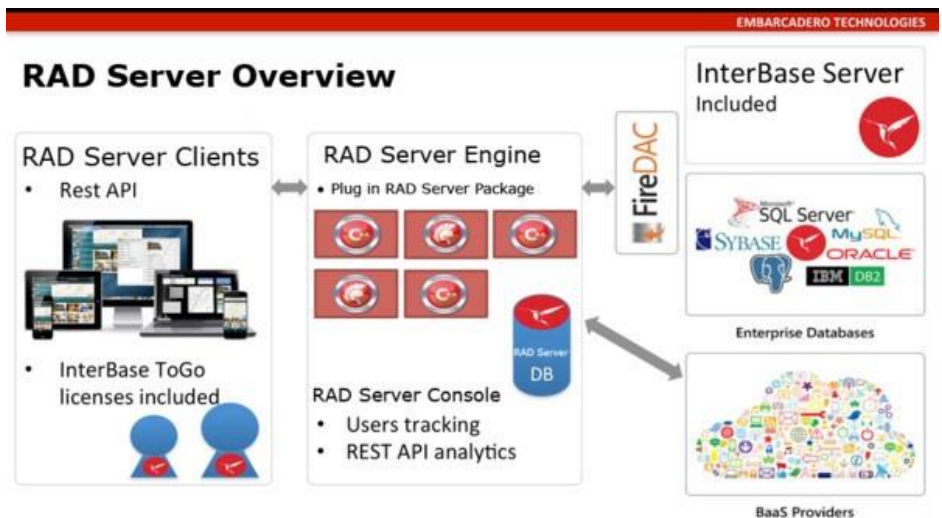
- 디스플레이 목록 (데스크톱 구성에 활용)
- 디스플레이 좌표, 크기, 기타 프로퍼티

관련 프로퍼티들 중 DisplayCount를 활용해 디스플레이 개수 정보 확인할 수 있고, Displays를 통해 디스플레이의 모든 속성을 액세스할 수 있다. 모든 좌표들은 가상 화면의 좌표값들로, 이 값들은 메인 디스플레이의 값을 기준으로 설정되며 픽셀 단위로 측정된다. 가상 화면은 데스크톱을 구성하는 모든 디스플레이 영역 전체를 의미한다.

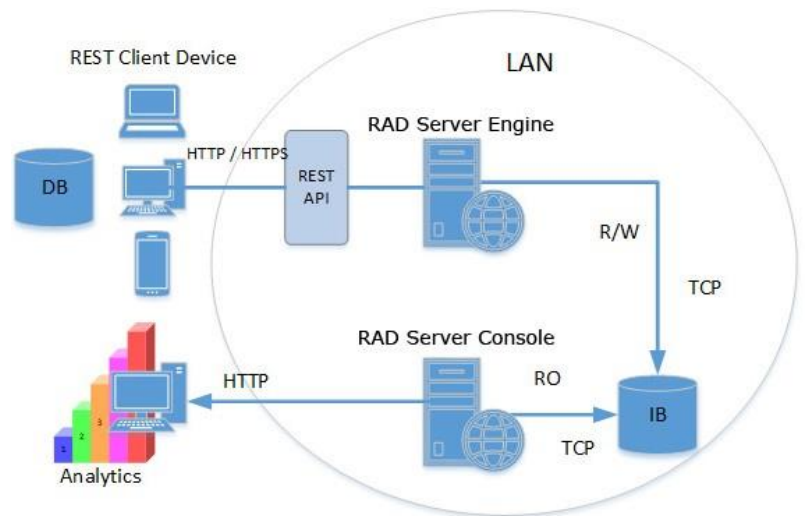
6.8 RAD서버

엠바카데로 RAD서버는 서비스 기반 애플리케이션을 빠르게 개발하고 배포할 수 있는 솔루션이다. 필요한 기능들이 이미 구축되어 있어 그대로 사용만 하면 되는 턴키 방식으로 RAD스튜디오, 델파이, C++ 빌더 개발자들에게 가장 적합한 솔루션이다. RAD서버는 REST

아키텍처 방식으로, JSON (또는 XML) 형식으로 매개 변수를 전달하고 결과값을 받는다. 눈에 띄는 기능으로는 API 게시, RAD서버에 연결될 기기와 사용자 관리, 애플리케이션 사용자와 사용 내용 분석 정보 제공, 파이어닥(FireDAC)을 활용한 로컬/엔터프라이즈 데이터베이스 연동, 사물인터넷(IoT) 기기 연동 등이 있다. 이 외에도 사용자 인증, 푸쉬 알림, 위치 정보 전달, 데이터 저장 등의 기능도 제공한다.



RAD서버 마법사, 컴포넌트, 도구들을 활용하면 미들웨어, 백엔드 애플리케이션 개발이 정말 빨라진다. 기존에 개발했던 델파이와 C++ 빌더 C/S 애플리케이션을 서버 또는 클라우드에서 실행할 수 있는 RAD서버 기반 애플리케이션으로 마이그레이션하는 작업도 많은 시간이 필요하지 않다. 데스크톱, 모바일, 콘솔, 웹 등 다양한 형태의 애플리케이션에서 REST 호출할 수 있도록 엔드 포인트 게시(publish)도 가능하다. RAD서버는 개발자가 서비스 응용 프로그램을 구축할 때 활용할 수 있는 도구, 컴포넌트, 데이터베이스 연결 및 인터페이스까지 풀 세트를 제공한다.



RAD서버 애플리케이션은 마이크로소프트 Windows IIS 및 아파치 웹 서버에 배포할 수 있다. 리눅스 인텔 64비트 서버도 델파이 기반 서비스를 배포할 수도 있다. RAD서버를 보다 쉽게 배포할 수 있도록 도커 (Docker) 컨테이너도 제공된다.

RAD서버 추가 정보:

- RAD서버 완벽 가이드(eBook) - <http://altd.embarcadero.com/releases/studio/general/The%20Complete%20Guide%20to%20RAD%20Server%20eBook.pdf>
- RAD서버 온라인 세미나 다시보기 - https://www.youtube.com/watch?v=eCyTiDq_jBY
- RAD서버 엠바카데로 아카데미 코스 (델파이용) - <https://www.embarcaderoacademy.com/courses/author/226326>
- RAD서버 도커 배포 - http://docwiki.embarcadero.com/RADStudio/Rio/en/RAD_Server_Docker_Deployment

6.9 애플테더링 (App Tethering)

RTL은 애플테더링 컴포넌트를 제공해 애플리케이션이 다른 애플리케이션과 상호 작용할 수 있도록 한다. 같은 장비에서 작동하는 애플리케이션들 간의 상호 작용뿐만 아니라, 원격 장비 간에도 가능하다. 애플테더링을 이용하면 다음의 기능들을 쉽게 구현할 수 있다.

- 애플테더링을 사용하는 다른 앱 탐지 - 동일한 디바이스에 있는 앱뿐만 아니라 연결가능한 디바이스에 있는 앱도 쉽게 탐지할 수 있다.
- 원격 조정 - 애플테더링을 통해 다른 애플리케이션에게 원격 작업을 전달하고 실행할 수 있다.
- 애플리케이션 간의 데이터 공유 - 표준 데이터 타입과 데이터 스트림을 공유할 수 있다.

앱테더링은 특정 전송 방식이나 프로토콜에 의존하지 않기 때문에, 앱테더링 API를 통해 새로운 전송 방식과 프로토콜을 반영할 수 있다. 앱테더링 RTL에는 IP 연결과 클래식 블루투스 연결이 이미 포함되어 있다. IP 연결 방식을 통해 같은 장비에서 실행되는 애플리케이션들을 서로 연결할 수 있다.

앱테더링용 컴포넌트는 다음 두 종류가 있다.

- TTetheringManager – 앱테더링을 사용중인 다른 애플리케이션을 찾을 때 사용한다 (동일한 디바이스 또는 다른 디바이스에서 실행중인 경우도 가능).
- TTetheringAppProfile – TTetheringManager를 통해 페어링된 적이 있는 애플리케이션들끼리 공유했던 행위와 데이터 정보가 필요할 때 사용한다.

C:\Users\Public\Documents\Embarcadero\Studio\20.0\Samples\Object Pascal\RTL\Tethering 폴더에서 델파이용 앱테더링 샘플들을 확인할 수 있다. 그 중 몇 가지를 소개한다.

- 재고 리스트 관리 샘플 (연결된 컴퓨터의 재고 정보를 앱의 재고 주문 리스트에 반영하기) – http://docwiki.embarcadero.com/CodeExamples/en/RTL.BDShoppingList_Sample
- 연결된 컴퓨터의 화면 스크린 샷 가져오기 샘플 – http://docwiki.embarcadero.com/CodeExamples/en/RTL.DesktopCast_Sample
- 앱테더링으로 연결된 원격 장비를 검색해 미디어 플레이어 원격 조정하기 샘플 – http://docwiki.embarcadero.com/CodeExamples/Rio/en/RTL.MediaPlayer_Sample
- 사진 촬영 후 연결된 컴퓨터 화면에 표시하기 샘플 – http://docwiki.embarcadero.com/CodeExamples/en/RTL.PhotoWall_Sample

6.10 블루투스, 비콘, 비콘펜싱 및 사물인터넷 (IoT)

델파이는 블루투스(클래식과 LE 모두) 구현을 위한 클래스와 컴포넌트를 제공한다. 블루투스 지원 디바이스에서 실행되는 애플리케이션이 블루투스를 통해 원격 디바이스와 통신하고 데이터를 교환할 수 있도록 지원한다.

클래식 블루투스 지원 구현하기:

- Bluetooth 컴포넌트를 추가한다.
- 원격 장치를 찾아서 페어링 한다.
- 페어링된 장치에 연결한다.
- 연결된 장치와 데이터를 교환한다.

블루투스LE로 표준 서비스 이용 구현하기

- TBluetoothLE 컴포넌트를 폼에 올려놓는다.
- 블루투스 LE 디바이스를 검색한다.
- 디바이스의 해당 서비스를 식별하고 반영한다.
- 해당 서비스의 특성을 식별하고 반영한다.
- 해당 특성을 읽거나 쓰거나 구독한다.
- 제공되는 원시 데이터를 통해 필요한 정보를 추출한다.

Platform	Classic Bluetooth	Bluetooth Low Energy	
		Client	Server
Windows*	✓	8+	
OS X	✓	10.7+	10.9+
iOS		5+	6+
Android	✓	4.3+	5+

비콘은 저전력(LE) 블루투스 장치로, 내장되어 있는 특정 정보를 발신한다. 특히 제조자가 지정한 특정 정보를 발신한다. 비콘 주변에 있는 블루투스LE 지원 장비는 해당 비콘을 식별하고 거리를 계산할 수 있다. 이를 위해 별도의 페어링이나 연결을 맺을 필요가 없다.

Platform	Supported
Windows(*)	10+
OS X	10.7+
iOS	5+
Android	4.3+

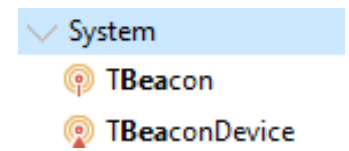
* 비콘용 RTL API는 Windows 10 이상 플랫폼에서만 사용할 수 있다. 사용할 수 있는 형식은 iBeacon과 AltBeacon 두가지이다. Windows 10 이전 버전은 BLE 알림을 지원하지 않는다.

Windows 외 다른 플랫폼에서는 3가지 비콘 형식(iBeacon, AltBeacon, Eddystone)을 사용할 수 있다. 비콘용 RTL API에는 TBeaconScanMode와 TKindofBeacon라는 특별한 두가지 타입이 있다. 개발자는 모니터링하려는 비콘의 형식을 알아야한다.

- iBeacon – 애플에서 정의한 형식 (전체 사양은 [여기](#)에서 확인)
- AltBeacon – 개방 형식 (전체 사양은 [여기](#)에서 확인)
- Eddystone – 구글에서 정의한 개방 형식 (전체 사양은 [여기](#)에서 확인)

비콘 컴포넌트와 클래스는 다음과 같다.

- TBeacon –비콘 컴포넌트용 클래스
- TBeaconDevice –비콘 장치 컴포넌트용 클래스
- TBeaconRegionCollection – TBeaconRegionItem 리스트 정보가 저장되어 있는 컬렉션
- TBeaconRegionItem – 등록된 지역 정보가 저장되어 있는 클래스
- TCustomBeacon – TBeacon 컴포넌트의 기반 클래스
- TCustomBeaconDevice – TBeaconDevice 컴포넌트의 기반 클래스



비콘에 대한 추가적인 정보:

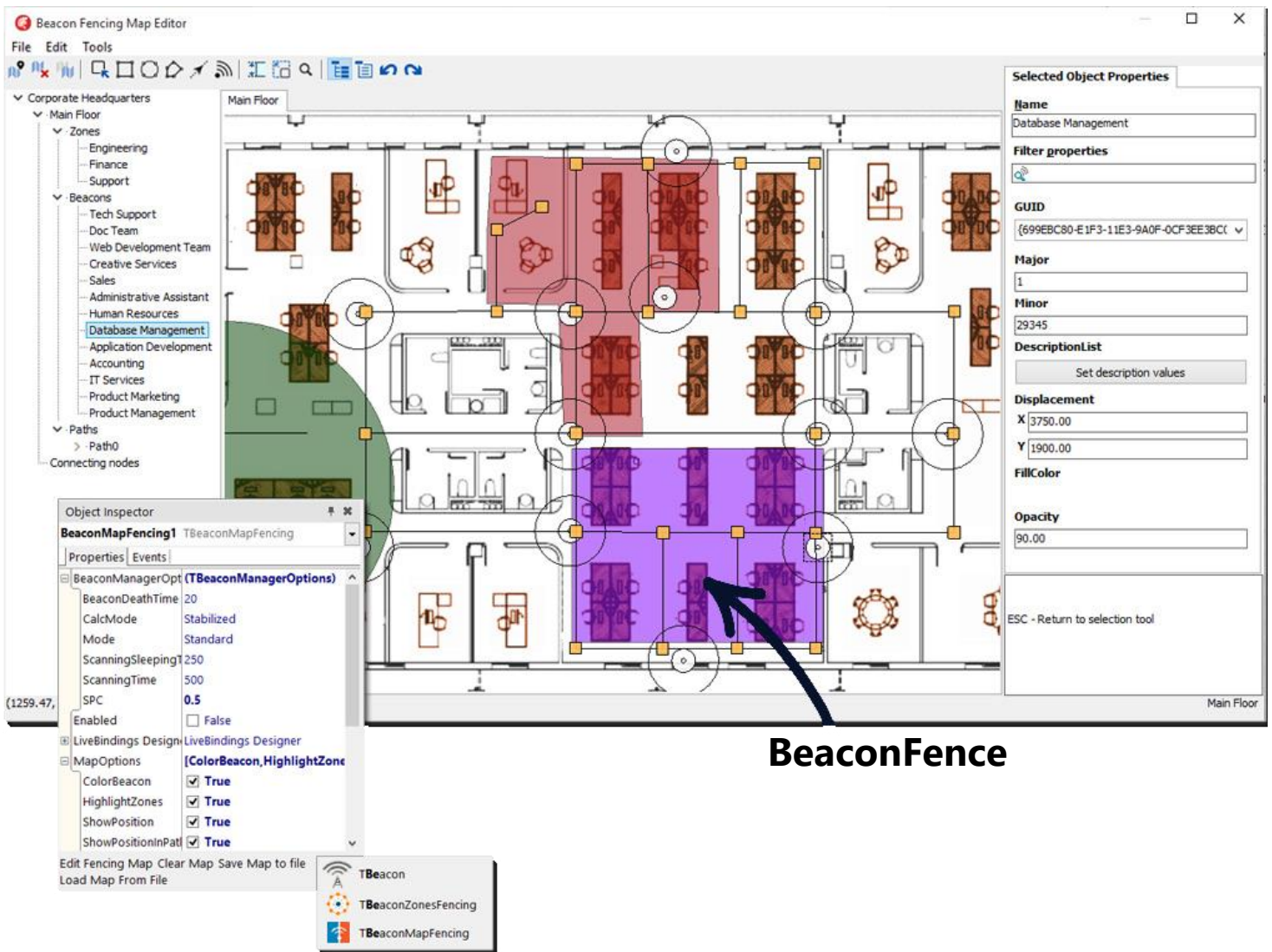
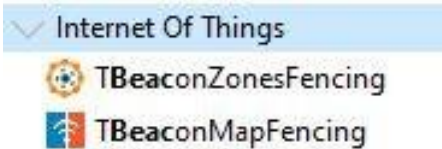
http://docwiki.embarcadero.com/RADStudio/en/Using_Beacons

비콘펜스(BeaconFence)를 활용하면 위치 기반 서비스를 손쉽게 구현할 수 있다. 비콘펜스 맵 편집기가 System.Beacon.TBeaconManager에 추가되어 있어 원하는 공간을 지도로 표현하고 필요한 설계를 반영할 수 있다. 특정 구역, 건물 등 실제 공간뿐만 아니라 가상 공간에도 적용할 수 있다. 비콘펜스는 겯잇 패키지 매니저에서 다운로드 할 수 있으며, 배포 횟수는 라이선스에 따라 제한이 있다.



비콘펜스 배포 라이선스는 RAD서버 엔터프라이즈 및 아키텍트 라이선스에 포함되어 있다. 비콘펜스에는 유용한 프로퍼티와 이벤트가 들어 있다. 예를 들어 OnZoneEnter는 사용자가 애플리케이션이 작동하는 장비를 가지고 특정 지역에 진입하면 작동하도록 되어 있는 이벤트이다.

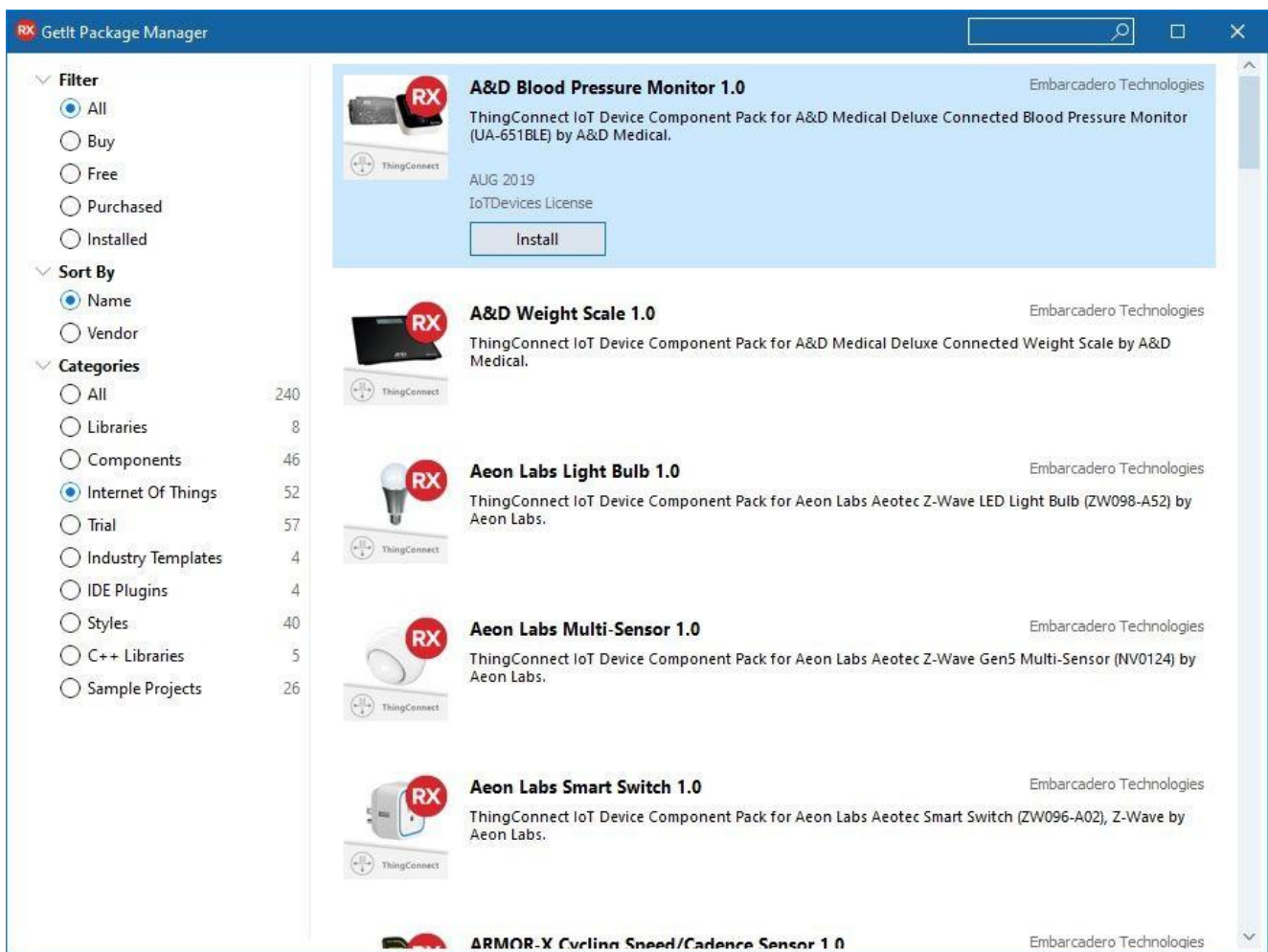
비콘펜스에 있는 비콘펜싱 맵 에디터는 비트맵을 만들 수 있는 GUI 에디터로, 평면도 등 기존 비트맵을 사용할 수 있도록 가져오는 기능도 제공한다.



비콘펜스로 구현 가능한 기능들:

- 비콘 모니터링과 근접 거리에 따라 원하는 이벤트를 발생시킨다.
- 구역을 설정하고 근접 거리에 따라 원하는 이벤트를 발생시킨다: 구역 진입, 구역 이탈, 구역 접근중 등
- 모바일 장치의 지도상 위치를 계산한다.
- 관심 지점 간의 지도상 경로를 계산한다.

델파이에는 블루투스를 지원하는 **사물인터넷(IoT) 장비**를 광범위하게 지원한다. 의료, 피트니스, 홈 오토메이션 등 다양한 장비용 사물인터넷 컴포넌트가 제공되고 있다. 사물인터넷 컴포넌트에는 블루투스LE (BLE)와 Z-Wave 장비 모두 해당된다. 컴포넌트가 지원되는 사물인터넷 장비로는 혈압 모니터, 심박수 모니터, 체중계, 전구, 스마트 스위치, 온도/습도 센서, 누수 점검 센서, 연기 및 일산화탄소 센서, 프로그래밍 가능한 온도 조절 장치, 도어 잠금 장치, 온도계 등 다양하다. 50가지 이상의 사물인터넷 컴포넌트와 샘플 프로그램들은 깃잇 패키지 매니저에서 다운로드할 수 있다.



6.11 인터베이스 데이터베이스

인터베이스는 SQL 표준 호환 데이터베이스로, SQL 산업 표준을 엄격하게 준수한다. 인터베이스는 유니 코드를 지원하며 전 세계 모든 어떤 언어 환경에도 잘 맞는다. 그리고 다중 조인 유니 코드, 실시간 이벤트 경고, 암호화 그리고 특허 받은 기능인 체인지뷰(Change Views)를 통한 SQL 기반 변경 추적을 제공한다.

인터베이스의 주요 기능은 다음과 같다:

체인지뷰(Change Views)는 데이터베이스 변경 사항만 식별하여 따로 가져오기 때문에 애플리케이션 실행 속도는 빨라지고 네트워크 트래픽과 비용은 절감된다. 체인지뷰 기술백서를 읽어볼 것을 권한다.

- 다세대 아키텍처(Multi-Generational Architecture) 기반으로 데이터베이스 스냅샷을 일관성있게 유지한다. 읽기 작업이 쓰기 작업을 통한 데이터 변경을 방해하지 않으며, 유연하게 확장할 수 있다.
- 이벤트 알림은 변경 내용이 커밋되면 연결된 클라이언트에게 실시간 경고를 전달한다. CPU에 의존하는 데이터베이스의 풀링은 줄어들고, 확장성은 높아진다.
- 인터베이스를 손쉽게 커스터마이징 할 수 있다. 커스터마이징 가능한 필드 타입, 저장 프로시저, 예외처리, 트리거를 사용하면 데이터베이스 로직과 데이터 변환을 자동화할 수있으며, 손쉽게 관리하는데 도움이 된다.
- 제너레이터는 프라이머리 키에 많이 사용되는 순차적 번호를 제공하며, 애플리케이션은 생성된 값을 받아들일 수 있다. 인터베이스는 중복 키를 할당하지 않으며, 제너레이터를 통해 데이터베이스 중앙에서 키 값을 생성한다.
- 인터베이스에는 통신 암호화, 저장 암호화, 분리 보안 로그인, 역할 기반 사용자 보안이 제공된다. 디스크 암호화 방식으로 전체를 암호화하는 경우에도 데이터베이스 속도 저하와 성능 부담은 거의 없다.

인터베이스는 **개발자 에디션과 배포용 에디션 4가지**가 있다.

- 개발자 에디션
- 임베디드 IBLite / 임베디드 IBToGo / 데스크톱 에디션 / 서버 에디션

6.12 엔터프라이즈 커넥터

엠바카데로는 2019년 11월부터 엔터프라이즈 커넥터 (CData FireDAC Professional) 1년 구독 라이선스 (999 달러 상당)를 10.3 델파이, C++ 빌더, RAD스튜디오의 엔터프라이즈와 아키텍트 에디션에 포함한다고 발표했다. 엔터프라이즈 커넥터의 주요 이점은 다음과 같다.

- 파이어닥(FireDAC) 엔터프라이즈 커넥터로 손쉬운 데이터 이동, 통합, 분석이 가능하다. 70개 이상의 엔터프라이즈 애플리케이션이 통합 제공된다. SQL을 사용하는 표준 모델 방식으로 연결된 데이터를 일관성 있게 다룰 수 있다.
- 퀵북스(QuickBooks) Desktop, 메일침프(MailChimp), 세일즈포스, 유튜브, SugarCRM, 지라(Jira), 서버이몽키, 아마존 DynamoDB, Couchbase, 페이팔(PayPal), 이베이, 구글시트, 페이스북, 트위터, Slack, 드랍박스 등을 연결할 수 있는 컴포넌트가 포함되어 있다.

엔터프라이즈 커넥터에 대한 추가 정보는 **CData 엔터프라이즈 커넥터 제품 페이지**를 참고하면 된다.



델파이 코드 생산성 및 성능

7. 델파이 코드 생산성 및 성능

2018년 전세계 엠바카데로 RAD스튜디오 고객들을 대상으로 진행한 설문조사에서 델파이와 C++빌더에 대한 평가는 매우 긍정적이었다. 1,200 곳이 넘는 고객사 중 89%가 RAD스튜디오 개발환경을 사용하여 개발 주기를 단축하는 성과를 이루었다고 답변했다.

독립 조사 기관인 **TechValidate** 는 제조, 컴퓨터 소프트웨어/서비스, 정부 기관, 전자, 보건/의료, 교육, 금융 서비스, 항공, 국방 등 다양한 분야에서 중소기업, 대기업, 기관에 이르기까지 광범위한 조사를 진행했다.

TechValidate에 따르면, 고객사들은 애플리케이션 개발 과정에서 RAD스튜디오를 통해 해결한 주요 과제를 "실행 속도가 빠른 성능 좋은 애플리케이션 개발 (50%)", "최신 플랫폼과 장비를 지원하는 능력 유지 (50%)", "프로젝트 마감일 준수 (37%)", "여러 플랫폼으로 배포 (36%)", "여러 디바이스 유형별 반응형 디자인 (27%)" 순으로 꼽았다.

RAD스튜디오를 사용해 애플리케이션을 개발하는 고객이 선호하는 기능은 "비주얼한 디자인 (82%)", 광범위한 컴포넌트 라이브러리 (69%), 네이티브 OS 컴파일러 (58%), 멀티 플랫폼 지원 (55%) 이었다.

델파이 또는 C++빌더로 애플리케이션을 20개 이상 개발/배포하고 있다고 답변한 RAD스튜디오 고객은 44%에 달했다. 이는 개발환경의 사용성, 확장성, 민첩성 측면의 우수성을 입증하는 것이다. 개발한 애플리케이션들은 "기업 (60%)", "공업 (36%)", "엔지니어링 (32%)", "생산 (29%)", "과학 (18%)" 분야에서 많이 사용되었다.

사용자들은 개발 비용 절감, 시장 출시 시기 단축, 미리 설정되고 테스트된 컴포넌트 활용, 현대식 Windows 10 UI 구현, 추가 플랫폼 지원을 통한 고객층 확장 능력 등을 장점으로 꼽았다.

7.1 생산성

관리자, 개발자가 개발 능력과 팀 생산성을 위해 가장 많이 고려하는 기능과 장점:

- 프로젝트와 소스 코드의 호환성 - 32/64비트 Windows 버전(7, 8.x, 10), 다양한 플랫폼(Windows, 맥OS, iOS, 안드로이드, 리눅스) 간의 호환성이 보장된다면 현재 사용자들과 미래 고객들이 원하는 플랫폼 앱을 언제든 개발할 수 있다.
- **컴포넌트** - 많은 기능들이 사전 구축되어 있고 확장이 쉬워 코딩 작업 많이 하지 않고도 UI와 비즈니스 로직을 구현할 수 있으므로 개발 속도가 빨라진다. 컴포넌트와 플랫폼 내부에도 필요한 경우 직접 코드를 작성할 수 있도록 하는 인터페이스 역시 제공된다.
- **라이브바인딩** - 개발 시점에 연결할 컴포넌트들을 직접 보면서 연결할 수 있다. 코드를 작성할 필요가 없다.
- **실제 데이터 확인** - 개발 시점에 (데이터베이스와 REST 연결을 통해) 실제 데이터를 확인할 수 있어서, UI를 신속하게 개발할 수 있고 SQL 문과 연결이 올바른지 확인할 수 있다.

- 파이어닥(FireDAC) 컴포넌트와 구현 - 동일한 코드를 여러 데이터베이스에서 사용할 수 있고, 특정 데이터베이스의 고유 기능용으로 따로 제공되는 컴포넌트도 활용할 수 있다.
- **코드 인사이트(Code Insight)** - 개발을 더 빠르게 할 수 있도록 코드 자동 완성, 파라미터 자동 완성, 블록 자동 완성, 에러 통찰력, 코드 템플릿, 표현식 검증 툴팁이 제공된다. (온라인 도움말에서 코드 인사이트를 참고)
- **FMX** - 단일 소스로 멀티 플랫폼과 멀티 디바이스를 커버한다. 장비 화면 미리보기를 사용하면 개발속도는 더 빨라진다. 장비의 실제 화면 크기와 해상도를 미리 보면서 개발하기 때문이다.
- 산업별 템플릿 애플리케이션 - 해당 산업의 복잡한 애플리케이션을 개발하고자 할 때, 어떻게 설계, 개발, 배포하면 좋은 지를 알 수 있다.

C:\Users\Public\Documents\Embarcadero\Studio\20.0\Samples\Industry Templates\Object Pascal 폴더에 들어있다 (참고 : 위 경로 중 20.0은 10.3 Rio 버전의 지정 폴더이다. 이 숫자가 작을수록 더 오래전 버전이 설치된 것이다. 예를 들어 10.2 버전의 폴더 경로 숫자는 19.0 이다.).

- 다양한 소스 코드와 예제 - 다양한 예제는 델파이 설치 중에 함께 설치할 수도 있고 온라인에서 확인할 수도 있다. VCL, FMX, RTL의 소스 코드가 들어있으므로 델파이 라이브러리와 컴포넌트가 어떻게 구현되어 있는지를 보고 배울 수 있다. 델파이를 만든 델파이 코드를 직접 보고, 배울 수 있으므로 가장 근본적인 도움말이 되어준다.
- **감사(Audits)와 측정(Metrics)** - 개발환경에 통합되어 제공된다. 또한 커맨드 라인 도구로도 따로 제공된다. 회사 표준과 규약 준수 강제, 실시간 코드 측정, 발생 가능한 문제 파악, 코드 품질 향상에 활용된다.
- 오픈 소스 **유닛 테스트** 프레임워크 - RAD스튜디오에 통합되어 단위별 테스트 케이스 자동화를 구현하고 실행할 수 있다. 델파이와 C++ 빌더에서는 DUnit과 DUnitX 프레임워크가 활용된다. 개발된 애플리케이션의 기능을 담당하는 클래스와 메소드에 대한 테스트를 자동화하는 과정을 간편하게 하는 프레임워크이다. 유닛 테스트를 리팩토링(Refactoring)과 함께 사용하면 더욱 안정적인 애플리케이션을 만들 수 있다. 코드가 조금만 수정되어도 항상 표준 테스트 세트를 사용하여 검증하기로 하면, 발생 가능한 문제를 조기에 잡아낼 수 있다.
- 테스트 자동화 솔루션 - 써드 파티 기술 파트너들이 델파이용으로 제공한다. **테스트레일(TestRail)**, 라노렉스(Ranorex), **테스트컴플리트(TestComplete)** 등이 있다.
- 지속적 통합 (Continuous Integration, CI) 시스템 - 손쉽게 델파이에 적용할 수 있다. 델파이 프로젝트 파일이 MS Build 표준 형식을 따르고 있기 때문이다. 다음은 관련된 지속적 통합 솔루션 목록이다.
 - Travis CI - <https://travis-ci.com/>
 - FinalBuilder - <https://www.finalbuilder.com/finalbuilder>
 - Continua CI - <https://www.finalbuilder.com/continua-ci> Continua와 델파이를 함께 쓰는 방법 안내.
 - JenkinsCI- <http://thundaxsoftware.blogspot.com/2011/07/continuous-integration-for-your-delphi.html>
 - MSBuild - <https://docs.microsoft.com/en-us/visualstudio/msbuild/msbuild?view=vs-2019>
 - Amazon Code Pipeline - <https://aws.amazon.com/codepipeline/>
 - Cruise Control - <http://cruisecontrol.sourceforge.net/>

7.2 성능

네이티브 코드는 바이트 코드나 인터프리터를 이용하는 코드 보다 빠르기 마련이다.

자바스크립트, 파이썬, 자바 등 기타 컴파일 되지 않는 언어를 사용하는 경우 종종 구동 시간이 느려지고, 여러 프레임워크에 종속되는 문제에 시달릴 수 있다. 네이티브 코드 최적화 컴파일러, 미리 컴파일된 컴포넌트, 런타임 라이브러리를 활용해 애플리케이션 용량은 적으면서도 빠르게 완성할 수 있는 델파이만의 기능과 장점을 정리하였다. 애플리케이션 성능 향상에 도움이 되는 추가 도구 목록도 정리되어 있으니 참고하면 도움이 될 것이다.

- 델파이의 모든 최적화 컴파일러는 인텔 프로세서와 ARM 프로세서(타겟 플랫폼 지정에 따라 달라진다)에 맞게 네이티브 기계어로 컴파일하여 애플리케이션을 생성한다. 관련 컴포넌트와 런타임 라이브러리 역시 모두 기계어로 컴파일되므로 애플리케이션이 빠르게 실행된다.
- 델파이 컴파일러는 함수(Functions)와 프로시저(Procedures)에 **인라인 지시자 (inline directive)** 태그를 지정하여 성능을 향상시킬 수 있다. 특정 조건에 부합하면 컴파일러가 코드를 호출하는 대신, 해당 코드 자체를 직접 넣는다. 이 방식은 코드를 더 빠르게 작동하게 만드는 성능 최적화 방법이다.
- 병렬 프로그래밍 라이브러리 (PPL) 사용을 고려해보는 것도 좋다. 애플리케이션 로직 일부에서 프로세서 코어를 더 많이 활용하도록 하면 더 빠르게 반응하는 애플리케이션을 만들 수 있다.
- VCL 은 Windows API 와 Windows 기능을 완벽하게 통합해 놓았다.
- 델파이는 멀티-티어와 클라우드 인프라를 활용하기 위한 수고를 덜어내는 용도로도 사용된다.
- 파이어닥(FireDAC) 데이터베이스 액세스는 각 데이터베이스별로 최적화되어 있고 각 데이터베이스가 가진 고유한 장점을 활용할 수 있도록 추가 기능을 제공한다. 파이어닥의 캐시와 모니터링 기능은 각 데이터베이스에서 최상의 성능을 보장한다.
- 코드 프로파일링, 병목 찾기, 코드 커버리지 식별 등 애플리케이션을 보다 최적화할 수 있는 도구를 활용할 수 있다. 프로파일러를 미리 그리고 자주 사용하면 데이터 구조, 알고리즘, 실행 등의 문제를 늦기 전에 해소할 수 있다. 대표적인 델파이용 프로파일링 도구:
 - **Smarbear AQTime** (RAD스튜디오 개발환경 확장 팩에 들어있어서 함께 사용하기 좋다).
 - **Raize Software CodeSite** - 애플리케이션 실행 내부를 더 깊이 통찰할 수 있도록 로그 생성
 - **EurekaLog** - 예외, 메모리 누수 등을 추적.
 - **ProDelphi32와 ProDelphi64** - 32비트와 64비트 델파이 애플리케이션 프로파일러.
유튜브 데모 링크: https://www.youtube.com/watch?v=cJ_G8XUWSOI
 - **DeLeaker** - 델파이 애플리케이션 프로파일링. 유튜브 데모 링크:
<https://www.youtube.com/watch?v=rnq44Td9opQ>
- 데이터 사이언트 애플리케이션이나 다른 요구 사항 때문에 파이썬을 사용할 필요가 있는데, 네이티브 애플리케이션의 뛰어난 성능도 원한다면, 무료 컴포넌트인 **Python4Delphi** 사용을 고려하기 바란다. 델파이 애플리케이션에 파이썬을 넣을 수 있다.



기타 기능, 장점 및 관련 리소스

8. 기타 기능, 장점 및 유용한 자료

이 장에서는 개발자가 Windows 애플리케이션이나 크로스 플랫폼 애플리케이션을 만들 때 도움이 되는 델파이의 추가적인 기능과 장점을 정리한다. 여기 있는 리소스 링크들은 개발 생산성을 강화하는데 도움이 될 것이다.

8.1 크로스 플랫폼

델파이에는 최적화된 네이티브 코드 컴파일러와 네이티브 API/장비 지원 기능이 있어 (Win32, Win64, iOS, 맥OS, 안드로이드, 리눅스 등) 지원되는 모든 플랫폼에서 동일한 오브젝트 파스칼 언어 기능을 사용할 수 있다.

Windows7, Windows 8.x, Windows 10 운영체제에서 검증된 바와 같이 VCL 애플리케이션은 Win32와 Win64 버전 모두에서 동일한 코드, 컴포넌트, 라이브러리를 사용할 수 있다.

FMX 애플리케이션은 Win32, Win64, iOS, 맥OS, 안드로이드에서 동일한 코드, 컴포넌트, 라이브러리를 사용할 수 있다. 리눅스용 콘솔, 서버, GUI 애플리케이션도 델파이로 만들 수 있다.

언어, 컴포넌트, 라이브러리뿐만 아니라 기반 API와 플랫폼 인터페이스 또한 함께 제공하기 때문에 각 네이티브 플랫폼의 UI 호환성과 배포 지원 기능을 유지하면서도 생산성은 향상된다.

8.2 델파이 애플리케이션 가상화 및 배포 방법

일부 델파이 개발자는 델파이 애플리케이션을 사용자 컴퓨터 여러 곳에 설치하는 대신, Windows 서버나 클라우드 플랫폼에 배포하는 방식을 선택하고 있다. 개발자들이 고려할만한 몇 가지 선택을 살펴보자.

8.2.1 Cybele Software Thinfinity® VirtualUI

Cybele Software는 엠바카데로 기술 파트너이다. 델파이 애플리케이션을 가상화한다. 코드 한줄과 런타임 라이브러리 하나만으로 손쉽게 Windows 서버로 애플리케이션을 생성/배포할 수 있고 사용자들은 웹 브라우저에서 바로 사용할 수 있다.

Cybele Software의 Thinfinity® VirtualUI는 Windows 기반인 데스크톱 애플리케이션을 웹 브라우저와 장비를 넘나드는 웹 애플리케이션처럼 작동시켜준다. Thinfinity VirtualUI 기반 애플리케이션은 Windows 환경뿐만 아니라 HTML5를 지원하는 웹 브라우저를 통해 원격에서 사용할 수 있다. 2015년 VirtualUI가 출시된 후, 수 천개 회사에서 비즈니스 크리티컬한 애플리케이션을 안전하고 비용 효과적으로 확장함으로써 디지털 전환 수준을 높이고 있다.

델파이 애플리케이션에 코드 단 한 줄 추가하는 것 만으로도 듀얼 플랫폼 기능이 활성화되어 단 몇 분만에 웹 브라우저에서 실행되도록 변환할 수 있다. 인터넷 익스플로러, 엣지, 파이어폭스, 구글 크롬, 사파리 등 모든 HTML 웹 브라우저를 지원한다.

더 자세한 내용은 Cybele Software ThinFinity 제품 페이지에서 확인할 수 있다:

- 제품 페이지
<https://www.cybelesoft.com/thinfinity/virtualui/>
- 플랫폼 요구 사항
<https://www.cybelesoft.com/thinfinity/virtualui/requirements/>
- 델파이 안내 페이지
<https://www.cybelesoft.com/delphi-web/>
- 델파이 5분 튜토리얼 유튜브 비디오
<https://youtu.be/13ZANj8sUJE>

8.2.2 아마존 AppStream 2.0

아마존 AppStream 은 중앙 관리형 애플리케이션 스트리밍 서비스이다. 앱 스트림 2.0에서 데스크톱 애플리케이션을 중앙 관리하면 사용자가 몇 명이든, 전세계 어디에 있든 상관없이 쉽고 안전하게 해당 애플리케이션을 사용하도록 손쉽게 확장할 수 있다. 별도로 하드웨어나 인프라를 구입, 제공, 관리할 필요가 없다.

앱 스트림 2.0은 AWS에 있기 때문에 보안에 가장 민감한 조직을 대상으로 고안된 데이터 센터와 네트워크 아키텍처 역시 매우 훌륭하다는 장점도 누릴 수 있다. GPU사용이 많은 3D 디자인이나 엔지니어링용 애플리케이션을 사용할 때에도 사용자 각자는 유연하고 반응성이 좋은 사용자 경험을 한다. 왜냐하면 특정 사용 상황에 맞게 최적화된 가상 머신 (VM)에서 애플리케이션이 작동되고, 스트리밍 세션은 상황에 맞게 자동으로 네트워크 조건을 조정하기 때문이다.

기업에서는 앱 스트림 2.0을 사용하여 애플리케이션 공급을 단순화하고 클라우드로 전환을 완수할 수 있다. 교육 기관에서는 각 학생별로 수업에 필요한 애플리케이션을 어느 컴퓨터에서도 사용할 수 있다. 소프트웨어 개발사는 앱 스트림 2.0을 사용하여 평가판 데모 버전을 제공하거나 애플리케이션 교육을 제공할 때 다운로드나 설치를 따로하지 않아도 된다. 또한 애플리케이션을 다시 작성하지 않고도 완전한 SaaS 클라우드 솔루션을 만들 수도 있다.

8.2.3 델파이 배포 관리자

RAD스튜디오 **배포 관리자**는 Windows 애플리케이션과 멀티-디바이스 애플리케이션을 배포할 때 개발환경 안에서 타겟 플랫폼을 지정할 수 있다.

Windows, 맥OS, iOS로 배포할 때, 개발환경의 배포 관리자는 플랫폼 어시스턴트(PA서버 설치 파일은 C:\Program Files (x86)\Embarcadero\Studio\20.0\WPAServer folder에 있다)와 함께 작동한다. 타겟 장비로 애플리케이션을 배포할 때에는 PA서버와 해당 연결 프로파일을 사용해야 한다.

안드로이드로 배포할 때에는, RAD스튜디오에 안드로이드 SDK를 추가하면 로컬 컴퓨터 또는 원하는 안드로이드 장비로 애플리케이션을 배포할 수 있다.

개발자는 배포 관리자를 통해서 배포할 파일을 조회, 추가, 삭제, 편집할 수 있다. 또한 타겟 플랫폼용 데이터베이스 드라이버와 같은 꼭 필요한 파일을 추가할 때에도 배포 관리자를 사용한다.

8.2.4 써드파티 설치 도구

애플리케이션을 만들고 사용자 정의 설치 파일을 만들고 싶다면, 설치 파일 생성 도구 중에서 하나를 선택하게 된다. 설치파일 생성도구 목록은 다음과 같다.

- InstallAware
<https://www.installaware.com/>
- Advanced Installer
<https://www.advancedinstaller.com/>
- InstallAnywhere
<https://www.flexerasoftware.com/install/products/installanywhere.html>
- Inno Setup
<http://www.jrsoftware.org/isinfo.php>
- Wix Toolset
<https://wixtoolset.org/>
- InstallShield
<https://www.flexerasoftware.com/install/products/installshield.html>
- NSIS
https://nsis.sourceforge.io/Main_Page
- InstallForge
<https://installforge.net/home/>

8.2.5 온라인 스토어에 애플리케이션 제출

개발자는 Windows, 맥OS, iOS, 안드로이드 애플리케이션을 해당 온라인 앱 스토어에 패키지를 만들어서 배포할 수 있다. 절차는 DocWiki 링크를 참고하자. 각 스토어에 애플리케이션을 제출하는 준비, 패키징 등의 과정이 설명되어 있다.

- **Windows 스토어에 앱 배포하기**
- **앱스토어 (iOS)에 앱 배포하기**
- **구글 플레이 스토어에 안드로이드 앱 배포하기**
- **맥 앱스토어에 앱 배포하기**
- **아마존 앱스토어에 안드로이드 앱 배포하기**

8.3 기술 파트너 생태계

델파이 팀은 처음부터 이미 컴포넌트, 라이브러리, 도구, 익스퍼트, 마법사, 템플릿, 문서, 튜토리얼, 도서 등 모든 것을 혼자 할 수 없다는 점을 잘 알고 있었다. 델파이 개발환경, 컴포넌트 라이브러리 등이 확장성 있게 설계된 이유이다. 해를 거듭하면서 전세계 기술 파트너 생태계가 크게 형성되었고 추가 개발 솔루션들이 광범위하게 지원되었다. 추가할 수 있는 도구, 컴포넌트, 기술 솔루션들이 제공되는 수많은 기술 파트너와 리포지토리 중 대표적인 몇 가지와 해당 링크를 소개한다.

- **TMS Software** - 1995년에 설립되어 컴포넌트, 프레임워크, 도구, 프로젝트 개발 용역, 컨설팅, 교육 사업을 하는 소프트웨어 개발사이다. 델파이용 TMS 제품군 중 일부는 다음과 같다.
 - **VCL, FMX, FNC, 웹, 인트라웹(IntraWeb)** 컴포넌트: 툴바, 그리드, 메뉴, 그래픽 UI 컨트롤, 클라우드, 차트, 지도, 비동기 처리 등을 컴포넌트로 제공한다.
 - **TMS Aurelius**는 델파이용 ORM 프레임워크이다. 데이터 다루기, 복잡한 쿼리, 상속, 다형화(polymorphism) 등을 모두 지원한다.
 - **TMS Data Modeler**는 데이터베이스 모델링을 쉽고 간편하게 할 수 있는 도구이다. 다이어그램, 역공학(reverse engineering), SQL 스크립트 생성, 버전 관리 기능 등이 뛰어나다. 델파이, C++ 빌더 등 개발툴과도 부드럽게 통합된다.
 - **TMS Echo**는 데이터 복제용 델파이 프레임워크이다.
 - **TMSFixInsight**는 개발자들이 소스 코드 프로젝트 안에서 이슈를 탐지할 수 있도록 하는 델파이 코드 분석기이다.
- **Woll2Woll** - 1993년에 설립되어, RAD스튜디오 애드온을 제공한다. 견고하고 전문적인 기술로 각종 수상을 한 InfoPower, 1stClass, FirePower 비주얼 컴포넌트 스위트는 델파이와 C++ 빌더를 사용하여 데스크톱 애플리케이션과 모바일 애플리케이션을 개발하는 데이터베이스 개발자들에게 강력한 능력을 주도록 특별히 설계되었다.
 - **InfoPower**는 델파이 VCL용 라이브러리이다. 데이터베이스 애플리케이션의 프론트 엔드를 개발할 때 가장 널리 사용되는 컴포넌트 스위트이다.
 - **Firepower**는 RAD스튜디오 파이어몽키에서 사용된다. 매우 직관적이고 강력한 컴포넌트로서 탁월한 장점이 특징이다.
- Steema Software의 TeeChart는 델파이용 차트 부문에서 명실상부한 리더이다. 7년 연속 인포맨트 그룹으로부터 최고 델파이 차트와 지도 도구에 선정된 것으로 유명하다. 여전히 엠바카데로 (구. 볼랜드)가 델파이, C++ 빌더, HTML5 빌더, RAD스튜디오 개발환경에 번들로 넣고 있는 유일한 네이티브 VCL 차트 도구이다.
 - **TeeChart Pro** 차트 컴포넌트 라이브러리: VCL용과 FMX용 모두 제공되며 데이터를 수백가지 그래프 스타일과 2D와 3D로 표현할 수 있고, 56가지 수학, 통계, 재무 기능이 들어있다. 이 기능을 연결할 때 축(AXES) 수에 제한이 없으며 30가지 팔레트 컴포넌트와도 함께 사용할 수 있다.
 - **TeeGrid** 컴포넌트: 델파이 VCL용과 FMX용 모두 제공되며, 그리드의 속도가 매우 빠르다.

- FastReport는 보고서와 의사 결정 큐브 컴포넌트를 델파이 VCL용과 FMX용으로 제공한다. 델파이에는 FastReport 개발자 버전이 들어있다.
 - FastReport: 리포트를 빠르고 효율적으로 생성할 수 있는 애드온 컴포넌트이다. 비주얼 리포트 디자이너, 리포팅 핵심 모듈, 미리보기 창 등 보고서 개발에 필요한 모든 도구가 들어있다. 델파이 VCL용과 FMX용 델파이 버전이 있다.
 - FastCube: OLAP 컴포넌트 세트이다. 개발자가 데이터를 분석하고, 요약 테이블 (Data Slices)을 만들 뿐만 아니라 다양한 보고서와 그래프를 손쉽게 바로 만든다. 데이터 배열을 효율적으로 분석할 때 간편하게 사용할 수 있는 도구이다. 델파이 VCL용과 FMX용 델파이 버전이 있다.
- /nSoftware IPWorks는 프로그래밍에 반영할 수 있는 인터넷 컴포넌트 세트를 전세계에서 가장 광범위하게 제공한다. IPWorks는 외부와 연결되는 애플리케이션을 만들 때 생기는 복잡한 과정 대부분을 없앤다. 이메일 보내기, 파일 전송, 인터넷 탐색, XML 웹서비스 사용 등등을 손쉽게 프로그래밍할 수 있는 컴포넌트들이 들어있다.
 - 겐잇 패키지 매니저에서 개발자 에디션을 받을 수 있다.
 - 네이티브 델파이 VCL 컴포넌트 풀세트에서는 암호화, 블루투스LE, 웹소켓, IPC, MQ, 사물인터넷(IoT), OFX 등이 지원된다.
- **Atozed Software**는(아토즈드라고 발음한다) 인트라웹(IntraWeb) 컴포넌트 세트를 개발한다. 개발할 때는 데스크톱 애플리케이션을 만들 때와 같은 프로젝트를 사용하지만, 배포 시에는 HTML과 자바스크립트로 웹에 배포된다. (별도 플러그인 필요없음)
- **Scooter Software**의 Beyond Compare는 파일과 폴더를 쉽고 빠르게 비교할 수 있도록 한다. 개발자는 단순하면서 강력한 명령을 사용하여 관심 있는 차이점에만 집중하고 다른 것들은 무시할 수 있다. 변경 사항을 병합하고, 파일을 동기화하고 관련 보고서를 생성할 수도 있다.
- **Raize Software**는 1995년부터 혁신적인 고품질 컴퓨터 소프트웨어를 만들어오고 있다. Konopka Signature VCL controls 에는 200여가지 Windows UI 컨트롤과 디자이너가 들어있다. RAD스튜디오의 보너스 팩에도 포함되어 있어서 겐잇 패키지 매니저에서 다운로드 할 수 있다. 레이즈 소프트웨어는 CodeSite 로깅 시스템도 제공하고 있는데, 개발자들이 코드가 어떻게 실행되는지에 대해 통찰력을 가질 수 있도록 하고, 문제되는 지점을 더 빠르게 찾고 코드가 얼마나 건강하게 작동되는지를 모니터링할 수 있다.
- **Project Indy**는 오픈 소스 프로젝트이다. VCL과 FMX 애플리케이션용으로 인터넷 컴포넌트를 제공한다. 인디에는 TCP, UDP, 에코(Echo), FTP, HTTP, 텔넷(Telnet) 등등 수많은 인터넷 프로토콜을 사용할 수 있는 클라이언트/서버 컴포넌트가 들어있다.

- **Project JEDI** (Joint Endeavor of Delphi Innovators)는 오픈 소스 프로젝트이다. VCL 컴포넌트, 런타임 라이브러리, 도구, Windows API 인터페이스 유닛 등을 제공한다.
 - **JEDI Visual Component Library (JVCL)** 는 델파이와 C++빌더 프로젝트에서 재사용할 수 있는 비주얼 컴포넌트와 외형이 없는 기능 컴포넌트가 모여있는 (현재 600여개에 이르는) 방대한 컬렉션이다. JEDI VCL 라이브러리는 깃 패키지 매니저에서 받아서 설치할 수 있다.
 - **JEDI Code Library (JCL)** 는 유틸리티 함수와 외형이 없는 기능 클래스 세트이다. 철저한 테스트를 거쳤고 문서화까지 완전히 잘 되어있다. 델파이와 C++빌더 프로젝트에서 재사용할 수 있다.
- **Torry's Delphi Pages**에는 수만개에 이르는 델파이 관련 회사, 컴포넌트, 라이브러리, 도서, 예제 등이 연결되어 있다.

8.4 교육 – 강의, 도움말(Help), 코드 샘플

델파이 학습, 문제 해결 방법, 애플리케이션 개발 방법 등의 자료를 확인할 수 있는 웹사이트와 자료는 매우 많다. 델파이 지식을 제공하고 개발 생산성 향상에 도움이 되는 수많은 웹사이트 중에서 대표적인 곳 몇 개만 정리해 놓았다.

8.4.1 LearnDelphi.org

LearnDelphi.org 는 오브젝트 파스칼 델파이로 코드를 작성하는 방법을 제공한다. 오브젝트 파스칼은 좋은 프로그래밍 방법을 가르치기 위해 고안되었다. 델파이는 파스칼에서 가장 좋은 점들을 가져와서 까다로운 애플리케이션을 모든 플랫폼에서 구현할 때 필요한 기술과 능력을 만들어 냈다. 이 웹사이트에는 소프트웨어를 가르치고 배우는 사람들을 위한 리소스와 무료 자료들이 가득하다.

8.4.2 엠바카데로 아카데미

엠바카데로 아카데미는 각자 알맞은 진도에 맞게 학습할 수 있는 강좌들을 제공한다. 업계에서 가장 해박하고 경험이 많은 개발자들이 제공하는 강좌들이다. 엠바카데로 아카데미는 프로그래밍 언어, 도구, 프레임워크 등 다양한 영역을 다루는 것이 특징이다. 현재 제공되고 있는 코스는 RAD스튜디오, 델파이, C++빌더, 인터베이스, 센차 등 엠바카데로 제품들 관련 언어와 기량에 대한 것들이 중심이다. 지금까지 대부분은 델파이와 C++을 중심으로 하는 강좌들이지만, 시간이 가면서 훨씬 다양한 프로그래밍 기량과 언어들을 다루는 코스가 더 많아질 것이다.

각 코스별로 유료 수강이 가능하고, 여러 주제를 묶어 놓은 관련 코스들을 한 번에 신청할 수도 있다. 대부분이 비디오로 스트리밍된다. 실시간 강좌 또한 앞으로 추가될 것으로 보인다.

엠바카데로 아카데미에는 부트캠프와 컨퍼런스 다시보기가 무료로 제공되며, 일부 강사가 직접 할인 쿠폰을 제공하는 경우도 있다. 쿠폰은 강사의 블로그, 소셜미디어, 기타 웹사이트 등에서 강사로부터 직접 받으면 된다. 할인 쿠폰은 가끔 제공되는데, 엠바카데로 아카데미의 **Deals** 페이지에도 공지된다.

8.4.3 개발자 스킬 스프린트

개발자 스킬 스프린트는 바로 적용 가능한 프로그래밍 팁, 기교, 기술 등을 다루는 20분 이내의 짧은 라이브 강의를 녹화한 것이다. 주로 Windows, 맥OS, 안드로이드, iOS 소프트웨어 개발 관련 주제를 다룬다.

<http://embt.co/skill-sprints-youtube>

8.4.4 유튜브 – 웨비나 다시 보기 및 기타 비디오

엠바카데로에서 제공하는 수백개의 웨비나 다시보기, 새 버전 소개, 다양한 교육과 정보를 전달하는 비디오가 여러 플레이 목록별로 분류되어 있다.

<https://www.youtube.com/user/EmbarcaderoTechNet/playlists>

8.4.5 How-To 비디오

How-To 비디오는 특정 기능과 기술을 전달하는 짧은 튜토리얼 비디오이다. RAD스튜디오, 델파이, C++ 빌더, 인터베이스에 대한 강의가 제공된다.

<https://embt.co/TutorialVideos>

8.4.6 엠바카데로 개발자 블로그

엠바카데로 제품 관리자들과 커뮤니티 리더들이 블로그를 통해 델파이 개발과 관련된 다양한 주제에 관한 기술 아티클들을 올린다. 엠바카데로 개발자 블로그에 가보자.

<https://community.idera.com/developer-tools/b/blog>

8.4.7 엠바카데로 소셜미디어

엠바카데로 팀은 소셜 미디어에서도 매우 활발하게 활동하고 있다. 뉴스, 팁, 이벤트 등등을 팔로우 할 수 있는 링크는 다음과 같다.

- 트위터
<https://twitter.com/EmbarcaderoTech>
- 페이스북
<https://www.facebook.com/embarcaderotech>
- LinkedIn
<https://www.linkedin.com/company/7548>

델파이 개발환경을 열면 웰컴 페이지 우측에 예정된 세미나와 엠바카데로 유튜브 채널에 새로 추가된 비디오 목록을 볼 수 있다.

8.4.8 델파이로 만들어진 프로그램 사례

누가 델파이를 사용하고 있는지, 어떤 애플리케이션을 만들고 있는지 궁금한가? “델파이로 개발된 애플리케이션” 목록을 제공하는 웹사이트를 소개한다.

- 델파이로 개발된 품질이 높은 애플리케이션들
https://delphi.fandom.com/wiki/Good_Quality_Applications_Built_With_delphi
- 델파이로 개발된 유명한 소프트웨어들
<https://jonlennartaasenden.wordpress.com/2014/11/06/famous-software-made-with-delphi/>
- 델파이로 개발된 멋진 애플리케이션들
<https://beyondvelocity.blog/2019/05/19/awesome-applications-built-in-delphi/>
<https://beyondvelocity.blog/2019/07/28/delphi-mobile-its-all-components/>
- 쿨 앱 수상작 – 유튜브 게시
<https://www.youtube.com/playlist?list=PLwUPJvR9mZHhHEPxc8i9RCNX4jTyBZOfw>
- 한글로 소개된 개발 사례들
- <https://tech.devgear.co.kr/447417>

8.4.9 코드 예제, 튜토리얼 및 샘플

온라인 도움말 DocWiki에서 수많은 델파이 코드 샘플들을 찾을 수 있다. 델파이가 설치된 폴더 C:\Users\Public\Documents\Embarcadero\Studio\20.0\Samples에도 코드 샘플이 많이 있다. (참고 : 위 경로 중 20.0은 10.3 Rio 버전의 지정 폴더이다. 이 숫자가 작을수록 더 오래전 버전이 설치된 것이다. 예를 들어 10.2 버전의 폴더 경로 숫자는 19.0 이다.)

http://docwiki.embarcadero.com/CodeExamples/en/Code_Examples_Index

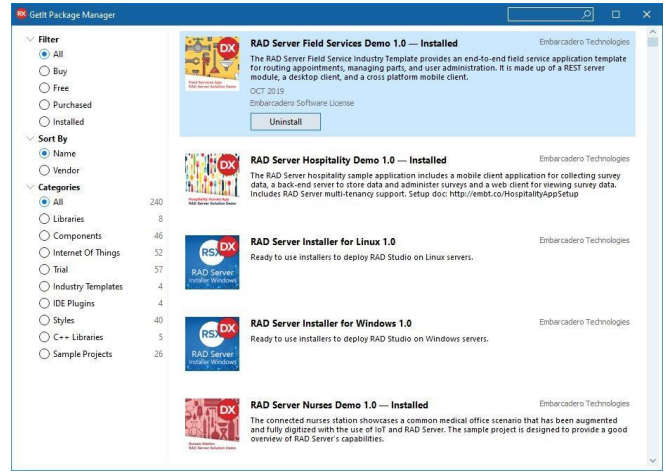
8.4.10 RAD스튜디오 개발환경 도구 API (Tools API)

RAD스튜디오 개발환경은 원하는 대로 확장하거나 커스터마이징할 수 있다. 메뉴 항목, 툴 바의 버튼들, 동적 폼 생성 마법사, 새 컴포넌트 등을 반영하려면 RAD스튜디오의 도구 API (Tools API)를 사용하자. 도구 API(Tools API) 는 개발환경을 다루거나 상호작용에 필요한 100여가지 인터페이스가 모여 있는 세트이다. 이를 통해 메인 메뉴, 툴 바, 메인 액션 리스트, 이미지 리스트, 소스 에디터의 내부 버퍼, 키보드 매크로와 바인딩, 폼 에디터 안에 있는 폼과 해당 컴포넌트들, 디버거, 디버깅 중인 프로세스, 코드 자동 완성, 메시지 뷰, 할 일 목록 등을 제어할 수 있다. 개발환경의 도구 API (Tools API)에 대해 더 알고 싶다면? 개발팀의 생산성을 향상시키기 위해 기능 추가가 필요하다면? 개발환경 확장 팩, 익스퍼트, 템플릿, 마법사를 만드는 방법을 알고 싶다면? 다음 웹사이트에서 오픈 도구 API에 대한 추가 정보를 찾을 수 있다.

- <http://www.gexperts.org/open-tools-api-faq/>
- <https://www.embarcadero.com/creating-ide-extensions-with-the-delphi-open-tools-api>
- https://www.davidghoyle.co.uk/WordPress/?page_id=1110

8.4.11 깃잇 패키지 매니저

RAD스튜디오에 있는 깃잇 패키지 매니저에서 패키지를 검색, 다운로드, 구입 및 설치할 수 있다. 패키지에는 라이브러리, 컴포넌트, 개발환경 확장, SDK 등이 들어있다. 깃잇은 메인 메뉴에서 Tools | GetIt Package Manager 메뉴를 선택하면 된다.



8.4.12 델파이용 깃허브 리포지토리

<https://github.com/search?q=delphi>

엠바카데로, 기술 파트너, 커뮤니티 회원들은 깃허브(GitHub)에 프로젝트들을 호스팅하고 있다. 9,600 개 이상의 솔루션, 컴포넌트, 프로젝트, 다양한 정보 등이 제공된다.

8.4.13 스택오버플로우 Q&A

델파이 태그를 사용하여 스택오버플로우에서 최신 질문과 답변을 찾아보자. 델파이 개발팀, MVP, 커뮤니티 회원들이 실시간으로 개발자 질문에 답변을 달고 있다.

<https://stackoverflow.com/questions/tagged/delphi>

8.4.14 플랫폼 지원 확인 페이지

델파이 특정 버전에서 확실하게 지원하는 플랫폼일까? 특정 플랫폼이나 버전에 알려진 이슈 또는 우회 방법이 있을까? 엠바카데로 온라인 DocWiki에는 RAD Studio의 다양한 버전별로 지원하는 플랫폼과 운영체제가 나와있다. 운영체제 옆에 * (별표) 기호는 해당 운영체제가 해당 RAD스튜디오 버전에서 알려진 문제가 있다는 표시이다. 해당 이슈에 대한 우회 방법을 보려면 운영체제 이름을 클릭하거나 해당 섹션으로 스크롤 한다.

http://docwiki.embarcadero.com/PlatformStatus/en/Main_Page

RAD Studio Version	Windows	macOS	iOS	Android	Linux
10.3 Rio	<ul style="list-style-type: none"> Windows 10 Windows 8.1 Windows 7 (SP1+) Windows Server 2016 Windows Server 2012 	<ul style="list-style-type: none"> macOS Catalina^[1] macOS Mojave macOS High Sierra macOS Sierra 	<ul style="list-style-type: none"> iOS 13^[2] iOS 12 iOS 11 iOS 10, including iOS 10.3 	<ul style="list-style-type: none"> Android 10^[3] Android 9 Android 8 Android 7 Android 6 Android 5.1 	<ul style="list-style-type: none"> Ubuntu 18.04 LTS Ubuntu 16.04 LTS Ubuntu 14.04 LTS RedHat Enterprise Linux (version 7)

비디오 - Windows 10과 VCL 용

9. 비디오 – Windows 10과 VCL 관련

델파이로 Windows 애플리케이션 만들기, 애플리케이션을 현대화하여 Windows 10에 적용하기, 델파이 배우기, Windows 개발에 필요한 다양한 영역의 기술 향상하기 등등 수백가지 비디오가 주제별로 제공됩니다.

9.1 Windows 10 관련 앱 현대화 비디오 목록과 링크

- 주요 Windows 10 기능을 VCL 애플리케이션에 반영하기
<https://www.youtube.com/watch?v=7KuzjODgR3c>
- Windows용 네이티브 앱 개발: 왜 중요한가, 세르지 필코 (엠바카데로 MVP)와 함께
<https://www.youtube.com/watch?v=4SL01nQbYKM>
- Windows 애플리케이션을 Windows 7에서 Windows 10으로 전환하기 (마이크로소프트와 엠바카데로)
<https://www.youtube.com/watch?v=3xKMGbUmwUs>
- WinAPI, COM & ShellAPI, WinRT를 VCL에서 사용하기
<https://www.youtube.com/watch?v=mM6sN6KS12U>
- Windows 10 고해상도 DPI로 빠르게 마이그레이션 (Denis Zubov와 함께)
<https://www.youtube.com/watch?v=niCFI8kgoOY>
- Windows 10과 RAD스튜디오 개발자를 위한 새 UX 디자인 원칙 (Sarina DuPont와 함께)
<https://www.youtube.com/watch?v=Qx71ytekamU>
- Windows 10의 Windows 런타임(WinRT) 다루기
<https://www.youtube.com/watch?v=bQI9GQB5y54>
- Windows 10 현대화 – RAD스튜디오와 TMS 활용
https://www.youtube.com/watch?v=sU0x3sn-m_o
- Windows 10 데이터베이스 액세스용 파이어닥(FireDAC)
<https://www.youtube.com/watch?v=rq16Lc1ZZOY>
- 기존 앱을 마이크로소프트 스토어에 배포하기 (Olaf Monien과 함께)
<https://www.youtube.com/watch?v=ODVHne9GOFY>
- 리눅스용 Windows 서브시스템(WSL)으로 델파이 리눅스 앱 배포하기
<https://www.youtube.com/watch?v=AD9XG4Y7MwA>
- 고해상도 DPI UI 디자인을 VCL에 반영하기
<https://www.youtube.com/watch?v=Q5mu8yigcK4>
- Windows 10 VCL
<https://www.youtube.com/watch?v=LbAEWketmMg>
- Windows 10 API를 활용하여 더 많은 사용자를 유치하기
https://www.youtube.com/watch?v=lvbQ9Hmfu_4

9.2 델파이 개발환경 기능과 생산성 관련 비디오 목록과 링크

- 처음으로 Windows용 VCL 애플리케이션 만들기 (델파이)
<https://www.youtube.com/watch?v=7oBTZ16dbc4>
- VCL 개발자를 위한 라이브바인딩
<https://www.youtube.com/watch?v=SVpbY4et2ug>
- 라이브바인딩을 VCL 애플리케이션에서 사용하기
https://www.youtube.com/watch?v=i9_6CJi5EIM
- VCL 스타일을 사용하여 멋진 모양을 갖춘 Windows 애플리케이션 만들기
<https://www.youtube.com/watch?v=m3RyP6AW9y0>
- VCL 스타일 활용 방법
<https://www.youtube.com/watch?v=c6E3DZrEdtQ>
- VCL – The best bits - 다크 모드 반영하기
<https://www.youtube.com/watch?v=tH61SOVBfW8>





델파이, 25년 혁신의 기록



10. 델파이, 25년 혁신의 기록

지난 25년간 출시된 델파이 버전들의 핵심 기능들을 간략히 정리해 놓았다. 델파이의 역사, 개발 경험, 개발자 이야기를 살펴볼 수 있다.

델파이에 대한 더 많은 정보: <https://devgear.co.kr/archives/products/delphi>

델파이 1 – 1995.2.15

- 네이티브 코드 최적화 컴파일러
- 디버깅 내장
- 16비트 Windows 3.1 지원
- 양방향 비주얼 도구
- 컴포넌트, VCL
- 데이터베이스 지원 기능 – BDE, SQL Links
- 데이터베이스 데이터를 개발 화면에서 실시간 확인

델파이 2 – 1996.2.10

- 32비트 Windows 95 지원
- 데이터베이스 그리드(Grid)
- OLE 자동화
- 비주얼 폼 상속
- 긴 문자열 지원
- 16비트 개발을 위해 델파이 1 포함

델파이 3 – 1997.8.5

- 인터페이스 (COM 기반)
- 코드 인사이트
- 컴포넌트 템플릿
- DLL 디버깅
- 웹브로커
- ActiveForms
- 컴포넌트 패키지
- MIDAS 멀티-티어 아키텍처

델파이 4 – 1998.6.17

- 도킹(Docking), 앵커(Anchors) 프로퍼티, Constraints 프로퍼티
- 메소드 오버로딩
- 동적 배열
- Windows 98 지원

델파이 5 – 1999.8.10

- 데스크톱 레이아웃
- 프레임
- XML 지원
- ADO용 DBGo
- 메뉴 다국어 지원

델파이 6 – 2001.5.21

- 구조 창 (Structure Window)
- SOAP 웹 서비스
- dbExpress
- BizSnap
- WebSnap
- DataSnap

델파이 7 – 2002.8.9

- RAD 웹 애플리케이션 개발
- 웹 서비스
- UML 디자이너
- 모델 기반 개발
- 닷넷(.NET) 상호운영 가능
- Windows XP 테마

델파이 8 – 2003.12.22

- 닷넷(.NET) 지원

델파이 2005 – 2004.10.12

- 멀티 유닛 네임스페이스
- 에러 인사이트
- 히스토리(History) 탭
- For..in 반복문
- Inline 지시자로 함수와 프로세서 성능 향상
- 개발환경에서 테마 적용 가능
- 리팩토링(Refactorings)
- USES 구문에 와일드 카드 사용
- 데이터 탐색기 (Data Explorer)
- 유닛 테스트 통합 (Integrated Unit Testing)

델파이 2006 – 2005.11.23

- 연산자 오버로딩
- 정적 매소드와 프로퍼티
- 디자이너 가이드라인
- Form Positioner 화면 추가
- Live 코드 템플릿
- 블록 완성
- 라인 숫자 표시
- 변경사항 확인 바
- 동시 편집 모드 (Sync Edit)
- 소스 코드 일부 접기, 메소드 네비게이션
- 디버깅 툴팁

- 툴 팔레트 검색
- FastMM 메모리 관리자 도입
- MySQL 지원
- dbExpress 유니코드 지원
- TTrayIcon
- TFlowPanel
- TGridPanel

델파이 2007 – 2007.3.16

- MS Build, 빌드 이벤트(Build Events), 빌드 설정(Build Configurations)
- Windows 비스타(Vista) 지원
- dbExpress 4 – 연결 풀링(Pooling) 프로퍼티 지정 가능, 델리게이트 드라이버
- CPU 조회 창
- FastCode 기능 향상
- 인트라웹(IntraWeb) / AJAX 지원
- 웰컴 페이지
- 영어, 프랑스어, 독일어, 일본어판 에디션 동시 제공

델파이 2009 – 2008.8.25

- 유니코드
- 제너릭(Generics)
- 익명 메소드 (Anonymous Methods)
- 리본 컨트롤
- 데이터스냅(DataSnap)
- 빌드 설정
- 클래스 탐색기 (Class Explorer)
- 타입 라이브러리 에디터 화면
- PNG 지원

델파이 2010 – 2009.8.15

- 개발환경 인사이트 (IDE Insight)
- 애트리뷰트 (Attributes)
- RTTI 중점 강화
- Direct2D 캔버스 추가
- Windows 7 지원
- 멀티터치/제스처

- 소스 코드 포매터
- 증분 검색 (Incremental Search)
- 특정 스레드에만 적용 가능한 브레이크 포인트
- 디버거 비주얼라이저
- IOUtils 유닛 (파일, 경로, 디렉토리용)
- 소스코드 감사(Audits)와 측정(Metrics)
- 백그라운드 컴파일
- MIDAS.DLL 소스 코드

델파이 XE – 2010.8.30

- 정규 표현식 라이브러리
- 버전 인사이트 – 서브버전 통합
- dbExpress – 필터, 인증, 프록시 생성, 자바스크립트 프레임워크
- REST 지원
- Indy 웹브로커
- 클라우드 – 아마존 EC2, 마이크로소프트 애저(Azure)
- 빌드 그룹
- 디버거에서 네임드 스레드
- 커맨드 라인 감사, 커맨드 라인 측정
- AQTime
- 코드사이트(CodeSite)
- 패스트리포트(FastReport), 파이널빌더(FinalBuilder), IP*Works!

델파이 XE2 – 2011.9.2

- 64비트 Windows 컴파일러 제공
- 맥OS X 지원
- 파이어몽키 (FMX) 플랫폼 추가
- 라이브바인딩 엔진 – FMX와 VCL 모두 지원
- VCL 스타일
- 유닛 범위 이름
- 플랫폼 어시스턴트(PA, Platform Assistant)
- 데이터스냅(DataSnap) 강화 - 모바일 장치 연동, 클라우드 API, HTTPS 지원, TCP 모니터링
- dbExpress ODBC 드라이버 지원
- 배포 관리자

델파이 XE3 – 2012.9.3

- 메트로폴리스 UI 추가 – Windows 8, 7, 비스타, XP용
- 맥OS X 마운틴 라이언 지원
- 비주얼 라이브바인딩
- 파이어몽키(FMX) 중점 강화
- 액션, 터치/제스처, 레이아웃 컴포넌트, 앵커(Anchors)
- 비트맵(Bitmap) 스타일
- TMaterialSource – FMX 3D 컴포넌트용
- 위치 센서 컴포넌트
- VCL/FMX 센서 디바이스 지원
- 가상 키보드 지원
- DirectX 10 지원

델파이 XE4 – 2013.4.22

- 파이어닥(FireDAC) 유니버설 데이터 액세스 컴포넌트
- 알림 기능
- 플랫폼 서비스
- 센서 컴포넌트 – 위치 정보, 움직임, 방향 센서
- TListView 컴포넌트
- 모바일 폼 디자이너
- TWebBrowser 컴포넌트
- iOS 지원
- 맥OS X 전체 화면 모드 지원
- 인터베이스 – IBLite, IBToGo

델파이 XE5 – 2013.9.11

- 안드로이드 개발 지원
- 알림 기능 컴포넌트
- iOS 7 스타일 지원
- 모바일 장치에 맞추어 원하는 대로 구성할 수 있는 폼 디자이너
- 안드로이드 장비용 배포 관리자
- REST 서비스 클라이언트 액세스와 인증 컴포넌트들

델파이 XE6 – 2014.4.15

- 애플터링 컴포넌트
- 클라우드 서비스(BaaS) RESTful 웹 서비스 통합 (Parse, Kinvey, App42)
- Taskbar 컴포넌트
- 새로운 VCL 스타일, VCL 센서 컴포넌트
- 모바일 앱 내 구매(In-App Purchase), 광고(In-App Advertising)
- 구글 글래스용 앱 개발
- 파이어몽키 데스크톱 액세스
- 파이어닥(FireDAC) DB 탐색기
- 아파치 지원 (웹브로커)
- DirectX 11 지원
- Windows 7, 8.1 스타일 지원
- 클라우드 기반 RESTful 웹 서비스 액세스 지원
- 파이어닥(FireDAC) 호환 가능 데이터베이스 확대
- 인터베이스(InterBase)와 완벽 통합

델파이 XE7 – 2014.9.2

- 멀티-디바이스 디자이너 – 기본 화면 및 커스터마이징 화면
- 파이어몽키 멀티-디바이스 애플리케이션 지원 (데스크톱, 모바일 플랫폼 모두)
- 파이어몽키 화면을 위한 장비 반응형 서비스
- 병렬 프로그래밍 라이브러리(PPL)
- IBLite 데이터베이스 플랫폼 임베디드 지원 – Windows, 맥, 안드로이드, iOS
- 멀티-디스플레이 지원
- TMultiView 컴포넌트
- 멀티-터치 지원, 제스처 업데이트
- 안드로이드 풀 스크린 몰입 모드 지원
- TListView에서 iOS, 안드로이드용 '당겨서 새로고침' 기능 지원
- FMX 상태 저장
- 인터베이스 IBLite – Windows, OSX 지원
- 엔터프라이즈 모빌리티 서비스 (EMS, 현재 RAD서버가 됨)
- 블루투스, 블루투스LE 런타임 라이브러리 제공
- TBluetoothLE 컴포넌트
- 애플터링 – 블루투스, IP 주소
- OmniXML 라이브러리
- Java2Pas 인터페이스 컨버터
- 깃(Git) 버전 콘트롤 지원

델파이 XE8 – 2015.4.7

- 갯잇 패키지 매니저(GetIt Package Manager)
- 파이어닥 (FireDAC) 강화
- 새 엠바카데로 커뮤니티 툴바
- iOS에서 네이티브로 표시되는 컨트롤 추가 – TListView, TSwitch, TMemo, TCalendar, TMultiView, TEdit
- 대화형 지도(Maps)
- 미디어 라이브러리에 새 옵션 추가
- InputQuery에서 마스킹 인풋 필드(Masking Input Fields) 지원

델파이 10 시애틀 – 2015.8.31

- 안드로이드 백그라운드 서비스 지원
- 파이어닥 NoSQL 몽고DB(MongoDB) 지원
- 파이어몽키 컨트롤 zOrder 지원- Windows용
- 새 TBeaconDevice 클래스 추가 – 블루투스LE 장비를 “비콘”처럼 바꾸어서 광고데이터(UUID, Major, Minor, TxPower)를 전송하도록 할 수 있다.
- Windows 10 스타일 제공
- 비트맵 스타일 디자이너에서 Windows 10용 스타일뷰어(StyleViewer) 지원
- High-DPI 인식, 4K 모니터 지원
- 개발환경에서 안드로이드 서비스 지원
- WinRT API 호출 지원

델파이 10.1 베를린 – 2016.4.16

- 안드로이드 6.0 지원
- Windows 데스크톱 브릿지 지원
- iOS용, 안드로이드용 주소록
- 새 리스트뷰 아이템 디자이너(ListView Item Designer) 추가
- 새 캘린더뷰(CalendarView) 컨트롤 추가
- VCL용 퀵에디트(QuickEdits)
- High-DPI Windows 지원
- 힌트 프로퍼티 변경
- RAD서버 아파치 서버 지원
- 갯잇 기반 웹 설치 (Installer)

델파이 10.2 도코 – 2017.3.22

- 64비트 리눅스 지원
- 파이어닥(FireDAC) 리눅스 지원 - 리눅스에서 작동되는 모든 DBMS
- 마리아DB(MariaDB) 지원 - v5.5
- MySQL 5.7 지원
- 파이어버드(Firebird) 직접 I/O
- FMX용 퀵에디트(QuickEdits)
- Windows10용 새 VCL 콘트롤 추가
- 개발환경 UI 업데이트 (다크테마)
- RAD서버 배포 라이선스 제공

델파이 10.3 리오 – 2018.11.21

- 새로운 델파이 언어 기능들 – 인라인 변수 선언, 타입 추론, 리눅스에서 AnsiString / AnsiChar 지원
- 파이어몽키(FMX)에서 안드로이드 zOrder, 네이티브 콘트롤, API 레벨 26 지원
- iOS 12 지원
- Windows 10, VCL, HighDPI 관련 기능 강화
- VCL에서 Per Monitor V2 지원
- 개발환경 UI 대폭 현대화 – 데스크톱 레이아웃, 개발환경 인사이트 콘트롤, 밝은 테마, 다크 테마 리비전
- RAD서버 아키텍처 확장
- RAD서버 - 새 엔드포인트(EndPoint) 애트리뷰트 추가 - Content-Type, Accept 기반 맵핑
- RAD서버 - HTTP 메소드에 커스텀 메소드를 연결할 수 있도록 애트리뷰트 추가
- RAD서버 - 요청 처리를 클래스 또는 컴포넌트로 위임
- 품질 및 성능 강화
- C++/델파이 ABI 호환

델파이 10.3.1 리오 – 2019.2.14

- iOS 12와 아이폰X 시리즈 지원 확장
- RAD서버 콘솔 UI 재설계, ExtJS 프레임워크로 마이그레이션 (갯잇에서 다운로드 가능)
- RAD서버 푸쉬 알림 지원 디바이스 확장
- 파이어닥(FireDAC) - 파이어버드(Firebird) 3.0.4, 파이어버드 임베디드 지원 강화
- Windows용 HTTP, SOAP 클라이언트 라이브러리 강화 – 압축 및 복호화 지원
- 개발환경 생산성 도구 추가: 북마크와 탐색기
- VCL, FMX용 새로운 스타일 15종 추가

델파이 10.3.2 리오 – 2019.7.18

- 맥OS 64비트 개발 지원
- RAD서버 마법사와 배포 기능 향상
- 구글 Firebase 푸쉬 알림 안드로이드 지원
- 델파이 리눅스 클라이언트 애플리케이션 개발 지원
- 런타임 라이브러리 성능 강화
- VCL 품질 중점 강화

델파이 10.3.3 리오 – 2019.11.21

- 안드로이드 64비트 개발 지원
- iOS 13, 맥OS 카탈리나(Catalina) 지원
- RAD서버 도커(Docker) 배포
- 엔터프라이즈 커넥터 무상 제공 (엔터프라이즈, 아키텍트 에디션에 포함)
- 파이어닥(FireDAC) 드라이버 - PostgreSQL, SQLite, DB2, 오라클 지원 기능 강화

델파이 10.4 시드니 – 2020.5.26

- 델파이 코드 인사이트 기능 강화
- 새로운 델파이 언어 기능들
- 델파이, C++이 지원하는 모든 플랫폼에 대한 통합 메모리 관리
- 새로운 VCL 컴포넌트들, 개별 스타일 적용, High DPI 스타일 등
- 대폭 강화된 델파이 멀티-디바이스 플랫폼 지원 기능들
- 업그레이드된 데이터베이스 지원 기능들
- 온라인/오프라인 용 통합 설치 프로그램
- 사용성 & 성능이 더 좋아진 IDE 기능들



저자 소개

저자 소개

데이비드 아이로도 잘 알려져 있는 데이비드 인터시모네(David Intersimone)는 혁신적이면서도 열정이 넘치는 소프트웨어 전문가이다. 특히 개발도구, 소프트웨어 개발, 소프트웨어 아키텍처 세계에 대한 애정이 크고 관련 교육도 하고 있다. 1985년 볼랜드 소프트웨어에 입사하며 DevRel (Developer Relations) 담당자가 되었다. 지난 47년간 소프트웨어 엔지니어, 개발 매니저, 개발자 커뮤니티 관리자, 개발자



변호인 등 다양한 역할을 수행하며 지속

성장 중인 세계적인 규모의 개발자 커뮤니티, 수 천 개의 기술자료들, 영상 등을 만들어왔다.

엠바카데로가 볼랜드 소프트웨어 개발도구 사업을 인수하기 전에는 20년 이상을 볼랜드에서 기술 전파, 엔지니어링, 개발 능력, DevRel 프로그램 개발 등 다양한 역할을 해왔고, 엠바카데로에 합류한 뒤로도 11년간 부사장으로서 DevRel 역할을 해왔다.

현재는 DevRel 분야의 선구자로서 얻은 노하우와 비전을 워크샵, 온라인 세미나, 가이드 등을 통해 프로그램 매니저, 관리자, 개발자 등 전문가들과 공유하고 있다. 그리고 엠바카데로 개발도구 분야 MVP(<https://www.embarcadero.com/embarcadero-mvp-program>)로서 개발자 커뮤니티에 기술 자료 기고, 강의, 온라인 세미나 진행을 하며 활발한 활동을 이어가고 있다.

최근에는 델파이의 상징적인 인물에게 수상하는 '2019 the Spirit of Delphi'로 선정되기도 했다. 이 상은 델파이와 개발자 커뮤니티에 기여한 공로를 바탕으로 후보자들을 선별하고, 엠바카데로 델파이 팀이 수상자를 선정한다.

엠바카데로 델파이와 Windows 개발 현황



데이비드 아이 (David Intersimone “David I”)

Copyright © 2020 Embarcadero Technologies, Inc