
C++Builder XE: C++Builder 6 이후의 새로운 디버그 기능들

Nick Hodges, 박지훈.임프

2010 년 9 월

Americas Headquarters
100 California Street, 12th
Floor
San Francisco, California
94111

EMEA Headquarters
York House
18 York Road
Maidenhead, Berkshire
SL6 1SF, United Kingdom

Devgear
서울특별시 반포 1 동 743-14
4 층 ㈜데브기어
(T) 02.595. 4288

Name Thread-ForDebugging

쓰레드 디버깅이 용이하도록 TThread 클래스에 NameThreadForDebugging 메소드가 추가되었으며, 이를 이용하여 IDE 의 Thread View 에 현재 디버깅중인 각 쓰레드들의 이름을 볼 수 있습니다.

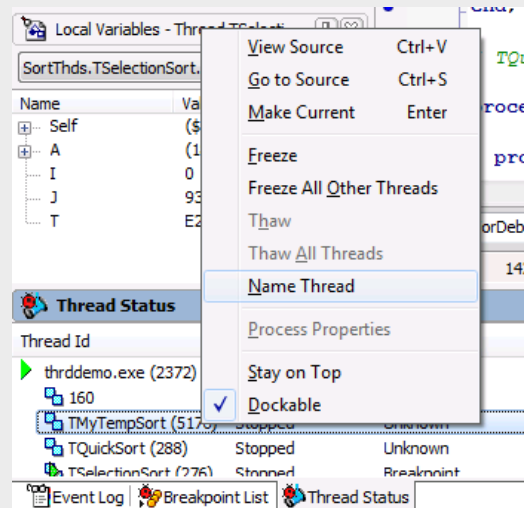
```

- /* The Execute method is called when the thread starts */
- //-----
-
- void __fastcall TSortThread::Execute()
- {
- 86  TThread::NameThreadForDebugging( (AnsiString)ClassName ());
-     Sort (&FSortArray);
- }

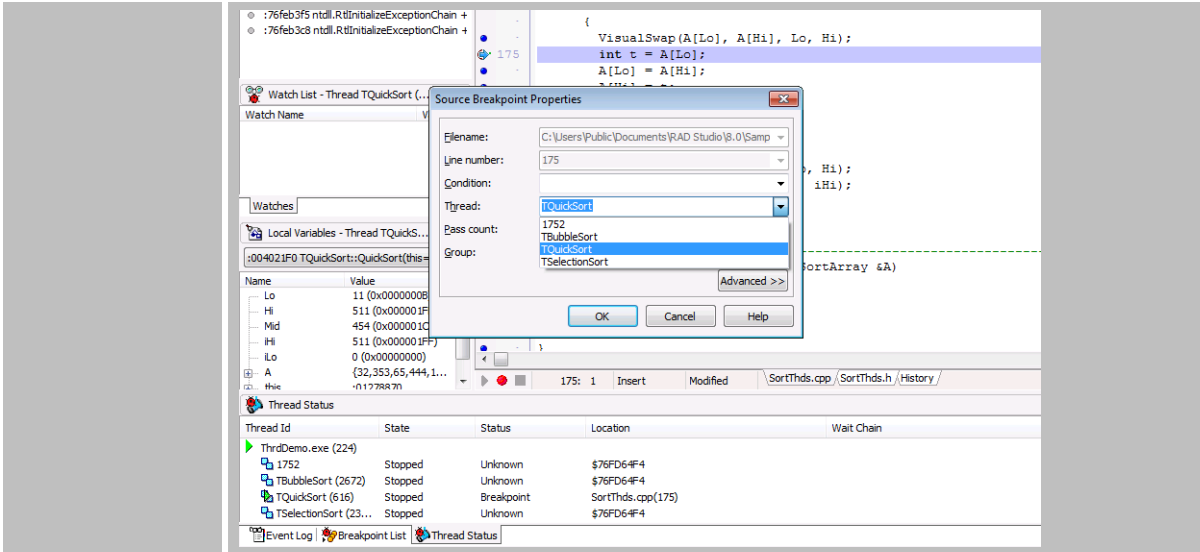
```

쓰레드에 임시 이름 지정 가능

코드에서 NameThreadForDebugging 메소드로 명시적으로 쓰레드 이름을 지정해주는 방법 외에도, C++ 빌더 XE 에서는 디버깅 진행 중에 특정 쓰레드에 임시 이름을 지정할 수 있습니다.

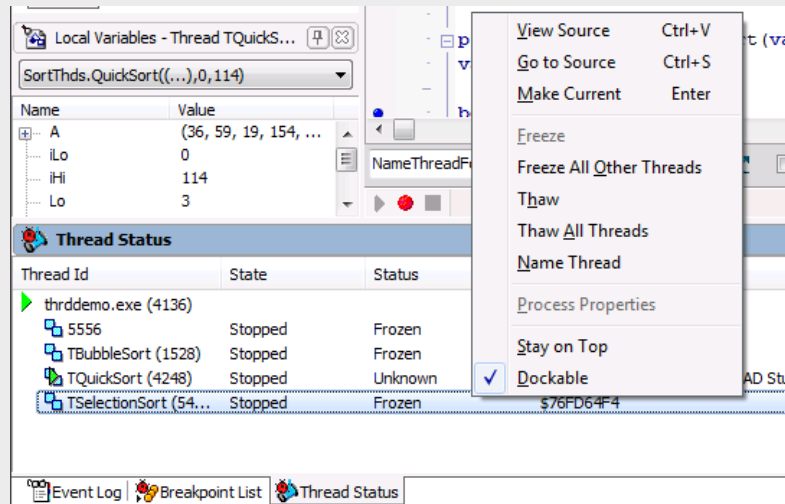
**Thread Specific Breakpoint**

특정 쓰레드만 디버깅하기 위한 Thread Specific Breakpoint 기능이 추가되었습니다.



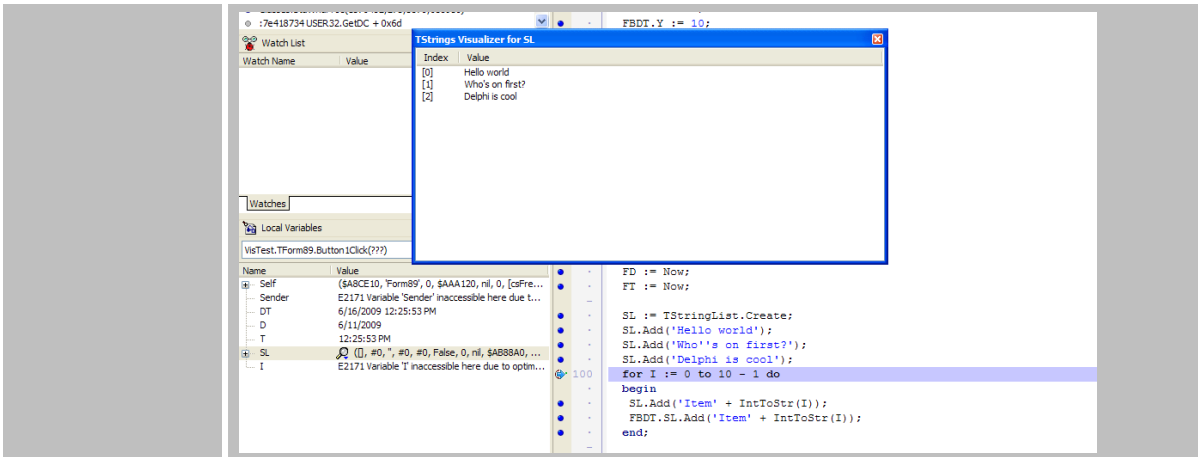
쓰레드 Freeze/Thaw

각 쓰레드를 일시적으로 얼리거나(Freeze) 다시 녹일(Thaw) 수 있어 원하는 쓰레드의 디버깅에 집중할 수 있습니다.



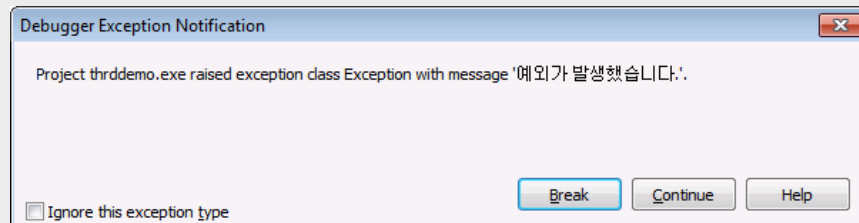
디버그 비주얼라이저

TDateTime, TStringList, std::string 및 std::wstring 등 기존에는 디버거에서 값을 추적할 수 없었던 타입들을 볼 수 있게 해주는 Debug Visualizer 가 추가되었습니다. 새로운 타입을 위한 비주얼라이저를 개발하여 등록할 수도 있습니다.



예외 다이얼로그 개선

C++ 빌더 6 이하 버전에서는 실행중 예외가 발생했을 경우 무조건 IDE 에서 실행이 멈추었지만(break), 이제는 Break 와 Continue 를 선택할 수 있게 되어 IDE 에서 해당 라인을 찾아가지 않고 곧바로 실행을 계속하도록 할 수 있게 되었습니다.

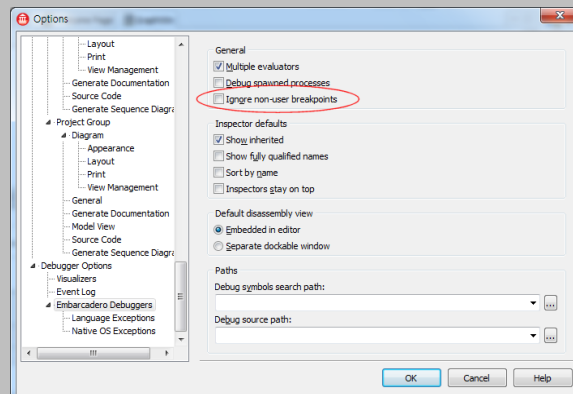


또한 다이얼로그 아래쪽에 있는 "Ignore this exception type" 옵션에 체크를 하면 IDE 가 해당 타입의 예외를 무시하도록 할 수 있습니다.

Ignore non-user breakpoints

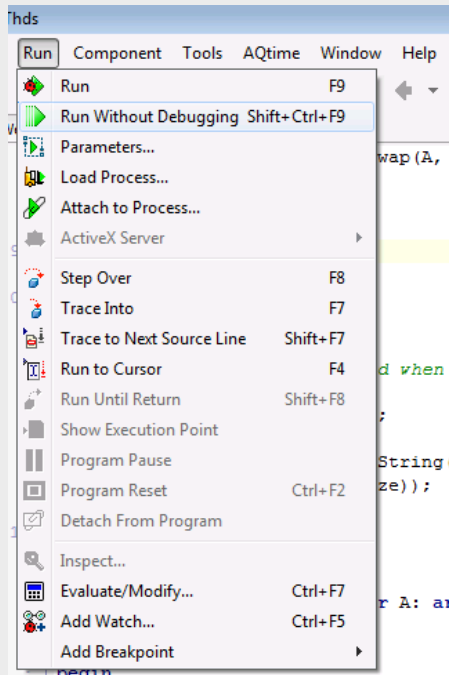
Tools | Options | Debugger Options | Embarcadero Debuggers 에 "Ignore non-user breakpoints" 옵션이 추가되었습니다. 이 옵션을 선택하면 개발자가 설정하지 않은 브레이크포인트(Non-user breakpoint)로 인한 IDE 일시 정지를 하지 않습니다.

Non-user breakpoint 는 DbgBreakPoint() API 함수 호출에 의해 코딩으로 지정된 것으로, 흔히 Windows OS 커널의 버그로 발생하며, 이런 의도하지 않은 코드에 의한 브레이크포인트를 무시할 수 있습니다.

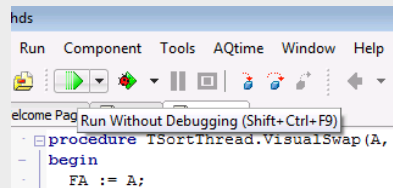


또한 C++ 빌더 XE 에서는 Non-user breakpoint 를 만날 경우 Event Log 윈도우에 표시합니다.

Run Without Debugging



IDE 에서 디버깅 모드로 프로젝트를 실행하는 Run 이외에, 디버그 모드가 아닌 일반 실행 모드로 프로젝트를 실행해주는 Run Without Debugging 명령이 추가되었습니다.



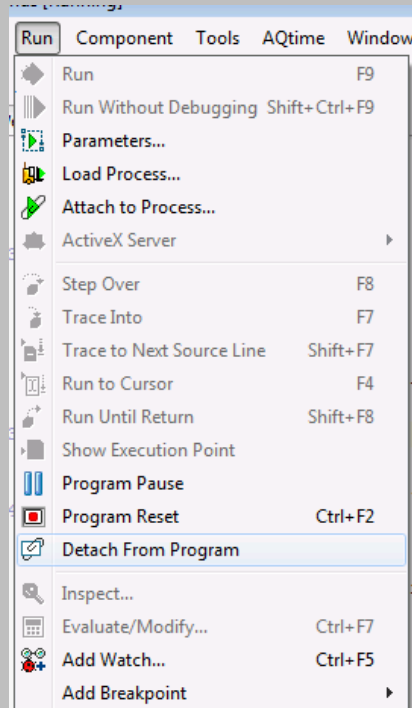
Run - Run Without Debugging 에서 실행할 수 있으며, 툴바에도 버튼으로 등록되어 있습니다.

예를 들면, 서버-클라이언트로 구성된 프로젝트 그룹에서 서버와 클라이언트

프로젝트를 서로 연동 테스트하려고 할 때 아주 편리합니다.

Detach from Program

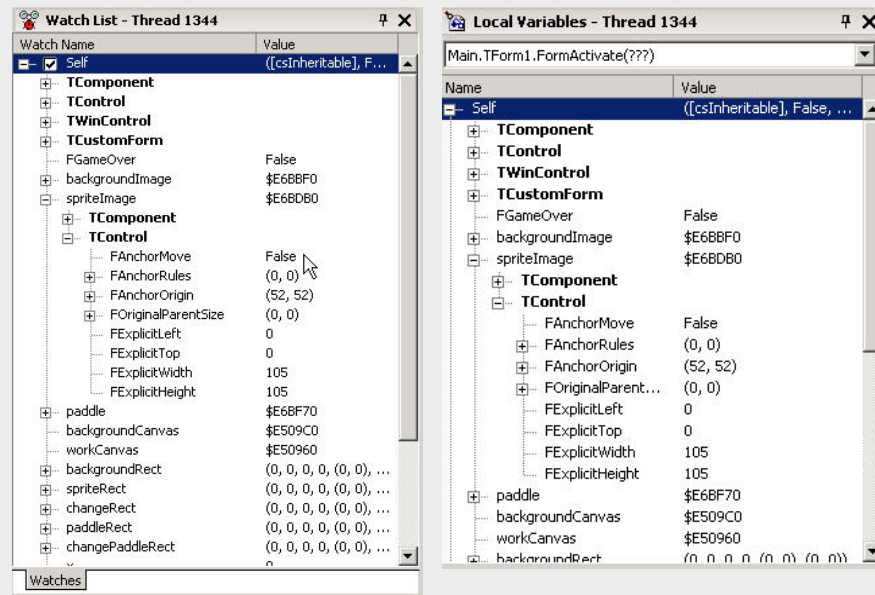
Detach from Program 명령은 실행중인 프로그램에 디버거를 연결하여 디버깅하는 Attach to Process 와는 정반대로, 디버깅 모드에서 실행중이던 프로젝트로부터 디버깅을 해제하고 일반 모드로 실행을 계속하도록 합니다.



**펼쳐지는
Watch, Local
Variables**

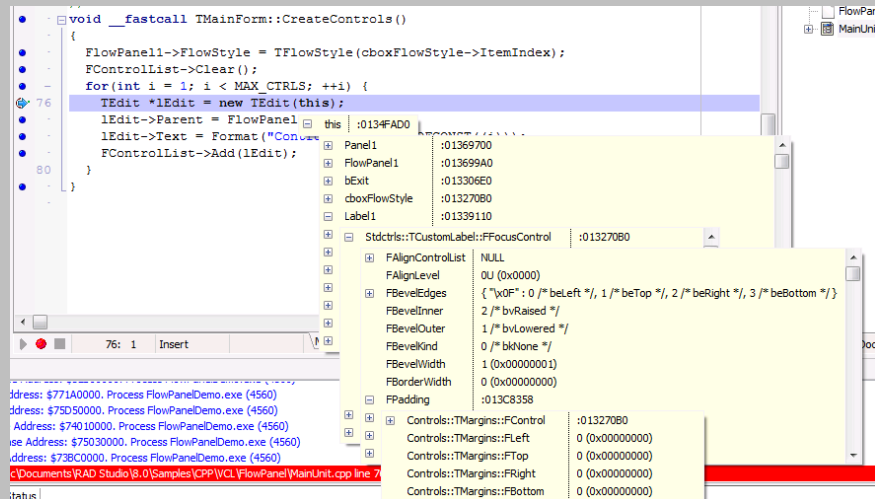
C++ 빌더 6까지는 Watch 나 Local Variables 윈도우에서 디버그 상태에서 객체의 값을 볼 수는 있었으나 그 멤버들로 드릴-다운 할 수가 없어 디버그 인스펙터를 써야 했습니다.

C++ 빌더 2006 버전부터는 Watch 와 Local Variables 윈도우에서 각 객체의 하위 멤버들의 값들을 드릴-다운하여 살펴볼 수 있게 되었습니다.



**펼쳐지는 디버그
툴팁**

C++ 빌더 2006 이후 버전부터는 디버그 툴팁에서 객체의 멤버들을 계속 드릴-다운하여 그 값들을 살펴볼 수 있게 되었습니다.



또한 Ctrl 키를 누르고 있는 동안 이 툴팁들이 투명하게 보이게 되어, 툴팁을 없애지 않고도 뒤의 코드 내용을 볼 수 있게 되었습니다.

**Wait Chain
Traversal**

Wait Chain Traversal(WCT)은 프로젝트의 어떤 스레드에서 데드락이 발생했을 때 그에 대한 자세한 정보를 찾아볼 수 있게 해줍니다. 스레드들 사이에 데드락이 발생했을 때, Thread Status 에 새로 추가된 Wait Chain 컬럼에 각

쓰레드들 사이의 상호 블로킹 상태를 표시해줍니다.

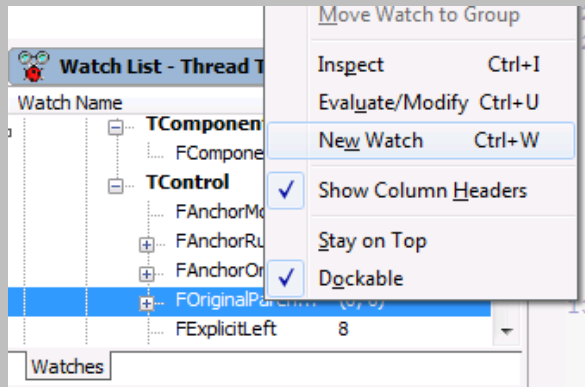
| Thread Id | State | Status | Location | Wait Chain |
|---------------------|---------|---------|------------|--|
| Project2.exe (3984) | | | | |
| 1284 | Stopped | Unknown | \$76EC0F34 | Blocked waiting for Thread 3792 |
| 2792 | Stopped | Unknown | \$76EC0F34 | Blocked on Critical Section owned by Thread 3792 |
| 3792 | Stopped | Unknown | \$76EC0F34 | Blocked on SendMessage call to a window owned by Thread 2792 |
| 916 | Stopped | Stepped | \$758173B4 | |

(WCT 기능은 Windows Vista 이상에서만 사용 가능하며, Wait Chain 컬럼도 Vista 이상에서만 나타납니다)

Watch 의 New Watch, Evaluate/ Modify 메뉴

Watch 윈도우에 추가된 객체의 하위 객체에서 오른쪽 클릭하여 팝업 메뉴에서 New Watch 를 선택하면, 해당 하위 객체가 Watch 윈도우에 추가됩니다.

또한, Evaluate/Modify 메뉴를 선택하면 해당 객체에 대한 Evaluate/Modify 창이 떠서 즉시 값을 변경해볼 수 있습니다.



기타 추가된 디버깅 기능들

다음과 같은 새로운 디버거 기능들이 추가되었습니다.

1. CPU 뷰가 에디터 탭의 바깥에 독립적으로 열릴 수 있습니다.
2. 디스어셈블리 창에 "Show Opcodes" 로컬 메뉴 항목이 추가되었습니다.
3. 디스어셈블리 창에 "Show Addresses" 로컬 메뉴 항목이 추가되었습니다.
4. Tools | Options | Debugger Options | Event Log 에 "Scroll new events into view" 옵션이 추가되었습니다.
5. 툴바를 커스터마이징할 때 "Notify on language exceptions" 버튼이 추가되었습니다. 이 버튼은 개발자들이 Tools | Options | Debugger Options | Embarcadero Debuggers | Language Exceptions 페이지에 있는 "Notify on Language Exceptions" 옵션을 빠르게 토글할 수 있게 해줍니다.
6. Call Stack 뷰에서 해당 프레임이 디버그 정보를 포함하고 있는지 보여주는 이미지를 보여줍니다.
7. Call Stack 뷰에서 지정된 스택 프레임 위치에서 개발자들이 브레이크 포인트를 설정할 수 있습니다.
8. Call Stack 뷰에서 한 항목을 더블 클릭하면 자동으로 로컬 뷰를 동기화해줍니다.
9. 에디터에서 CTRL-F5 핫키를 치면 현재 라인에 설정되어 있는 브레이크 포인트를 활성화/비활성화해줍니다.
10. 이제 C++ 빌더에서도 Project | Options | Debugger 다이얼로그에서 "Debug Source Path" 설정을 지정할 수 있습니다.

11. "Automatically close files implicitly opened while debugging" 옵션 설정에 CPU 뷰도 적용되게 되었습니다.



Embarcadero Technologies Inc.는 애플리케이션 개발자 및 데이터베이스 전문가가 자신이 선택한 환경에서 소프트웨어 애플리케이션을 설계, 빌드 및 실행하는 도구를 사용할 수 있도록 합니다. 전 세계 3 백만 이상의 커뮤니티와 Fortune 지 선정 100 대 기업 중 90 개 기업이 Embarcadero의 CodeGear™ 및 DatabaseGear™ 제품군을 기반으로 하여 생산성을 향상시키고 개방적인 협업 및 자유로운 혁신을 추구하고 있습니다. Embarcadero는 1993년에 설립되어 캘리포니아 샌프란시스코에 본사가 있으며 전 세계에 사무소를 두고 있습니다. Embarcadero의 온라인 주소는 www.embarcadero.com입니다.



데브기어는 미국 Embarcadero Technologies Inc.와 기존의 코드기어 한국 지사의 협력으로 전략적으로 설립된 엠바카데로 솔루션 전문 공급 기업입니다. 데브기어는 Delphi, C++Builder, JBuilder, Delphi Prism 등 개발툴 제품들과 ER/Studio, PowerSQL, DB Artisan, EA/Studio 등의 데이터베이스 툴 제품들에 대한 한국 시장에 공급은 물론 기술지원 및 교육 등의 기술 서비스를 제공합니다. 데브기어 웹 사이트는 <http://www.devgear.co.kr/>이며 제품에 대한 문의는 ask@embarcadero.kr로 하면 됩니다.