

사용자 지정 이벤트 공급자 샘플

*이 샘플은 2016 년 3 월 Xbox One XDK 이상과 호환됩니다.*

# 설명 이 샘플에서는 Xbox One에서 사용자 지정 ETW 이벤트 공급자를 사용하는 방법을 보여줍니다. 샘플 사용하기

이 샘플에서는 다음 컨트롤을 사용합니다:

|  |  |
| --- | --- |
| 액션 | 게임 패드 |
| 샘플을 종료하십시오. | 왼쪽 트리거 + 오른쪽 트리거 + 오른쪽 숄더 |

구현 정보  
이 샘플은 기존 Windows ETW 공급자와 동일한 구조를 따르지만 Exclusive 파티션에서 실행되는 제목은 이벤트 공급자를 레지스트리에 추가할 수 없으므로 생성된 이벤트 데이터를 올바르게 해결하기 위해 호스트 PC에 몇 가지 추가 단계가 필요합니다.

이벤트 매니페스트 **etwprovider.man**는 직접 또는 Windows SDK의 일부로 제공되는 ecmangen.exe 도구를 사용하여 편집할 수 있습니다. Visual Studio 메시지 컴파일러 (mc.exe)를 사용하여 리소스 파일 (**etwproviderGenerated.rc**) 및 헤더로 (**etwproviderGenerated.h**) 컴파일됩니다. 리소스 파일과 헤더 모두 제목 프로젝트에 포함됩니다.

타이틀 초기화 중, 이벤트 프로바이더는 **EventRegisterCEP\_Main**을 호출하여 등록됩니다, 그 후 종료 처리 중 **EventUnregisterCEP\_Main**을 호출하여 등록이 취소됩니다. "Mark" 이벤트는 단일 유니코드 문자열을 매개 변수로 하는 **EventWriteMark**에 대한 호출을 통해 발생됩니다.

샘플은 다른 샘플처럼 빌드, 배치 및 활성화할 수 있습니다. 일단 이벤트를 캡처하기 위하여 트레이스 로그를 사용하여 실행하고 있습니다 (xbperf는 사용자 정의 이벤트 프로바이더를 지정하는 기능이 부족합니다). 타이틀의 이벤트 프로바이더는 레지스트리에 추가되지 않았기 때문에 이름으로가 아니라 GUID를 통해 식별되어야 합니다 (GUID는 이벤트 매니페스트에 지정된 프로바이더 GUID와 일치해야 함):

C:\temp>xbrun /x/title /O tracelog -start CustomSession -f d:\custom03.etl -eflag PROC\_THREAD+LOADER+DPC+INTERRUPT+CSWITCH+PROFILE -guid #{A4A76336-4BA7-4CD9-85C3-B9C236D3041C} -stackwalk PROFILE+CSWITCH

필요한 데이터가 캡처되면 세션을 일반적인 방법으로 중지할 수 있습니다:

C:\temp>xbrun /x/title /O tracelog -stop CustomSession

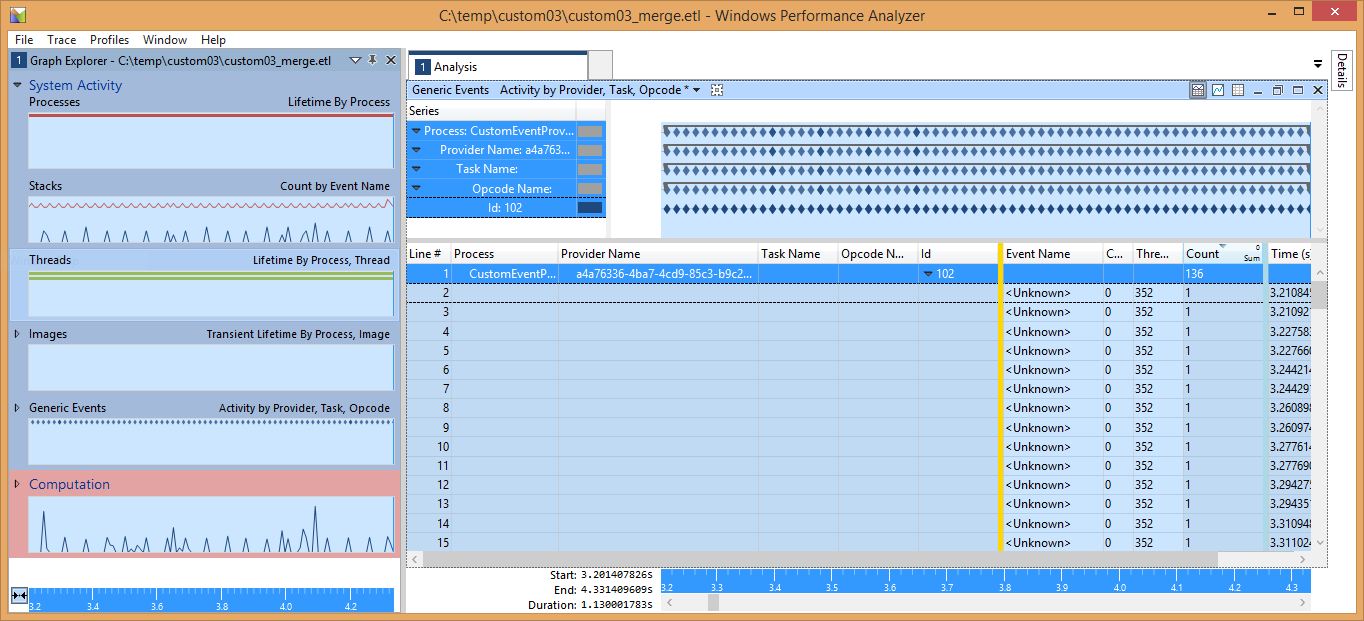
ETL 파일을 devkit에 병합하여 시스템 이벤트 프로바이더를 해결하세요. 이것은 사용자 정의 이벤트 프로바이더를 해결하지 *못*합니다:

C:\temp>xbrun /x/title /O tracelog -merge d:\custom03.etl d:\custom03\_merge.etl

이제 병합된 파일을 호스트 PC에 다시 복사할 수 있습니다:

C:\temp\custom03>xbcp /x/title xd:\custom03\_merge.etl .

이 파일은 WPA에 로드할 수 있으며 사용자 정의 이벤트는 "System Activity" 그룹의 "Generic Events"그래프에 나타납니다. 그러나 이 시점에서 이벤트는 GUID로만 식별되며 Task Name 및 Opcode Name과 같은 정보는 표시되지 않습니다. 더 중요한 것은 각 이벤트(유니코드 문자열)에 대해 제공한 사용자 정의 데이터도 표시되지 않습니다.



각 사용자 정의 이벤트에 대한 전체 정보를 표시하기 위해 우리는 트릭을 사용합니다. *우리는 devkit보다는 호스트 PC에 이벤트 프로바이더를 등록하고 이벤트를 해결합니다.*

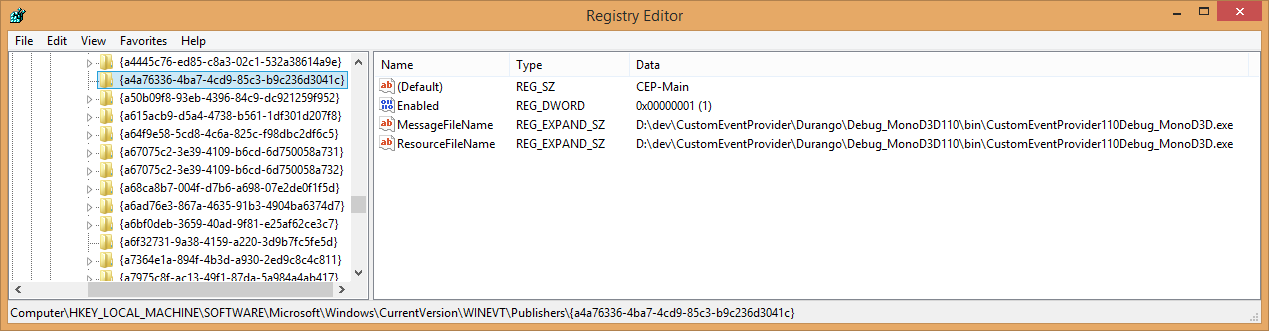
먼저 이벤트 매니페스트(evtprovider.man)의 프로바이더 노드를 편집하며 **resourceFileName** **messageFileName** 특성은 Xbox One 실행 파일이 빌드된 개발 PC의 위치를 가리킵니다:

<provider name="CEP-Main" guid="{A4A76336-4BA7-4CD9-85C3-B9C236D3041C}"   
symbol="CEP\_MAIN"   
resourceFileName="D:\dev\CustomEventProvider\Durango\Debug\_MonoD3D110\bin\CustomEventProvider110Debug\_MonoD3D.exe"   
messageFileName="D:\dev\CustomEventProvider\Durango\Debug\_MonoD3D110\bin\CustomEventProvider110Debug\_MonoD3D.exe">

그런 다음 상승된 명령 프롬프트에서 wevtutil.exe 도구를 실행하여 호스트 PC에 이벤트 공급자를 등록하세요:

D:\dev\CustomEventProvider>wevtutil im etwprovider.man

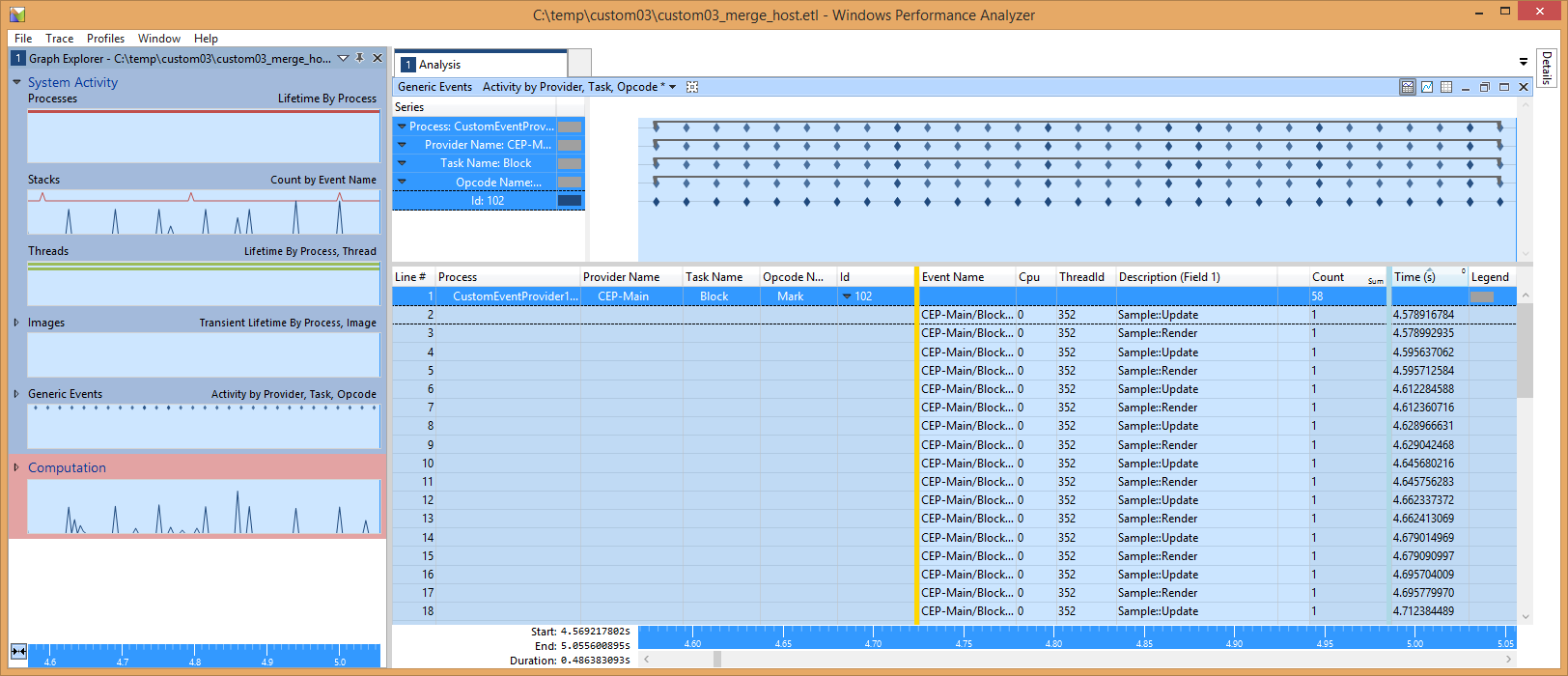
호스트 PC의 레지스트리를 체크인하면 HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Publishers에 나열된 프로바이더가 표시됩니다.



마지막으로 xperf를 사용하여 호스트 PC의 ETL 파일을 해결합니다:

C:\temp\custom03>xperf -merge custom03\_merge.etl custom03\_merge\_host.etl

호스트가 병합 된 ETL 파일이 WPA로 로드되면 이제 올바르게 해결된 이벤트가 표시됩니다:



Description(Field 1) 열에 이벤트와 함께 기록된 문자열이 포함되어 있는 방법을 확인하세요. Task 및 Opcode 이름도 볼수 있습니다.

성능 분석 세션을 마쳤으면 호스트 PC에서 프로바이더를 제거할 수 있습니다:

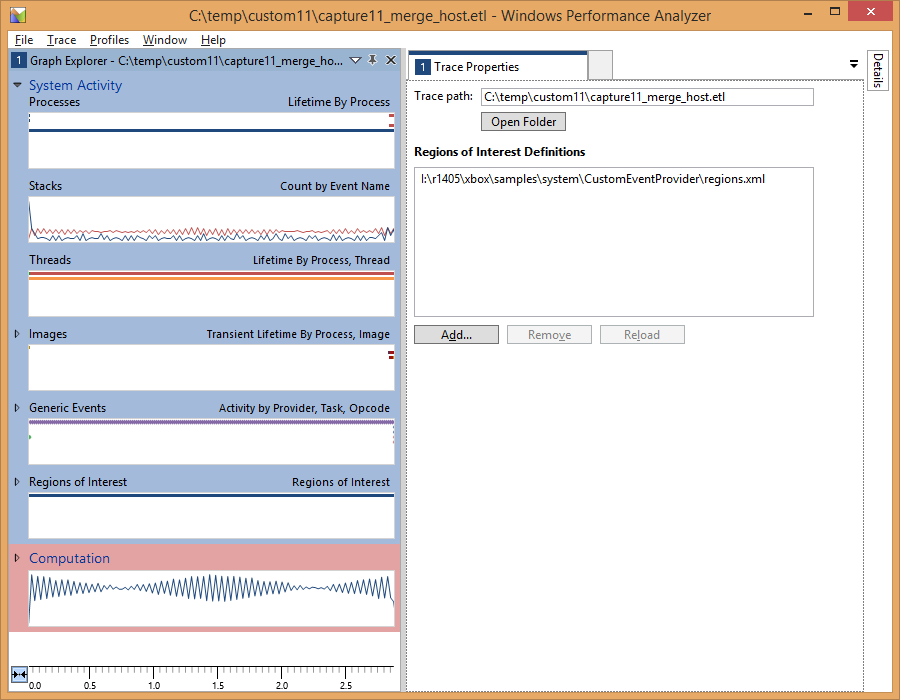
D:\dev\CustomEventProvider>wevtutil um etwprovider.man

BlockCulled 이벤트는 Mark 이벤트와 비슷하며 문자열 페이로드가 아니라 단일 UInt32 페이로드를 포함한다는 점은 제외입니다. 안타깝게도 현재 WPA에서 사용자 지정 이벤트의 숫자 필드를 그래프로 표시 할수 없습니다.

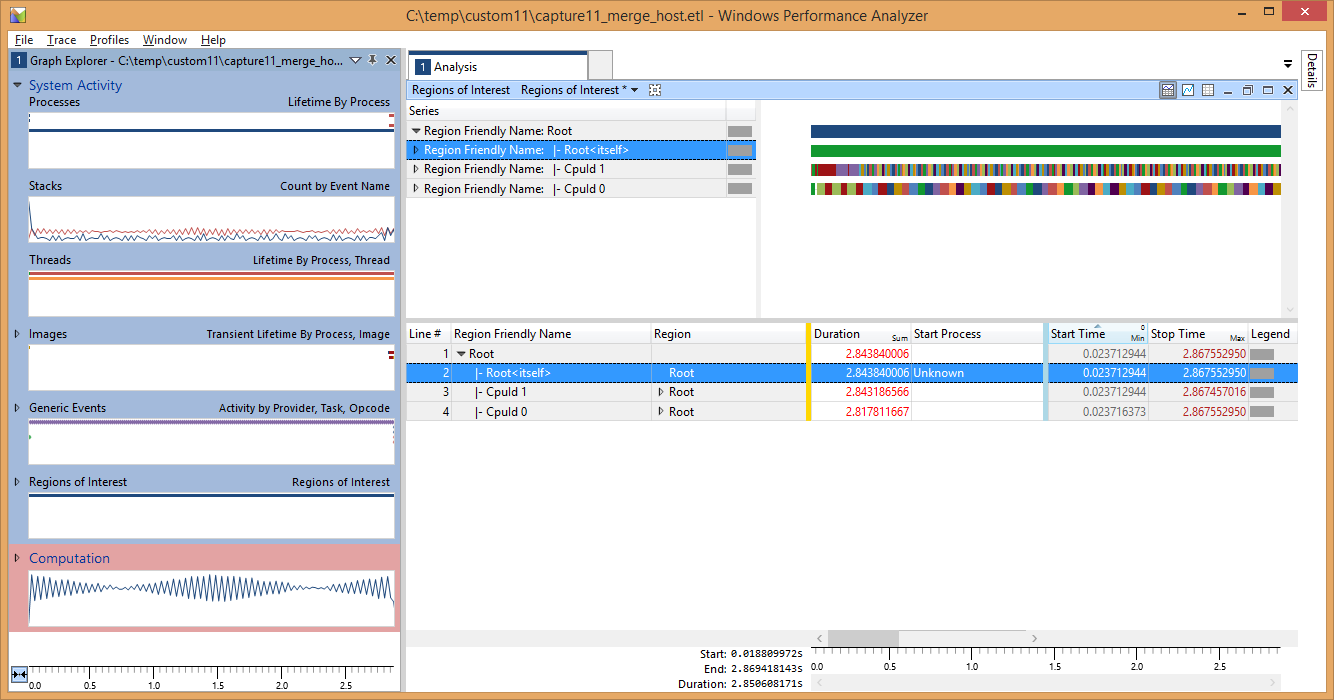
**관심 영역**

2013 년 10 월부터 WPA는 [관심 영역](http://msdn.microsoft.com/ko-kr/library/windows/hardware/dn450838.aspx)이라는 개녕을 제공하였는데: 캡처 내에서 시간 범위를 표시하고 레이블을 지정할수 있는 기능입니다. 그만큼 **EtwScopedEvent** 클래스와 **ETWScopedEvent()** 매크로는 적절한 페이로드로 관심 영역(ROI)을 사용하여 **PIXBeginEvent()** 및 **PIXEndEvent()**와 유사한 브라케팅 기능을 제공하는 방법을 보여줍니다.

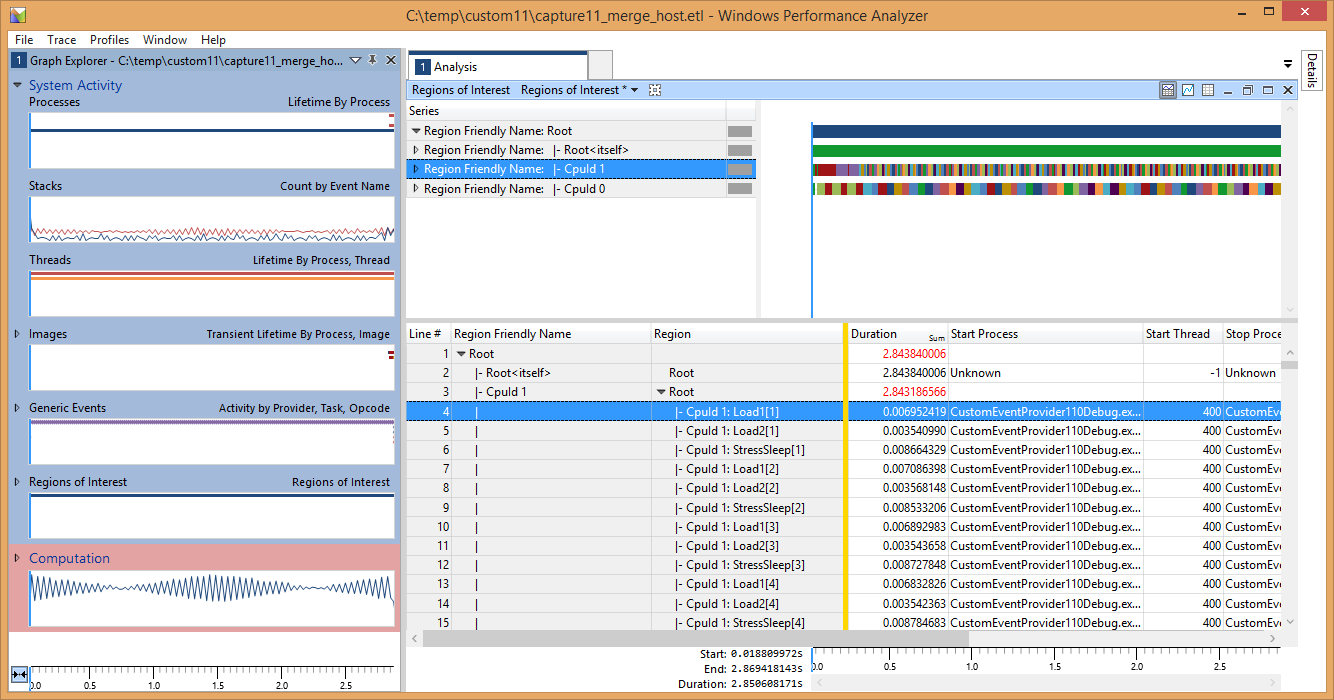
ROI를 표시하려면 먼저 영역 정의 파일을 로드해야 합니다. Trace 메뉴에서 Trace Properties를 선택한 다음 샘플과 함께 제공되는 regions.xml 정의를 로드합니다.



이제 Generic Events 그래프 아래에서 사용 가능한 관심 영역 그래프를 볼수 있습니다. 분석 영역으로 ROI 그래프를 드래그하여 확장하세요; 기본 뷰 프리셋 (툴바에서)은 "관심 영역"이어야 합니다. 테이블에 Region 열을 추가하여 (각 영역이 고유 한 색을 갖도록) Root 노드를 확장합니다; 다음과 유사한 화면이 나타납니다:

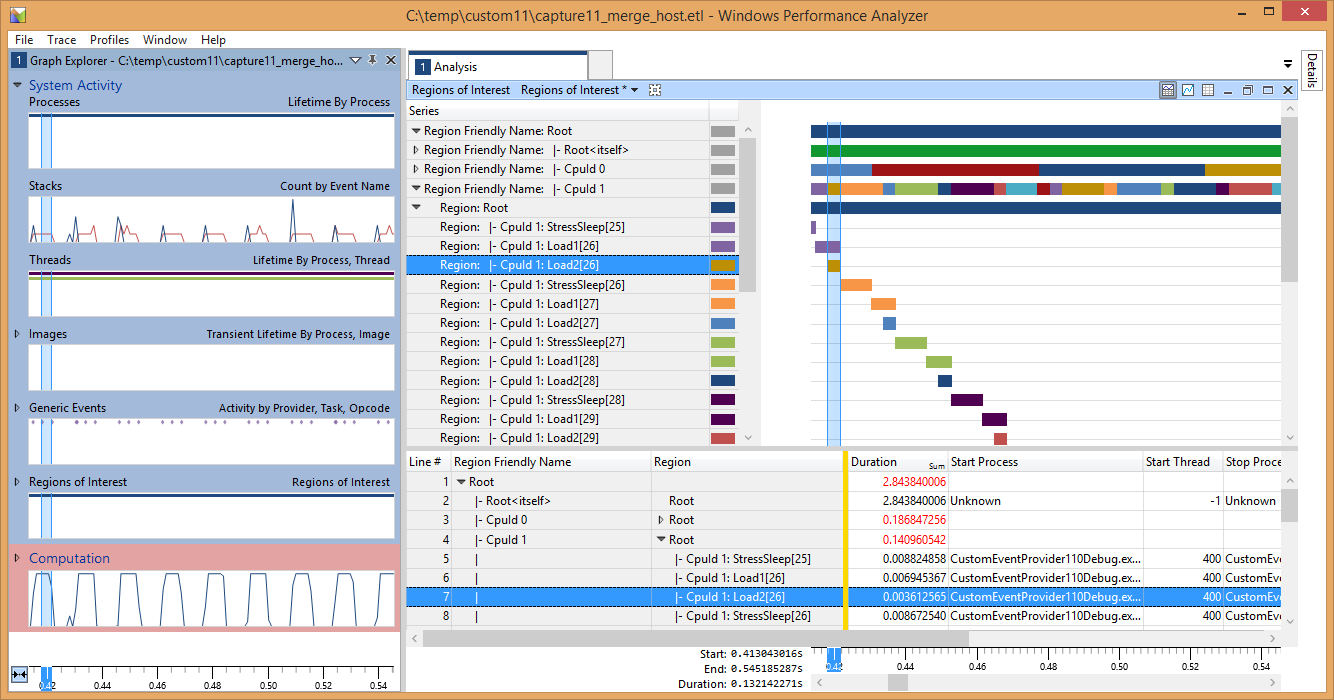


테이블의 Region 노드를 확장하면 개별 대괄호에 대한 정보가 제공됩니다:

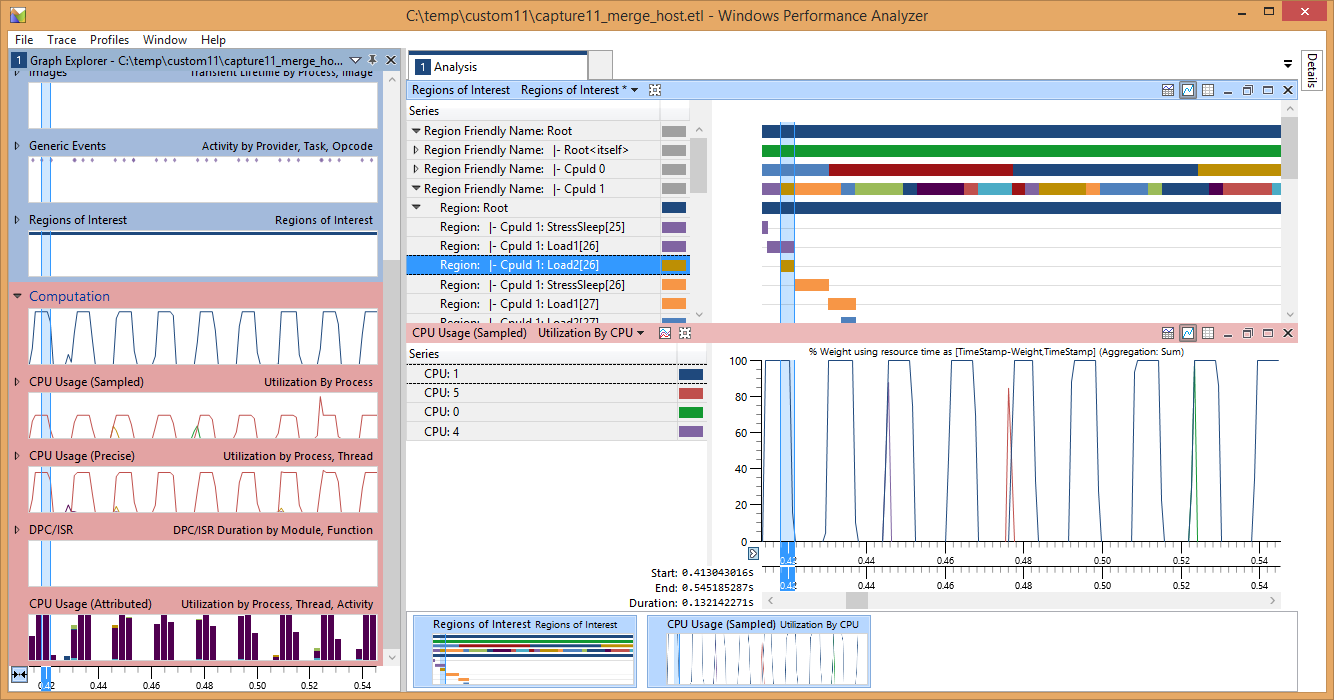


보시다시피, **ETWScopedEvent ()** 호출에 제공된 라벨은 여기에 표시됩니다 (숫자는 특정 라벨의 인스턴스입니다).

그래프를 확장하면 각 영역의 타임 라인이 별도로 표시됩니다:



더 나아가, 이것을 활성화했다면, 영역을 샘플링 캡처의 데이터와 상관시킬 수 있습니다:



# 알려진 문제점

호스트 PC에서 이벤트 프로바이더 GUID를 해결하므로 해당 ID를 가진 ETW 프로바이더가 이미 등록되어 있지 말아야 합니다. 샘플에서 새 이벤트 매니페스트를 만드는 경우 매니페스트를 편집할 때 ecmangen을 사용하여 GUID를 다시 생성하거나 guidgen.exe (Visual Studio에서 제공)를 사용하여 새 GUID를 생성합니다.

호스트 PC에서 ETL 파일이 해결되면 Generic Events보기에 몇 가지 추가 이벤트 프로바이더가 나타날 수 있습니다; 이들은 무시할 수 있습니다.

# 개인정보처리방침

샘플을 컴파일하고 실행할 때 샘플 실행 파일의 이름이 Microsoft로 보내져 샘플 사용을 추적 할 수 있습니다. 이 데이터 수집을 거부하려면 Main.cpp에서 "샘플 사용 텔레메트리"라는 코드 블록을 제거하면 됩니다.

Microsoft의 개인 정보 취급 방침에 대한 일반적인 내용은 [Microsoft 개인 정보 취급 방침](https://privacy.microsoft.com/ko-kr/privacystatement/)을 참조하십시오.