

カスタム イベント プロバイダーのサンプル

*このサンプルは、Microsoft ゲーム開発キットのプレビュー (2019 年 11 月) に対応しています。*

# 説明 このサンプルでは、Xbox One でカスタム ETW イベント プロバイダーを使用する方法について示します。 サンプルの作成

Project Scarlett を使用している場合は、Gaming.Xbox.Scarlett.x64 プラットフォーム構成をプロジェクトに追加する必要があります。この操作は、*構成マネージャー*を使って行うことができます。[アクティブなソリューション プラットフォーム] で [構成マネージャー] オプションを選択し、次に [新規作成...] を選択します。[新しいプラットフォームの入力または選択] を Gaming.Xbox.Scarlett.x64 に、[設定のコピー元] をGaming.Xbox.XboxOne.x64 に設定します。次に [OK] を選択します。

*詳細については、*GDK ドキュメントの 「サンプルの実行」を参照してください。

# サンプルの使用

この例では、次のコントロールを使用します。

|  |  |
| --- | --- |
| 動作​​ | ゲームパッド |
| サンプルを終了する。 | 左トリガー + 右トリガー + 右ショルダー |

実装に関する注意事項  
このサンプルは、従来の Windows ETW プロバイダーと同じ構造です。ただし、排他的パーティションで実行されているタイトルは、そのイベント プロバイダーをレジストリに追加できないため、生成されたイベント データを正しく解決するには、ホスト PC でいくつかの追加手順が必要です。

イベントを生成するには、最初にイベント マニフェスト ファイルを作成する必要があります (このサンプルでは、**etwprovider.man** にあります)。

イベント マニフェスト ファイルは、手動で作成することができます。これは XML ベースか、または **ecmangen.exe** (Windows SDK の一部として出荷された GUI ベースのツール) で構築できます。通常、Manifest ジェネレーター ツールは、c:\Program Files (x86)\Windows Kits\10\bin\{sdk version}\x64\ecmangen.exe にあります。

イベント マニフェストを取得したら、Visual Studio メッセージ コンパイラー (mc.exe) を使用して、リソース ファイル (**etwproviderGenerated.rc**) とヘッダー (**etwproviderGenerated.h**) にコンパイルされます。リソース ファイルとヘッダーはいずれもタイトル プロジェクトに含まれています。

ヘッダー、リソース、バイナリ ファイルを生成するには、次のパラメーターを使用して mc.exe を呼び出します。

mc.exe -um *inputmanifestfile.man*

このステップで生成された .BIN ファイルは、生成された .RC ファイルにより自動的に読み込まれ、実行可能ファイルまたは DLL にコンパイルされます。

タイトルの初期化中に、イベント プロバイダーは **EventRegisterCEP\_Main** の呼び出しで登録され、その後、シャットダウン処理中に **EventUnregisterCEP\_Main** の呼び出で登録が解除されます。「マーク」イベントは、1 つの Unicode 文字列をパラメーターとして理解する **EventWriteMark** の呼び出しにより生成されます。

このサンプルは、他のサンプルと同様に、構築、展開、アクティブ化が可能です。実行すると、xperf を使用してイベントをキャプチャします (tracelog には、ETL プロバイダー イベントを結合する機能がなく、xbperf にはカスタム イベント プロバイダーを指定する機能はありません)。タイトルのイベント プロバイダーはレジストリに追加されていないため、名前ではなく GUID で識別する必要があります (GUID は、イベント マニフェストで指定されたプロバイダー GUID と一致する必要があります)。

C:\temp>xbrun /x/title /O xperf -start -on PROC\_THREAD+LOADER+DPC+INTERRUPT+CSWITCH+PROFILE -stackwalk PROFILE+CSWITCH -f d:\kernel.etl

必要なデータがキャプチャされると、通常の方法でセッションを停止できます。

C:\temp> xbrun /x/title /O xperf -start "user" -on A4A76336-4BA7-4CD9-85C3-B9C236D3041C -f d:\user.etl

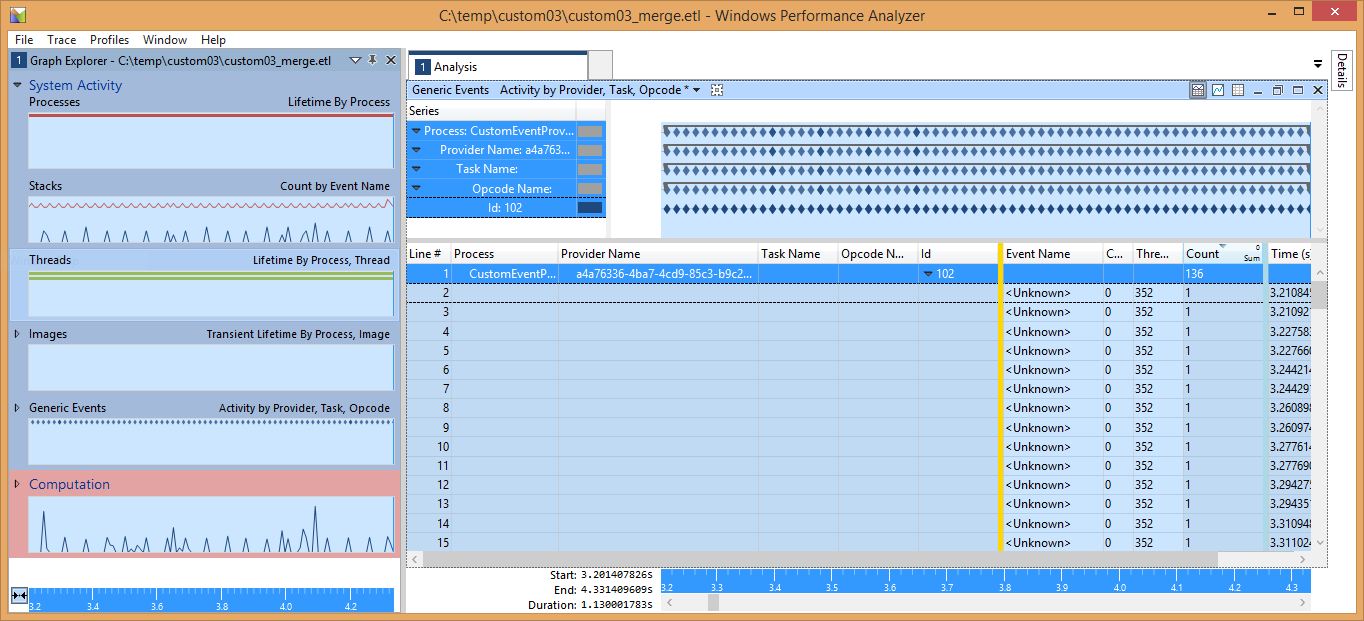
devkit の ETL ファイルを結合してシステム イベント プロバイダーを解決します。カスタム イベント プロバイダーは解決*しません*。

C:\temp> xbrun /x/title /O xperf -stop -stop "user" -d d:\merged.etl

結合されたファイルをホスト PC に戻せるようになりました。

C:\temp\> xbcp xd:\merged.etl .

このファイルは、WPA に読み込むことができ、カスタム イベントは、[システム アクティビティ] グループ内の [Generic Events] グラフに表示されます。ただし、この時点では、イベントは GUID のみで識別され、タスク名やオペコード名などの情報は表示されません。さらに重要なのは、各イベントに対して指定したカスタム データ (Unicode 文字列) は表示されないという点です。



各カスタム イベントの全情報を表示するには、あるトリックを使います。*イベント プロバイダーを devkit ではなくホスト PC に登録し、そこでイベントを解決します。*

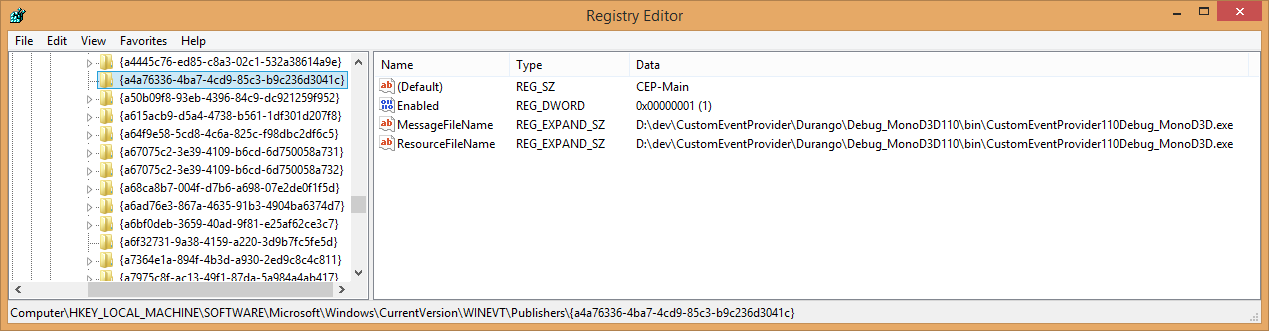
最初に、イベント マニフェストのプロバイダー ノード (evtprovider.man) を編集して、**resourceFileName** と **messageFileName** 属性が、Xbox One 実行可能ファイルが構築された開発用 PC の場所を指定するようにします。

<provider name="CEP-Main" guid="{A4A76336-4BA7-4CD9-85C3-B9C236D3041C}"   
symbol="CEP\_MAIN"   
resourceFileName="S:\samples\gx\_dev\Samples\System\CustomEventProvider\Gaming.Xbox.x64\Debug\CustomEventProvider.exe"   
messageFileName="S:\samples\gx\_dev\Samples\System\CustomEventProvider\Gaming.Xbox.x64\Debug\CustomEventProvider.exe">

次に、管理者特権でのコマンド プロンプトから wevtutil.exe ツールを実行して、ホスト PC にイベント プロバイダーを登録します。

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

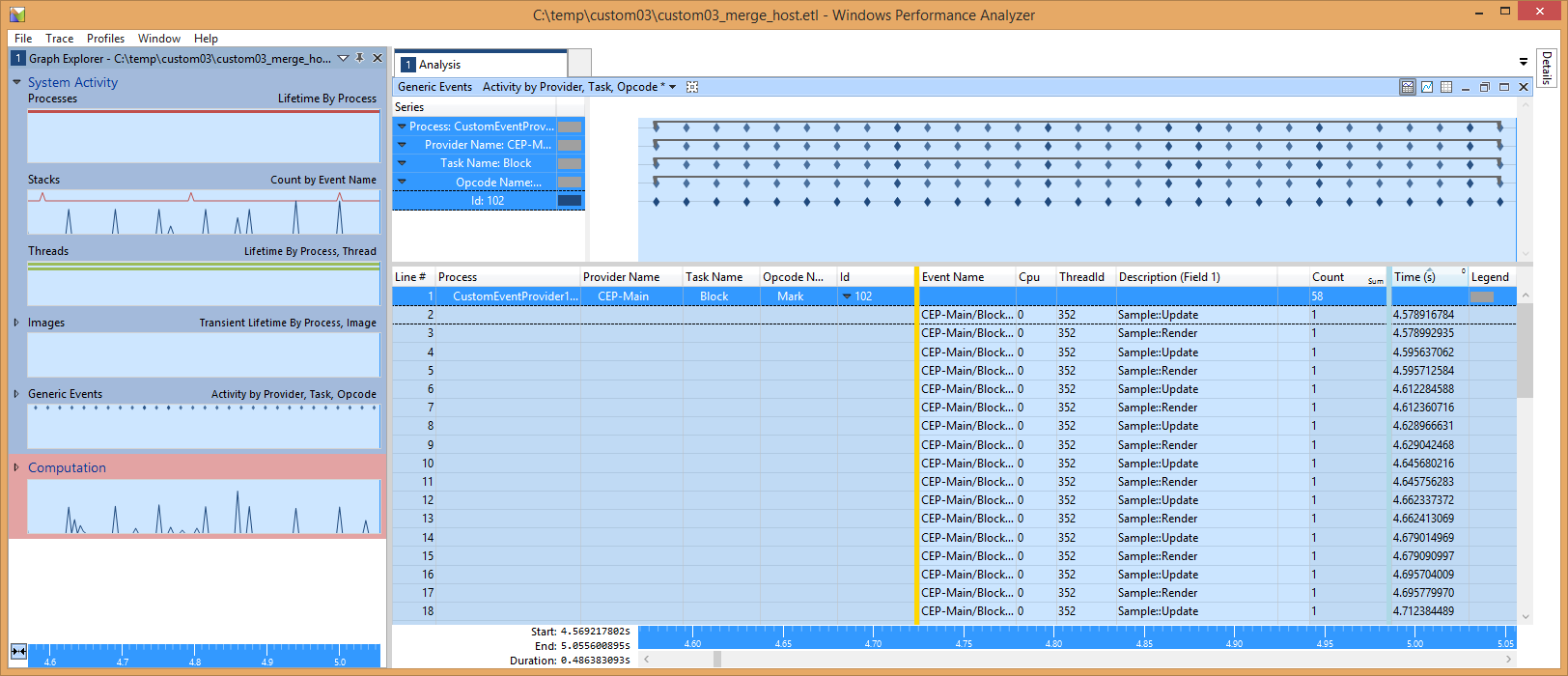
ホスト PC のレジストリをチェックするには、 HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Publishers にリスト表示されたプロバイダーを確認します。



最後に、xperf を使用してホスト PC で ETL ファイルを解決します。

C:\temp\custom03>xperf -merge merged.etl final.etl

ホストの結合 ETL ファイル (**final.etl**) が WPA に読み込まれると、適切に解決されたイベントが次のように表示されます。



[説明] (フィールド 1) 列に、これらのイベントで記録された文字列が表示されます。[タスク] 名と [オペコード] 名も確認できます。

パフォーマンス分析セッションが完了したら、ホスト PC からプロバイダーを削除できます。

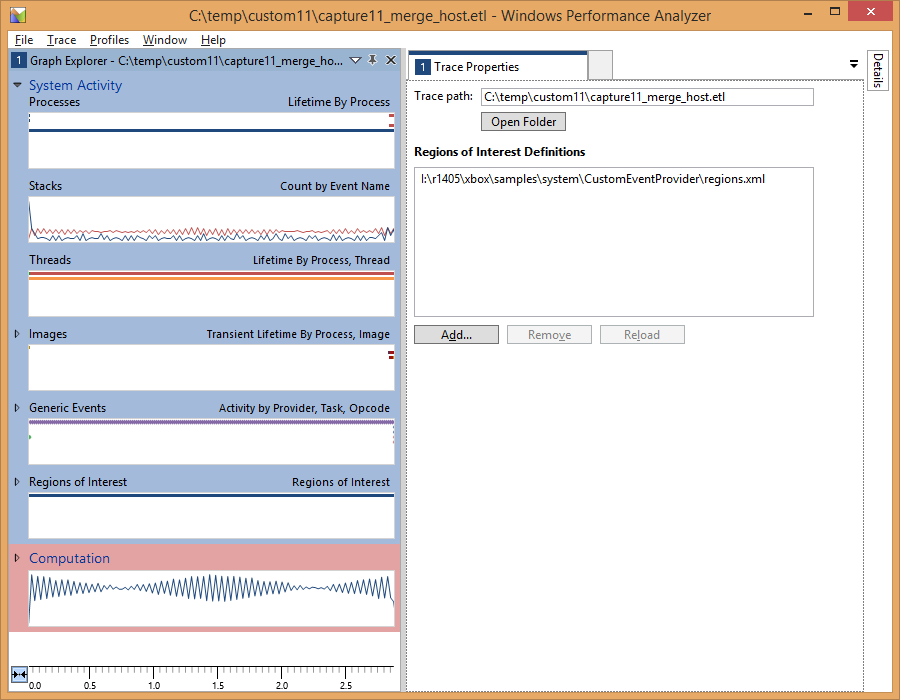
D:\dev\CustomEventProvider>wevtutil um etwprovider.man

BlockCulled イベントはマーク イベントに似ていますが、文字列ペイロードではなく UInt32 ペイロードが 1 つあることに留意してください。残念ながら、現在のところ、WPA のカスタム イベントの数値フィールドをグラフ化することはできません。

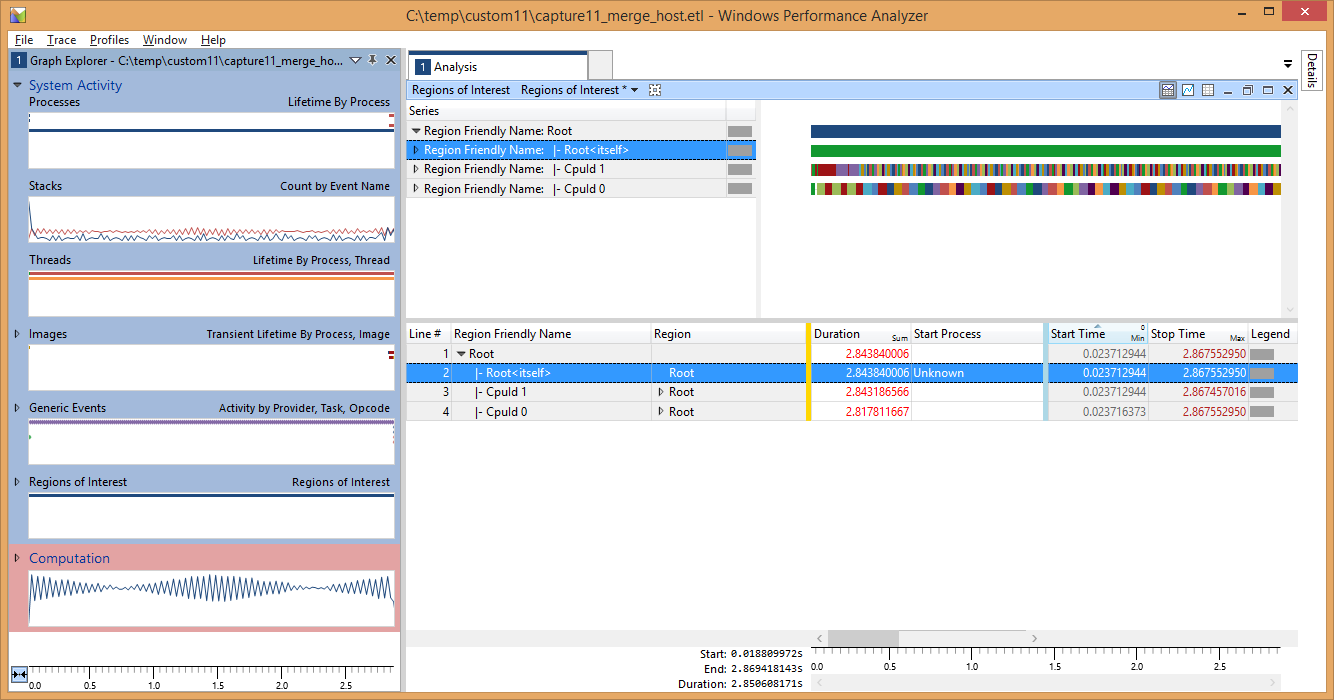
**関心領域**

2013 年 10 月以降、WPA は[関心領域](http://msdn.microsoft.com/en-us/library/windows/hardware/dn450838.aspx)の概念、つまりキャプチャ内の時間範囲を示し、ラベル付けする機能に対応しています。**EtwScopedEvent** クラスと **ETWScopedEvent()** マクロは、適切なペイロードにより、関心領域 (ROI) を使用して **PIXBeginEvent()** と **PIXEndEvent()** に似たブラケット機能を提供できます。

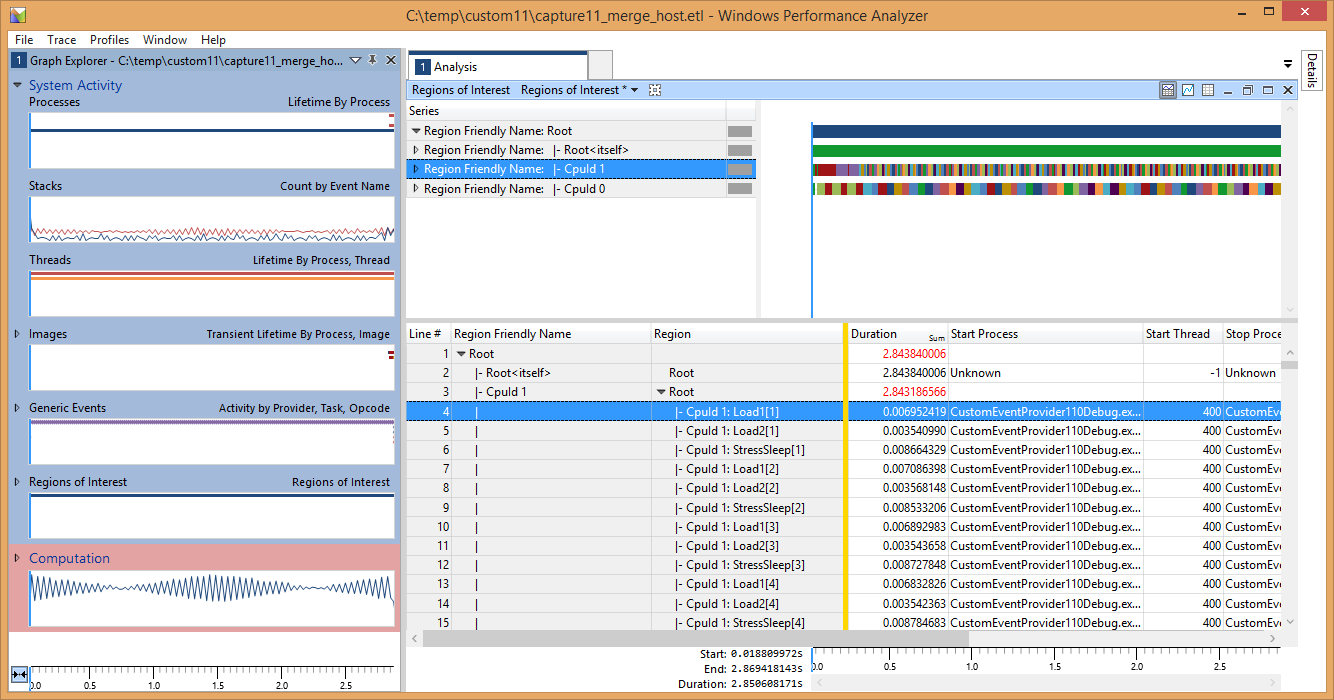
ROI を表示するには、まず領域定義ファイルを読み込む必要があります。[トレース] メニューから [トレースのプロパティ] を選択して、サンプルに付属の xml 定義を読み込みます。



[Generic Events] グラフの下に、利用可能な関心領域のグラフが表示されます。ROI グラフを分析領域にドラッグして展開します。事前設定された (ツールバー上の) 既定の表示は、[関心領域] です。テーブルに [領域] 列を追加し (各領域がそれぞれの色で表示されるようになります)、ルート ノードを展開します。次のような画面が表示されます。

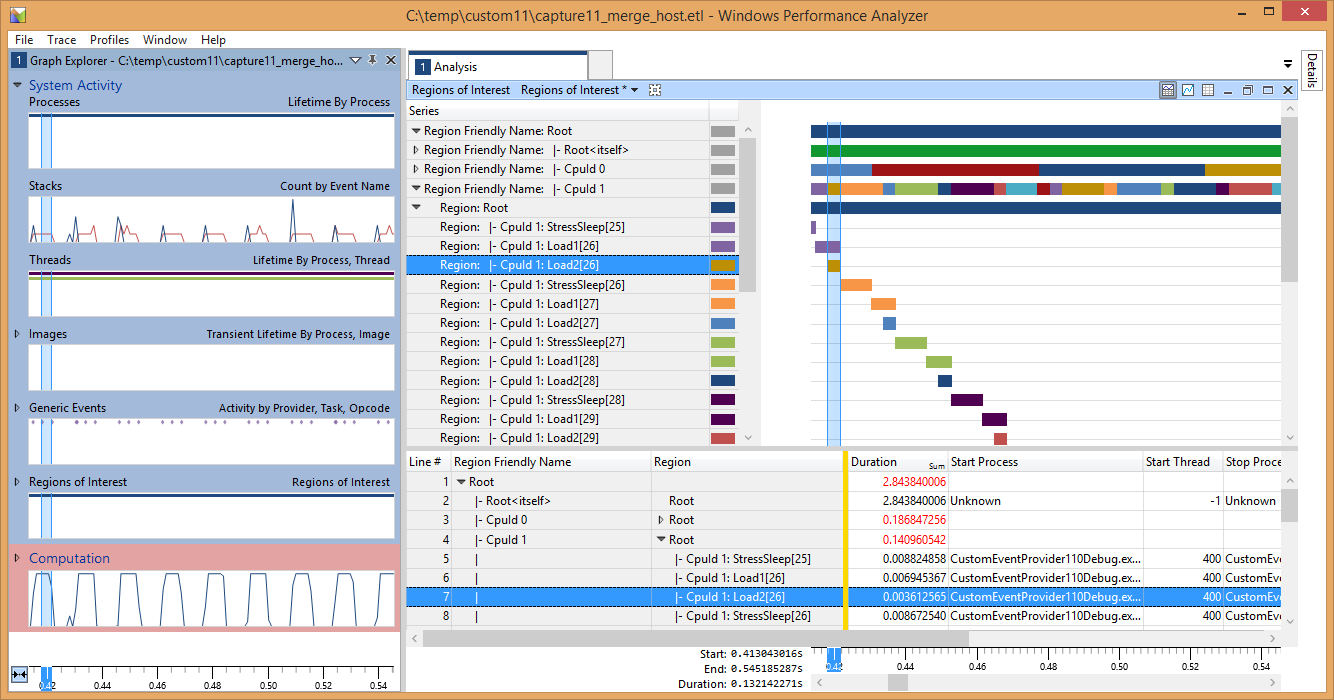


テーブルの領域ノードを展開すると、個々のかっこの情報が表示されます。

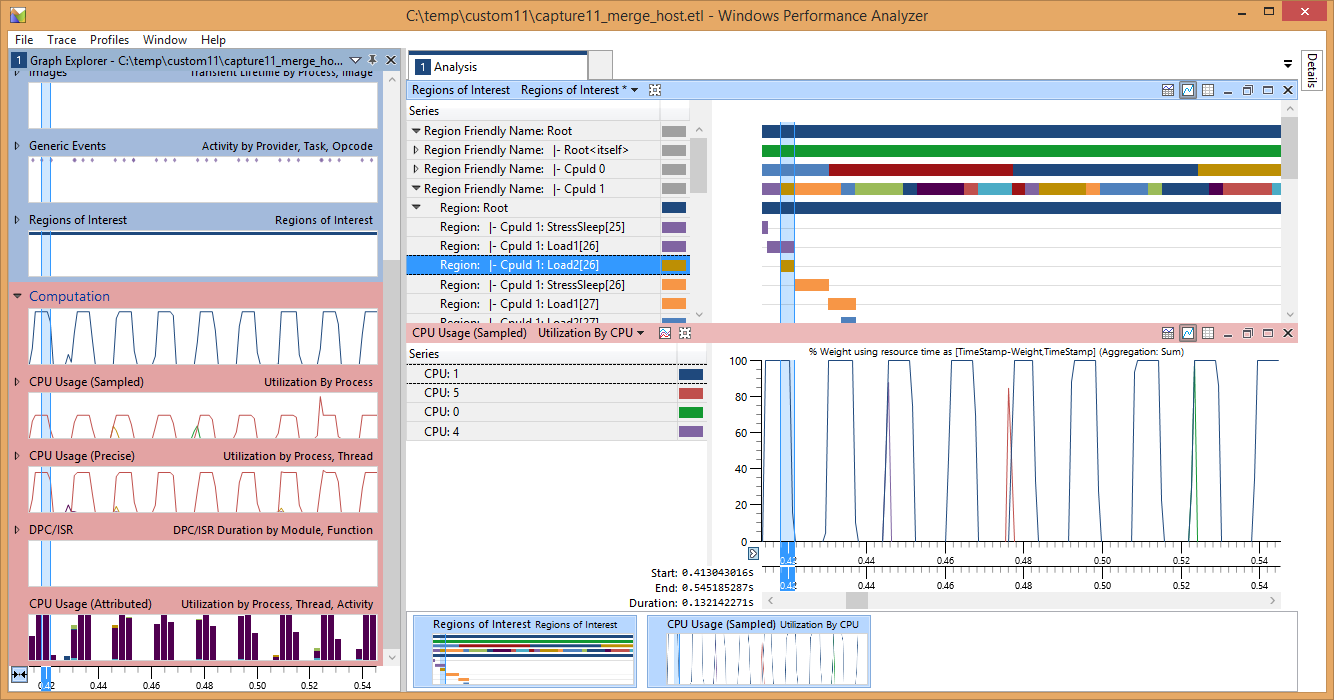


ご覧のとおり、**ETWScopedEvent ()** への呼び出しで指定されたラベルがここに表示されています (数値は特定のラベルのインスタンスです)。

グラフを展開すると、各領域のタイムラインが個別に表示されます。



さらに、この機能が有効なとき、領域をサンプリング キャプチャのデータと関連付けることができます。



# 既知の問題

ホスト PC でイベント プロバイダーの GUID を解決するため、その ID を持つ ETW プロバイダーが登録されていないことが必須です。このサンプルの 1 つから新しいイベント マニフェストを作成するには、マニフェストの編集中に **ecmangen** を使って GUID を再生成するか、または guidgen.exe (Visual Studio に付属) を使用して新しい GUID を生成します。

ホスト PC で ETL ファイルが解決されると、[Generic Events] ビューに追加のイベント プロバイダーが表示される場合がありますが、これらは無視できます。

# その他の参照情報

## イベント マニフェストの作成

* [Windows イベント ログの使用](https://docs.microsoft.com/en-us/windows/desktop/WES/using-windows-event-log)
* [インストルメンテーション マニフェストの記述](https://docs.microsoft.com/en-us/windows/desktop/WES/writing-an-instrumentation-manifest)
* [インストルメンテーション マニフェストのコンパイル](https://docs.microsoft.com/en-us/windows/desktop/WES/compiling-an-instrumentation-manifest)
* [メッセージ コンパイラー (mc.exe)](https://docs.microsoft.com/en-us/windows/desktop/WES/message-compiler--mc-exe-)

## Windows パフォーマンス アナライザー

* [Windows パフォーマンス アナライザー](https://docs.microsoft.com/en-us/windows-hardware/test/wpt/windows-performance-analyzer)
* [関心領域](https://docs.microsoft.com/en-us/windows-hardware/test/wpt/regions-of-interest)
* [関心領域ファイルを作成する](https://docs.microsoft.com/en-us/windows-hardware/test/wpt/creating-a-regions-of-interest-file)

# プライバシーに関する声明

サンプルをコンパイルして実行すると、サンプルの使用状況を追跡するため、サンプル実行可能ファイルのファイル名が Microsoft に送信されます。このデータ収集を無効にするには、「Sample Usage Telemetry」とラベル付けされた Main.cpp 内のコードのブロックを削除します。

Microsoft のプライバシー ステートメントの詳細については、「[Microsoft のプライバシー ステートメント](https://privacy.microsoft.com/en-us/privacystatement/)」を参照してください。