

# トランザクション内のイベント

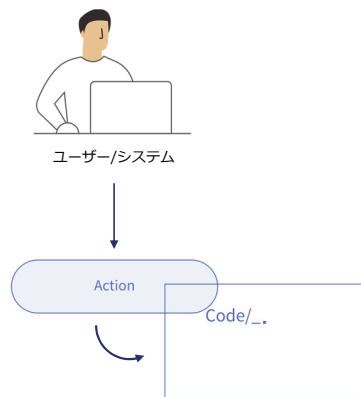


引き続き Transaction オブジェクトを見ていきます。

ここまでに、GeneXus ではトランザクションで定義されている構造を使用して自動的にフォームが作成されること、その動作はルールを宣言することで定義できることを確認しました。

ここでは、トランザクション内のイベントについて説明します。

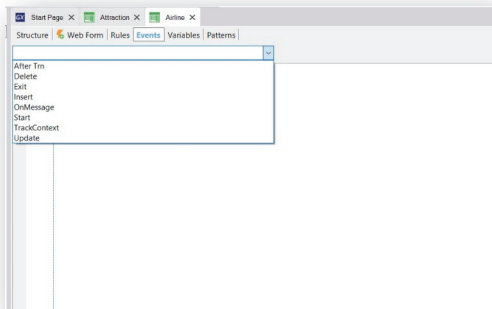
## イベント: 概念



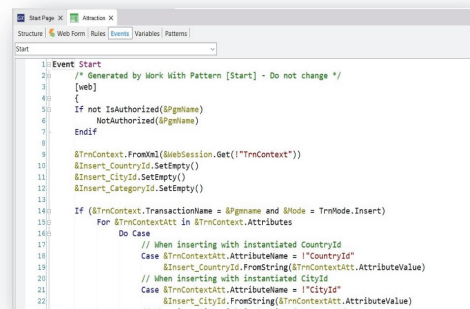
イベントとは、ユーザーまたはシステムが実行するアクションであり、そのアクションに応じて特定のコードがアクティベートされます。

## トランザクション内のイベント

一般的なトランザクション



Work With for Web パターンのトランザクション



最初に Airline トランザクションを見てみます。[Events] エLEMENTは空白で、必要なイベントを編集したり、スケジュールを設定したりできるようになっています。

右側は宣言されたコードです。詳しい説明は省きますが、Work With for Web パターンが適用されたときに自動生成されたものであることを覚えておいてください。

## 利用可能なイベント

**Start:** トランザクションを開き、操作を開始するときに実行するアクションを定義します。

**TrackContext:** トランザクション実行のコンテキストに変更があった場合に実行するアクションを設定できます。

**OnMessage:** Web 通知に関連し、リアルタイムでアクションを実行できます。

**Insert, Update, Delete:** ダイナミックトランザクション (このコースの対象外) の更新の概念に関連します。

**Exit:** トランザクションの実行が完了したとき、つまりトランザクションが閉じた後で実行するアクションを指定できます。

**After Trn:** トランザクションの各実行サイクルの後、つまりコミットの直後に実行するアクションを指定できます。

**Start** イベントから見ていきます。このイベントはシステムイベントであり、一般にトランザクションの実行中に使用する変数に値を割り当てるために使用します。

次は **TrackContext** イベントです。このイベントでは、トランザクション実行のコンテキストに変更があった場合に実行するアクションを設定できます。たとえば、このイベントで特定のコントロール内でのカーソルの位置を追跡し、プログラミングされているコードを実行することができます。

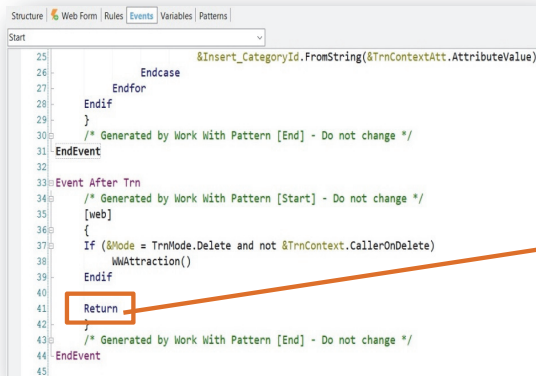
**OnMessage** イベントは Web 通知に関連し、リアルタイムでアクションを実行できます。

**Insert, Update, Delete** の各イベントは、ダイナミックトランザクション (このコースの対象外) の更新の概念に関連します。これらのイベントは、トランザクションを使用する特殊なケースが対象となります。

**Exit** イベントでは、トランザクションの実行が完了したとき、つまりトランザクションが閉じた後で実行するアクションを指定できます。

最後に **After Trn** イベントです。このイベントでは、トランザクションの各実行サイクルの後、つまりコミットの直後に実行するアクションを指定できます。先に説明した、ルールをトリガーするタイミングで言えば、この **After Trn** イベントの実行は、トリガーのタイミング **on AfterComplete** と同じになります。

## After Trn イベント



```
25      &Insert_CategoryId.FromString(&TrnContextAtt.AttributeValue)
26      Endcase
27      Endfor
28      Endif
29      }
30      /* Generated by Work With Pattern [End] - Do not change */
31  EndEvent
32
33  Event After Trn
34      /* Generated by Work With Pattern [Start] - Do not change */
35      [web]
36      {
37          If (&Mode = TrnMode.Delete and not &TrnContext.CallerOnDelete)
38              WvAttraction()
39          Endif
40
41      Return
42
43      /* Generated by Work With Pattern [End] - Do not change */
44  EndEvent
45
```

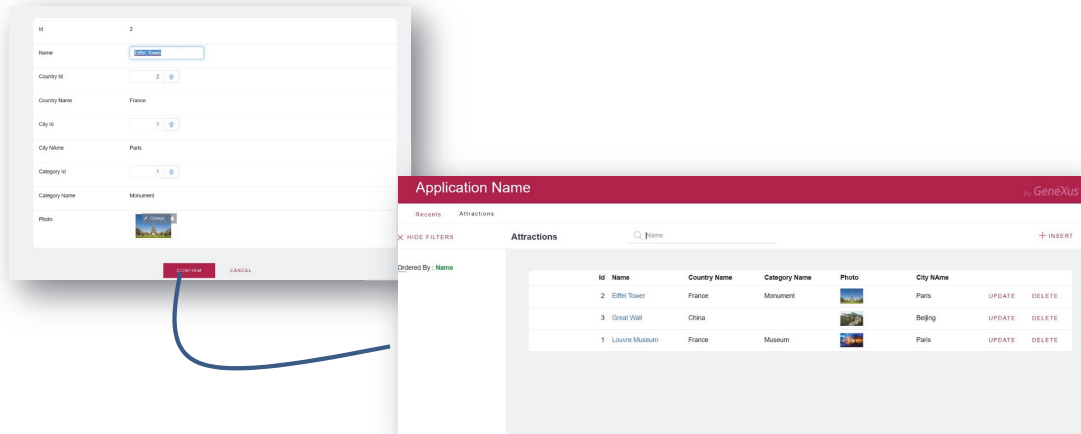
実行時に観光名所を挿入、削除、変更すると、自動的に Work With Attractions のメイン画面に戻る。

Attraction トランザクションに戻り、After Trn イベントで宣言されているコードを確認します。

After Trn イベントのコードは、Work With for Web パターンの適用時に追加されました。具体的には Return コマンドが追加されています。

このため、実行時に観光名所を挿入、削除、変更すると、自動的に Work With Attractions のメイン画面に戻ります。トランザクションフォームの作業サイクルごとにコミットが実行されるため、その都度 After Trn イベントがトリガーされ、この Return コマンドが実行されます。

## 実行時の After Trn イベント



名前を変更するとします。[実行] ボタンをクリックすると、トランザクションフォームの作業サイクルが終了し、コミット、After Trn イベントの順にトリガーされます。

このイベントで宣言されている Return コマンドが実行され、Work With Attractions の最初の画面に戻ります。

on AfterComplete のタイミングで条件を設定して Return を開始しないのはなぜでしょうか。これはコマンドであり、ルールではないからです。このため、イベント内で宣言する必要があり、コミット後にトリガーされるイベントは After Trn イベントです。

最後に注意事項です。何らかの機能を実装するためにイベントにコードを追加する必要があり、イベントには GeneXus によって自動生成されたコードが既に存在する場合、独自のコードは、GeneXus によって自動的に管理されるコードのマークの外で宣言する必要があります。