

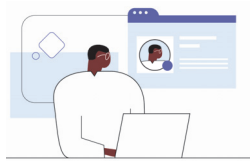
# 顧客向けに重点を置いた Web 画面

イベント実行スキーム、クライアント側イベントの文法、  
およびベーステーブルの決定

**GeneXus™**

この章では、まず、顧客向けアプリケーションの画面、特に Web 画面で定義できるイベントについて説明します。また、Panel オブジェクトのいくつかの特性についても説明します。

## クライアントとサーバーのイベント



UX にフォーカスした  
Web クライアント

- ClientStart
- Back
- ユーザーイベント
- コントロールイベント



サーバー

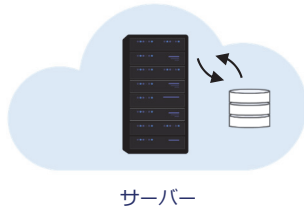
- Start
- Refresh
- Load

このタイプのアプリケーションには、クライアント側でトリガーされるイベントとサーバー側でトリガーされるイベントの 2 つのタイプのイベントがあります。

サーバー側イベントは、Web パネルを学習したときに見たものと同じで、データベースとの対話に使用できる Start、Refresh、および Load イベントが該当します。クライアント側イベントは、ClientStart、Back、ユーザー定義イベント、および画面コントロールに関連付けられたイベントが該当します。

パネルオブジェクトを使用してモバイルデバイス用のアプリケーションを生成する場合は、モバイルプラットフォーム関連のほかのイベントも扱うことになります。たとえば、デバイスのタイプや、アプリケーション実行時の方向に関連するものです。

## サーバー側イベント



サーバー

- Start
- Refresh
- Load

- Start イベントは 1 回だけ実行されます。
- Refresh イベントは Start イベントの後に実行されます。
- Load イベントは Refresh イベントによってトリガーされます。
- Load イベントは、最後に実行されるシステムイベントです (グリッドが存在する場合のみ)。
  - ベーステーブルを持つグリッド: Load はベーステーブル内のレコードの数だけ実行されます。
  - ベーステーブルを持たないグリッド: Load は 1 回だけ実行されます。
  - SDT ベースのグリッド: Load はトリガーされません。

次に、サーバー側イベントについて説明します：

- 最初に実行されるのは Start イベントです。パネルを開いたときに 1 回だけ実行され、オブジェクトを一度離れて開きなおさない限り、再度実行されることはありません。
- Refresh イベントは、Start イベントの後に、通常は 1 回だけ実行されますが、Refresh コマンドを使用して再度呼び出すことができます。その場合、Start が再度実行されることはなく、このイベントが最初のイベントになり、複数回実行されることもあります。
- Refresh イベントが呼び出されると、最後に Load イベントがトリガーされます。このイベントは、パネルに少なくとも 1 つのグリッドが存在し、そのグリッドがコレクション SDT 変数でない場合にのみ発生します。
- Load イベントは、最後に実行されるシステムイベントです：
  - ベーステーブルが存在する場合は、そのベーステーブル内のレコードの数だけ実行されます。
  - グリッドにベーステーブルが存在しない場合は、1 回だけ実行されます。
  - また、グリッドが SDT に基づいている場合、Load イベントは実行されません。

## サーバー側イベントの例

```

1  Event Load
2      &Trips = Count(TripDate)
3  Endevent
4
5  Event Refresh
6      &TotalTrips = 0
7      For each Trip.Attraction
8          &TotalTrips += 1
9      Endfor
10 Endevent
11
12 Event &CountryId.ControlValueChanged
13     Grid1.Refresh()
14 Endevent
15
16 Event &AttractionNameFrom.ControlValueChanged
17     Grid1.Refresh()
18 Endevent
19
20
21 Event &AttractionNameTo.ControlValueChanged
22     Grid1.Refresh()
23 Endevent
24
25 Event Start
26     &Details = "Details"
27 Endevent
28
29 Event &Details.Tap
30     AttractionDetail_CFPANEL.Call(AttractionId)
31 Endevent

```

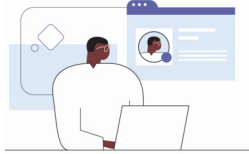
前の章の例では、Attractions\_CFPANEL パネルオブジェクトを実装したときに、イベント Start、Refresh、および Load をプログラムしました。

Load イベントでは、TripAttraction テーブルにアクセスする式を使用して、特定の観光名所が含まれるツアーの合計を計算しました。

Refresh イベントでは、TripAttraction テーブルにアクセスして、各観光名所が含まれるツアーの数を全体で合計するように For Each コマンドをプログラムしました。

また、Start イベントで、画面に表示したいテキストで &Details 変数を初期化するようにプログラムしました。

## クライアント側イベント



顧客に重点を置いた  
Web クライアント

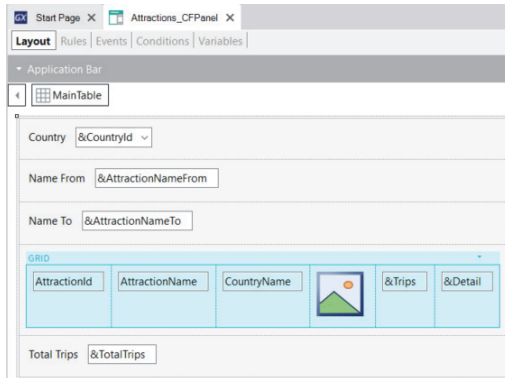
- ClientStart
- Back
- ユーザーイベント
- コントロールイベント

- タイプ:
  - システムイベント: ClientStart、Back
  - ユーザーイベント: ユーザー定義
  - コントロールイベント: コントロールごとに事前定義済み
- サービスを呼び出してサーバーリソースにアクセス
- サーバー側イベントのトリガーなし (Refresh コマンドを使用した場合を除く)
- 異なる文法を使用

クライアント側イベントは、ユーザー操作に対するアプリケーションの応答です。

- クライアント側イベントには、システムイベント、ユーザーイベント、およびコントロールイベントの 3 つのタイプがあります。システムイベントは、ClientStart と Back です (このコースでは取り上げません)。ユーザーイベントは、ユーザーによって作成されるイベントであり、コントロールイベントは、Tap、Double Tap、ControlValueChanged イベントなど、画面コントロールに関連するイベントです。
- これらのイベントに関連付けられたコードは、データベースへのアクセスが必要な場合など、サーバーサービスへのアクセスが必要でない限り、クライアントで実行されます。
- クライアント側イベントの実行中は、Refresh コマンドで明示的に要求しない限り、サーバー側イベントは実行されません。
- これらのクライアント側イベントには、サーバー側イベントとは異なる特定の文法があります。

## クライアント側イベントの例



```

1  Event Load
2      &Trips = Count(TripDate)
3  Endevent
4
5  Event Refresh
6      &TotalTrips = 0
7      For each Trip.Attraction
8          &TotalTrips += 1
9      Endfor
10 Endevent
11
12 Event &CountryId.ControlValueChanged
13     Grid1.Refresh()
14 Endevent
15
16 Event &AttractionNameFrom.ControlValueChanged
17     Grid1.Refresh()
18 Endevent
19
20
21 Event &AttractionNameTo.ControlValueChanged
22     Grid1.Refresh()
23 Endevent
24
25 Event Start
26     &Details = "Details"
27 Endevent
28
29 Event &Details.Tap
30     AttractionDetail_CFPANEL.Call(AttractionId)
31 Endevent

```

前に取り上げたものと同じオブジェクトで、コントロールに関連するイベントのみをプログラムします。ユーザーイベントはプログラムしません。画面上のフィルタの各変数に対して `ControlValueChanged` イベントをプログラムし、グリッドの `Refresh` メソッドをトリガーします。これにより、サーバーの `Refresh` イベントと `Load` イベントがトリガーされ、指定されたフィルタでグリッドのコンテンツが更新されます。

また、`&Details` 変数の `Tap` イベントで `AttractionDetail_CFPANEL` を呼び出すようにプログラムし、パラメーターで送信された `AttractionId` 識別子に対応する特定の観光名所の詳細が表示されるようにします。

## クライアント側イベントの文法

### コマンド

Composite

<コントロール>.<プロパティ> = <値>

If <ブール式>

Do case... endcase

Do while <ブール式>

Do-sub (Menu for Smart Devices を除く)

選択した行ごと

単純な変数の割り当て: &var = <式>

SDT または BC 要素の割り当て:

&SDT.A = <値>

&BC.A = <値>

Return

Refresh

式内:

変数

項目属性

制約

メソッド

関数

コントロールプロパティ

演算子 (+, -, /, ^)

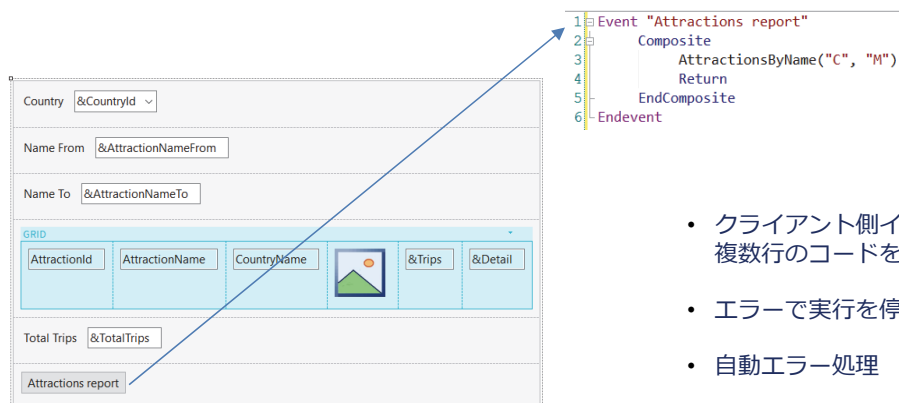
使用不可: 値を返すプロシージャ、  
外部オブジェクト

次に、クライアント側イベントの文法を見てみましょう。

クライアント側イベントについては、サーバー側イベントとは異なり、使用可能なコマンドに関して制限があります。

使用可能なコマンドはここに示すとおりです。これらの詳細については、このコースでは取り上げません。

## Composite コマンド



- クライアント側イベントの場合のみ複数行のコードを使用
- エラーで実行を停止
- 自動エラー処理

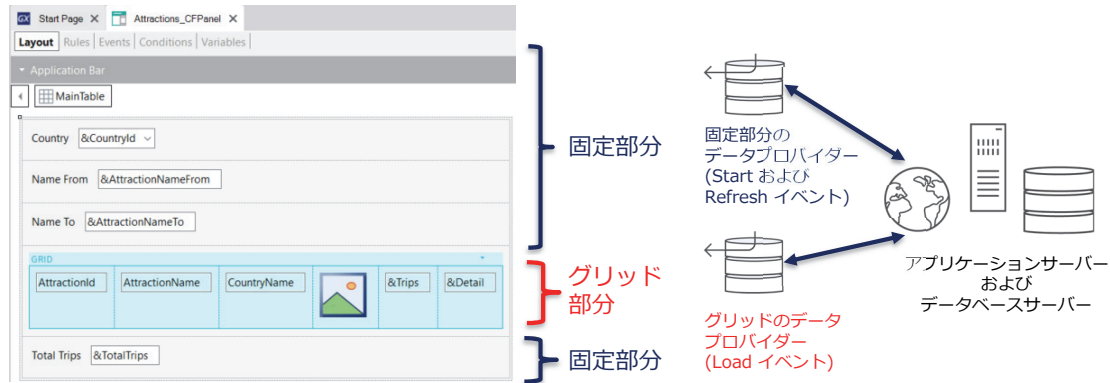
Composite コマンドは、パネルオブジェクトにおけるクライアント側イベントで複数行のコードを記述する必要がある場合に使用します。このケースでは、このコマンド内でイベントのコード全体をグループ化する必要があります。

このコマンドは、呼び出しシーケンスでエラーが発生した場合は実行を停止し、何らかのプログラムを実装しなくても、エラーを自動的に処理して画面に表示するという点で重要です。

これは Web パネルとの大きな違いです。Web パネルの場合、イベントで呼び出されたオブジェクトに起因するエラーが発生しても、実行は中断されず、後続のステートメントに進んでしまいます。また、開発者がエラーを処理して対応策をプログラムする必要があります。



## Panel オブジェクトの各部分



次に、Panel オブジェクトのいくつかの特性を見てみましょう。

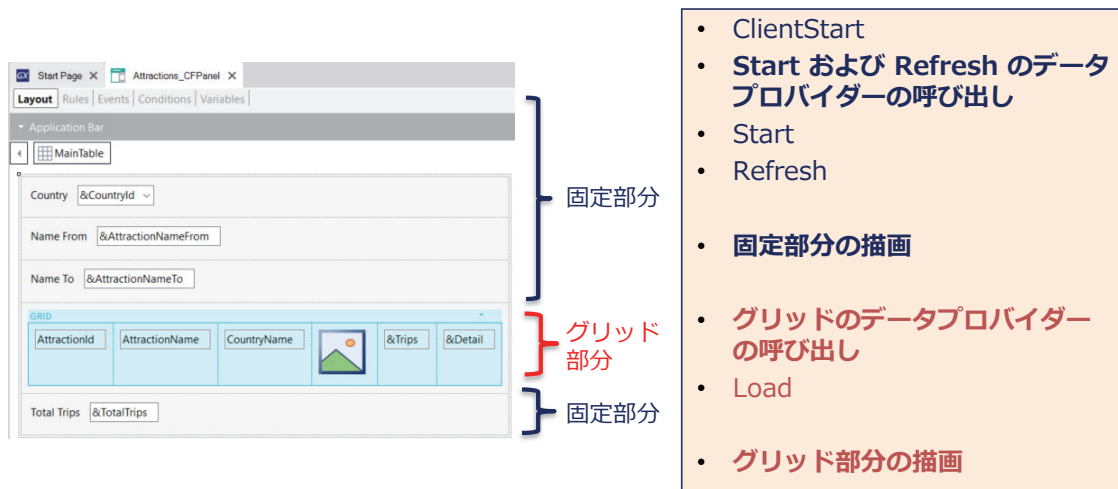
前に言及したとおり、Panel オブジェクトには、2 つの異なる部分があります。1 つ目は固定部分またはフラット部分と呼ばれる、フォーム内のグリッド以外のすべての部分です。画面に表示されている例では、固定部分はフィルタ変数 &CountryId、&AttractionNameFrom、&AttractionNameTo、およびツアーの合計数 &TotalTrips で構成されています。

2 つ目はグリッド部分または可変部分と呼ばれ、この例では、観光名所のデータを含むグリッドで構成されます。

固定部分は必ず存在し、可変部分はパネルのグリッドごとに扱います。

GeneXus では、各部分 (固定およびグリッド) のデータプロバイダーが自動的に生成され、サーバー上のデータアクセス用サービスとして公開されます。これらのデータプロバイダーは、GeneXus で自動的に生成されて保守されるため、ナレッジベースには存在しません。ただし、前に説明した方法でナビゲーション表示を表示すると、各プロバイダーがアクセスするデータを確認できます。

## イベントの実行順序



次に、Panel オブジェクトを実行するとどうなるか見てみましょう。

まず、ClientStart イベントが実行されます。既に説明したとおり、このイベントはクライアントで 1 回だけ実行されます。その後、固定部分のデータアクセスを処理するデータプロバイダーが実行されます。サービスから結果が返されると、それを使用して Start イベントが実行され、サーバー側で Refresh イベントが実行されます。その後、パネルの固定部分が描画されます。

次に、グリッドに必要なデータを取得するために 2 つ目のデータプロバイダーが実行され、グリッドにベーステーブルが存在する場合は、Load イベントが N 回 (レコードごとに 1 回) 実行されます。

画面が 2 つの異なるタイミングで描画されるということは、次に説明することを意味します。

## 固定部分とグリッドのベーステーブルの決定

**固定部分のベーステーブルの決定に関連する項目属性:**

- パネル (フォーム) の固定部分の項目属性
- Refresh イベント内、および固定部分とアプリケーションバーのボタンやコントロールに関連するイベント内の For Each コマンドの外部にある項目属性
- [Conditions] エLEMENTの項目属性

**グリッド部分のベーステーブルの決定に関連する項目属性:**

- グリッドの列の項目属性
- 並べ替え、検索、詳細検索、条件の項目属性
- Load イベント内、およびグリッド内のボタンやコントロールに関連するイベント内の For Each コマンドの外部にある項目属性

Panel オブジェクトでは、並行する 2 つの For Each コマンドが存在するかのよう、固定部分とグリッドでナビゲーションが切り離され、それぞれ異なるベーステーブルが使用されます。これが Web パネルとの重要な違いを生み出します。

固定部分のベーステーブルを決定する際には、フォームの固定部分に属する項目属性、固定部分に関連付けられたイベントに属する項目属性 (Refresh イベント内、および固定部分とアプリケーションバーのボタンやコントロールに関連するイベント内の For Each コマンドの項目属性を除く)、そして Panel オブジェクトの [Conditions] エLEMENTの項目属性が考慮されます。

グリッドのベーステーブルを決定する際には、グリッドの列に含まれている項目属性 (表示と非表示の両方)、グリッドの [Order]、[Search]、[Advanced Search] および [Conditions] プロパティで参照されている項目属性、および Load イベント内、そしてグリッド内のボタンやコントロールに関連するイベント内の For Each コマンドの外部にある項目属性が考慮されます。

## 固定部分とグリッドのベーステーブルの決定

グリッド部分のベーステーブル: Attraction

```

1  Event Load
2      &Trips = Count(TripDate)
3  -Endevent
4
5  Event Refresh
6      &TotalTrips = 0
7      For each Trip.Attraction
8          &TotalTrips += 1
9      Endfor
10 -Endevent
11
12 Event &CountryId.ControlValueChanged
13     Grid1.Refresh()
14 -Endevent
15
16 Event &AttractionNameFrom.ControlValueChanged
17     Grid1.Refresh()
18 -Endevent
19
20
21 Event &AttractionNameTo.ControlValueChanged
22     Grid1.Refresh()
23 -Endevent
24
25 Event Start
26     &Details = "Details"
27 -Endevent
28
29 Event &Details.Tap
30     AttractionDetail_CFPANEL.Call(AttractionId)
31 -Endevent

```

ここで取り上げた例では、フォームの固定部分に項目属性がなく (変数のみ)、ボタンもありません。また、パネルの [Conditions] エlementに項目属性がなく、変数に関連付けられているイベントにも項目属性がありません。Refresh イベントでは、For Each の外部に項目属性はなく、パネルの固定部分にはベーステーブルが存在しません。

グリッドには、AttractionId、AttractionName、CountryName、AttractionPhoto という項目属性があります。Load イベントには Count 式に含まれている項目属性しかないため、ベーステーブルは Attraction になります。

## 詳細情報

<http://wiki.genexus.jp/hwikibypageid.aspx?24829>

この章で取り上げたトピックについて詳しく学習したい場合は、ネイティブ モバイル デバイス (スマートデバイス) 用のプログラミングに関するドキュメントを参照してください。前に説明したとおり、これらのアプリケーションでも Panel オブジェクトが使用されるためです。

詳細情報は、画面に表示されている Wiki ページで確認することができます。



動画	<a href="https://www.genexus.com/community-and-support-jp/training?ja">https://www.genexus.com/community-and-support-jp/training?ja</a>
ドキュメント	<a href="http://wiki.genexus.jp/">http://wiki.genexus.jp/</a>
認定資格	<a href="https://training.genexus.com/certifications">training.genexus.com/certifications</a>