

バックオフィスに重点を置いた Web 画面

Web Panel オブジェクト: 複数のグリッド

GeneXus™

複数グリッドを保有する Web パネル

The screenshot shows the GeneXus IDE interface for a Web Panel named 'ViewCountryInfo'. The panel contains several controls: a text box for '国名' (CountryName), two text boxes for 'カテゴリ番号' (Category Number) with labels '&CategoryIdFrom' and '&CategoryIdTo', a 'GRID' control with columns for '観光名所番号' (AttractionId), '観光名所名' (AttractionName), '観光名所写真' (AttractionImage), '旅行数' (Trips), and a '&NewTrip' button, and a '合計旅行数' (TotalTrips) control. A second 'GRID' control is also visible, with columns for '都市番号' (CityId) and '都市名' (CityName). The 'Events' tab is selected, showing the following code:

```

1 Event Load
2   &Trips = Count(TripDate)
3   &TotalTrips = &TotalTrips + &Trips
4 Endevent
5
6 Event Refresh
7   &TotalTrips = 0
8 Endevent
9
10 Event Start
11   &UpdateImage.FromImage(edit)
12   &NewTrip = "新規旅行"
13 Endevent
14
15 Event &UpdateImage.Click
16   Attraction(TrnMode.Update, AttractionId)
17 Endevent
18
19 Event &NewTrip.Click
20   &Trips = InsertTrip(AttractionId)
21   Refresh
22 Endevent
23

```

The '出力' (Output) window at the bottom shows the following error message:

```

表示: Background Build
===== Specification 開始 =====
Specifying WMAAttractionsScratchOneCountry (1 of 1) ...
エラー spc0089: Event Load cannot be specified when there are multiple grids.

```

Web パネルには、複数のグリッドコントロールを配置することができます。

では、複数のグリッドコントロールがある場合、Load イベントは、どのグリッドコントロールを対象にするのでしょうか。

複数のグリッドコントロールを配置した Web パネルに、Load イベントが定義された状態で保存した場合、[出力] ウィンドウの表示コンボボックスが [Background Build] を選択した場合の表示に次のエラーが表示されます。

spc0089 : Event Load cannot be specified when there are multiple grids.

複数のグリッドがある場合、GeneXus は、Load イベントの対象を決定することが出来ません。

このようなケースの対応方法についてこの資料では説明を行います。

対象グリッドを指定した Load イベント

ViewCountryInfo x

Web Layout | Rules | **Events** | Conditions | Variables | Help

Events

```

1 Event Grid1.Load
2   &Trips = Count(TripDate)
3   &TotalTrips = &TotalTrips + &Trips
4 Endevent
5
6 Event Refresh
7   &TotalTrips = 0
8 Endevent
9
10 Event Start
11   &UpdateImage.FromImage(edit)
12   &NewTrip = "新規旅行"
13 Endevent
14
15 Event ブロック (&UpdateImage.Click)
16
17
18
19 Event ブロック (&NewTrip.Click)
20
21
22
23
24 Event ブロック (AttractionName.Click)
25
26
27

```

国名 フランス

カテゴリ番号 (開始)

カテゴリ番号 (終了)

都市番号	都市名
1	パリ
2	ニース

観光名所名	観光名所写真	旅行数	
ルーブル美術館		3	新規旅行
エッフェル塔		2	新規旅行
マティス美術館		1	新規旅行
ノートルダム寺院		2	新規旅行

合計旅行数 8

グリッドコントロールが複数配置した場合、前述の通り、「Load」という記述だけのイベントの場合、どのグリッドコントロールを対象とするか GeneXus が判断できないため、エラーとなりました。

この場合、解決方法として、対象のグリッドコントロールを指定した Load イベントを記述します。

イベント名のフォーマットとしては、<グリッドコントロール名>.Load となります。グリッドコントロールの名前は、プロパティから指定できますが、既定のままの場合、初めて配置したグリッドコントロールは、Grid1 となり、配置されたグリッドコントロールが増えるたび、末尾の数字がカウントアップされていきます。

また、複数のグリッドコントロールを配置した場合、それぞれのグリッドは独立したデータベースへのアクセスとなります。

各グリッドの Load イベント

The screenshot shows the GeneXus IDE interface for a web application named 'ViewCountryInfo'. The main workspace displays a web layout with several components: a 'CountryName' input field, two category range inputs ('&CategoryIdFrom' and '&CategoryIdTo'), a main grid listing attractions with columns for 'AttractionId', 'AttractionName', a photo icon, and '&Trips', and a '&NewTrip' button. Below this is a summary row for '合計旅行数' ('&TotalTrips'). To the right, a smaller grid displays city information with columns for 'CityId', 'CityName', and '&Attractions'. The 'Events' panel on the right defines the 'Load' events for these grids. The 'Event Grid2.Load' is expanded, showing the following logic:

```

Order: CountryId
Index: ICOUNTRYCITY
Start from: CountryId = @CountryId
filters: CountryId = @CountryId
Join location: Server

CountryCity ( CountryId, CityId )
-count( AttractionName ) navigation
Attraction ( CountryId, CityId )
  
```

グリッドコントロール名を指定することで、各グリッドコントロールに対応した Load イベントを定義することができます。

そして、前述の通り、各グリッドは独立してデータベースへアクセスするため、各グリッドコントロールに対する Load イベントもそれぞれのグリッドコントロールのデータベースアクセスに準じて実行されます。

そのため、グリッドコントロール間で影響を与えることはありません。
この点は、ナビゲーション表示からも確認できます。

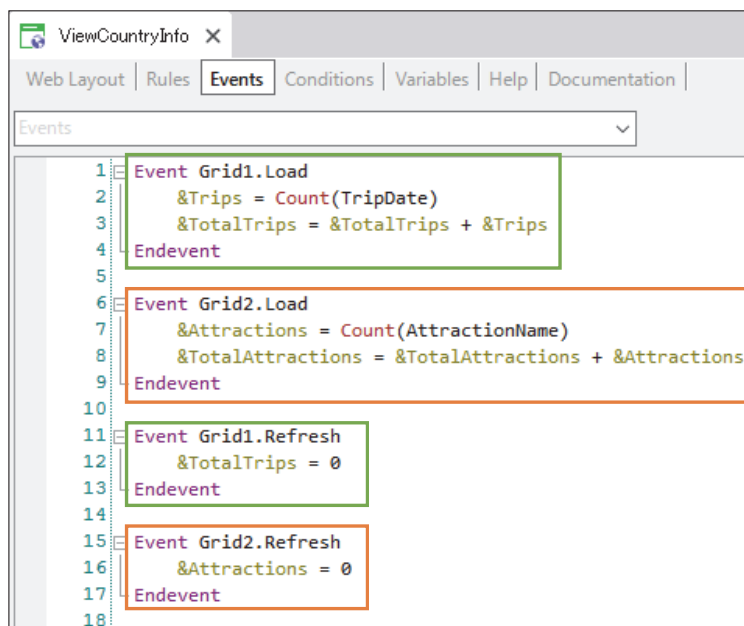
複数グリッドを保有する Web パネルの Refresh イベント

The screenshot shows the GeneXus IDE interface for a web panel named 'ViewCountryInfo'. The main workspace displays a form layout with several controls: 'CountryName', 'CategoryIdFrom', 'CategoryIdTo', a main grid 'GRID' with columns for 'AttractionId', 'AttractionName', 'AttractionImage', 'Trips', and 'NewTrip', and a summary row 'TotalTrips'. To the right, a secondary grid 'Grid2' is shown with columns for 'CityId', 'CityName', and 'Attractions', along with a summary row 'TotalAttractions'. The 'プロパティ' (Properties) window on the right is open for 'Grid2', showing its 'Conditions' property set to 'CityName Like ...'. A red arrow points from this ellipsis to the 'Grid2のConditions' dialog box at the bottom, which contains the condition: 'CityName Like &CityName When not &CityName.IsEmpty();'. The 'OK(0)' and 'キャンセル' buttons are visible at the bottom of the dialog.

複数グリッドがある場合の Load イベントにフォーカスをしてきました。
Load イベントを含むシステムイベントについて、説明を進める中で Refresh イベントが必ず Load イベントの前に実行されていました。

では、複数のグリッドがある中で、いくつかのグリッドコントロールで、再表示が行われるタイミングで実行したい処理ができた場合、これまで通り Refresh イベントを利用すると、意図しない結果になる場合があります。
そのため、Refresh イベントも各グリッドコントロールを対象にした Refresh イベントが記述可能です。

各グリッドの Refresh イベント



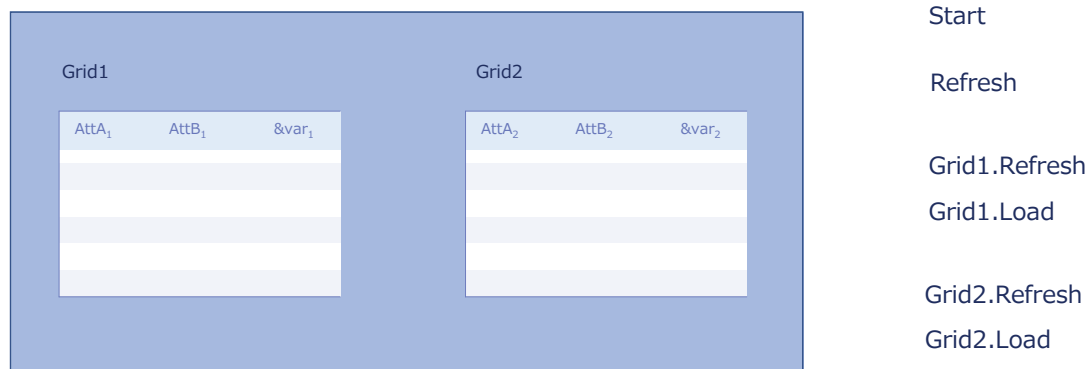
これまでに利用していた Refresh イベントは、画面全体を対象に、再表示が発生するたびに実行されるイベントでした。

そのため、グリッドの再表示に関する処理が含まれ、グリッドコントロールが複数となった場合、前のスライドで記載のような意図しない結果となる場合があります。

もし、各グリッドが再表示される場合にのみ必要となる処理であれば、対象のグリッドの再表示時に実行される <グリッドコントロール名>.Refresh イベントを利用します。

このように、Refresh イベントの使い分けも重要です。

システムイベントの実行順序



既に、システムイベントの実行順序についての説明を行っていましたが、グリッドコントロールごとの Refresh イベント、Load イベントが含まれている場合について説明しています。

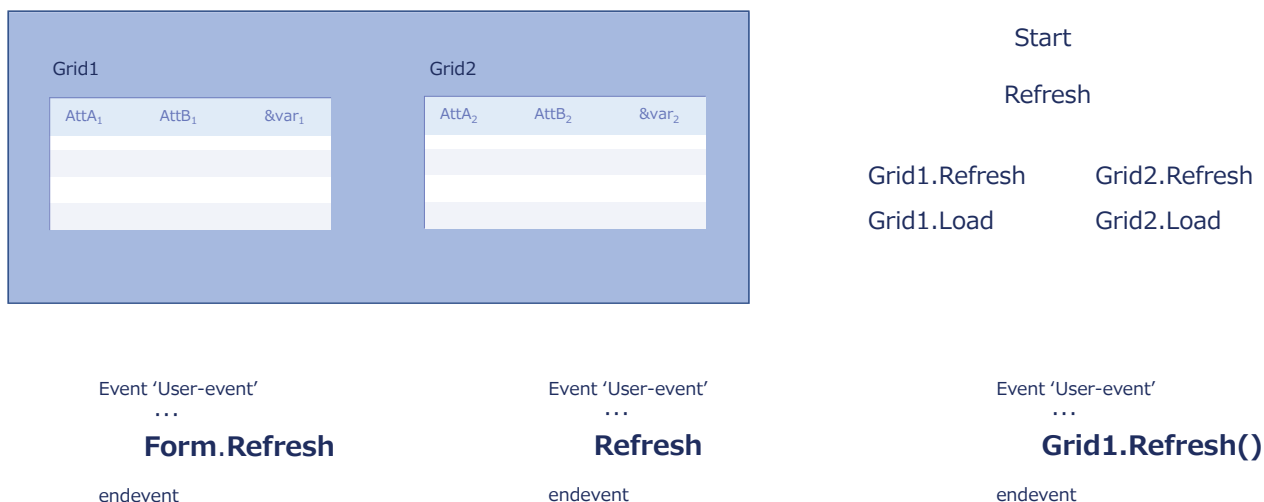
ここでは、グリッドコントロールが 2 つの場合を例に利用しています。

この場合、Web パネルが初めて実行されるときにイベントは、下記の順番で実行されます。

1. Start イベント
2. 汎用（画面全体を対象）の Refresh イベント
3. 1 つ目のグリッドコントロールの Refresh イベント
4. 1 つ目のグリッドコントロールの Load イベント
5. 2 つ目のグリッドコントロールの Refresh イベント
6. 2 つ目のグリッドコントロールの Load イベント

もし、グリッドコントロールがベーステーブルを持つ場合、Load イベントは、表示する行数分実行され、ベーステーブルを持たない場合、1 回だけ表示されます。この点は、対象のグリッドコントロールを指定していない場合と同様です。

Refresh コマンドの対象



ここまでのコースで、任意のタイミングで、画面の再表示を実行するための Refresh コマンドについて説明していました。

この Refresh コマンドも <グリッドコントロール名>.Refresh() のように記述することで、汎用の Refresh イベントからの実行でなく、指定されたグリッドコントロールに対する Refresh イベントと、Load イベントのみを実行させることができます。

引き続き Refresh コマンドも利用することができます。

この場合、画面の再表示が強制的に行われるため、Start イベント以外のシステムイベントすべてが実行されます。

また、ここまでに説明していませんでしたが、Form.Refresh というコマンドを実行した場合、Start イベントからシステムイベントを実行させることができます。

つまり、画面をはじめて表示した場合と同じロジックを呼び出すことができます。

Load コマンドの対象



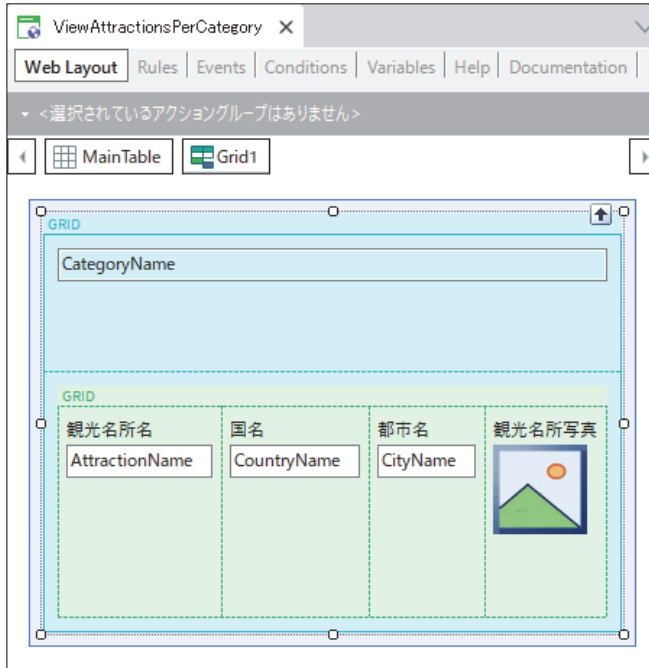
ベーステーブルが無いグリッドを対象とした Load イベントの場合、Load コマンドによって行として描画する内容を確定していました。

このコマンドは、Refresh コマンドのように対象のグリッドコントロール名を指定する必要はありません。

その理由としては、Load コマンドを記述するイベントは、対象のグリッドコントロールが特定された状態となるためです。

ただし、もしも任意のユーザーイベントなどからグリッドの行を追加する処理を実装したい場合、<グリッドコントロール名>.Load() というコマンドを記述する必要があります。

グリッドのネスト: フリー スタイル グリッド



ミュージアム			
観光名所名	国名	都市名	観光名所写真
ルーブル美術館	フランス	パリ	
スミソニアン博物館	アメリカ	ワシントンD.C.	
マティス美術館	フランス	ニース	
遺跡			
観光名所名	国名	都市名	観光名所写真
万里の長城	中国	北京	
紫禁城	中国	北京	
モニュメント			
観光名所名	国名	都市名	観光名所写真
サグラダ・ファミリア	スペイン	バルセロナ	
有名なランドマーク			
観光名所名	国名	都市名	観光名所写真

ここまでで説明してきたケースは、並列のグリッドと呼ぶことができる関連性の無いグリッドコントロールを配置するものでした。

しかし、他の実装として、グリッドコントロールをネストし、関連性のあるグリッドコントロールを配置することもできます。

ただし、この実装を行うためには、ここまでで利用していたグリッドコントロールだけでは実装出来ません。

この理由として、グリッドコントロールは、項目属性または変数のみを含めることができるためです。

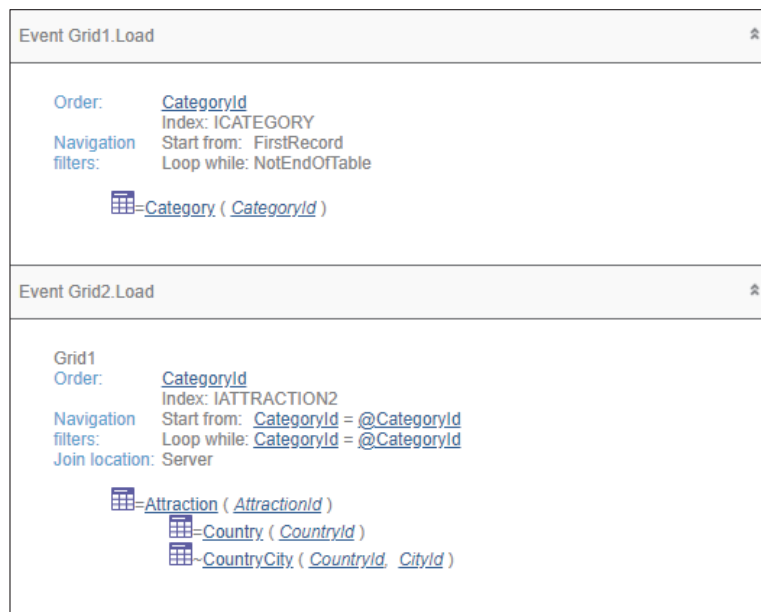
そのため、グリッドコントロールを内包するグリッドを実装するためには、「フリースタイルグリッド」というタイプのグリッドコントロールを利用します。

フリースタイルグリッドコントロールは、名前の通りフリースタイルにグリッドとして繰り返し表示する内容を定義することができます。

そのため、フリースタイルグリッドコントロール内には、項目属性、変数以外のコントロールも配置することができ、その 1 つに標準のグリッドコントロールも含まれます。

標準のグリッドコントロールを配置した場合、ネストされた For each コマンドのように外側と内側のグリッドタイプのコントロールに設定されるベーステーブルの関係性により、データが表示されます。

グリッドのネスト: ナビゲーション表示



ネストされたグリッドがあるオブジェクトのナビゲーション表示を確認すると、自動的に絞り込みが実装されていることが確認できます。

ただし、これは外側のグリッドと内側のグリッドに設定されたベーステーブルに関連性がある場合のみです。

この点もネストされた For Each コマンドの場合と似ています。

ただし、For each コマンドの場合と異なる点として、ナビゲーション表示自体がネストされているような表示ではなく、各グリッドコントロールに対する並行した表示になります。

その中で、Navigation filters において、絞り込みの値（イコールの右辺）が「@」から始まる場合、外側のグリッドコントロールで表示したデータを利用していることが確認できます。

ネストされたグリッドの実装では、このような単純な 1 対 N のデータ表示だけでなく、For each コマンドで実装したような複雑な実装も行えますが、本コースでは、対象外となり、説明はここまでとします。

GeneXus[™]