

Web Panel オブジェクト

概要

GeneXus™

Web Panel オブジェクト

Enter AttractionsFilter X

Web Layout Rules Events Conditions Variables Help Documentation

< 選択されているアクショングループはありません >

MainTable

国番号 &CountryId

カテゴリ番号 (開始) &CategoryIdFrom

カテゴリ番号 (終了) &CategoryIdTo

PDF出力

変数: 入力コントロール (読み取り専用ではない)

Web パネルは、GeneXus において、もっとも多目的に利用できるオブジェクトです。ここまでの機能の説明でも部分的に利用していました。

Web パネルには、インタフェースをデザインするための [Web Layout] エlementが用意されています。

このElementを利用し、目的に合わせた機能を実装することができます。

Web パネルで定義された変数は、[Web Layout] Elementに配置すると、入力コントロールとして生成されます。

つまり、読み取り専用ではないコントロールとなります。

入力コントロールのタイプ

The screenshot illustrates the configuration of a 'Dynamic Combo Box' control in the GeneXus IDE. The 'Web Layout' tab shows a form with three input controls. The '国番号' control is highlighted with a red arrow pointing to the 'プロパティ' (Properties) window. The 'プロパティ' window shows the '項目属性/変数: &CountryId' (Item Properties/Variable: &CountryId) section. The 'Control Type' is set to 'Dynamic Combo Box'. The 'Data Source From' is 'Attributes', 'Item Values' is 'CountryId', and 'Item Descriptions' is 'CountryName'. To the right of the properties window, a database icon labeled 'COUNTRY' is shown with an arrow pointing to the 'CountryName' property.

[Web Layout] エlementに配置された変数は、[Control Type] プロパティを変更することで、様々な入力コントロールを実装することが可能です。

この選択可能なコントロールタイプのうち、[Dynamic Combo Box] および [Dynamic List Box] タイプは、[Combo Box] と [List Box] に機能が追加されたコントロールタイプです。

通常、これらのコントロールタイプでは、事前に選択可能な値を設定しておく必要がありますが、項目属性を指定するプロパティを利用することで、アプリケーション実行時に動的に選択肢となる値をデータベースから読み込みます。

[Item Values] プロパティには、変数に格納する値となる項目属性を指定し、[Item Descriptions] プロパティには、画面上に表示する際にユーザーがわかりやすい値となる項目属性を指定します。

例: Web パネル内の変数

The screenshot illustrates the integration between a Web Layout and its Events in GeneXus. The Web Layout (left) contains a 'MainTable' with a 'PDF出力' button. The Events panel (right) shows an event 'CallAttractions' that calls the 'AttractionsByCategoryAndCountry' rule. The rule's parameters are linked to variables in the Web Layout: &CountryId, &CategoryIdFrom, and &CategoryIdTo.

Web Layout (MainTable):

- 国番号: &CountryId
- カテゴリ番号 (開始): &CategoryIdFrom
- カテゴリ番号 (終了): &CategoryIdTo
- PDF出力

Events:

```

1 Event ブロック ('CallAttractionList')
4
5 Event ブロック ('CallAttractionsByCategory')
8
9 Event 'CallAttractions'
10   AttractionsByCategoryAndCountry(&CountryId,&CategoryIdFrom,&CategoryIdTo)
11 Endevent
12

```

AttractionsByCategoryAndCountry Rules:

```

1 Parm(IN:&CountryId, IN:&IdFrom, IN:&IdTo);
2

```

変数に入力された値は、様々な用途で利用できます。

例えば、Web パネルにおいて、手続き型の処理を記述できる [Events] エlementにおいて、別のオブジェクトを呼び出すパラメーターに利用します。

[Events] Elementには、[Web Layout] Elementに配置されたコントロールに紐づく「イベント」を定義することができ、ユーザーの操作によって実行する処理を定義できます。

また、定義できるイベントには、一部システムイベントもあり、後ほど説明を行います。

PDF 出力と同等の Web アプリケーション表示

The screenshot displays the GeneXus IDE interface for a web application. The main window shows a layout design with three main sections: a title bar, a header, and a table of attractions. The header section contains a suitcase icon and the title '観光名所一覧'. The table section has columns labeled '番号', '名前', '国', and '写真'. Below the table, there is a section for attractions with columns 'Attr', 'AttractionName', and 'CountryName'. The 'Source' tab is open, showing the following code:

```

1 Print PBTitle
2 Print PBColumnTitles
3
4 For each Attraction
5   Order CountryName, AttractionName
6   Where CategoryId >= &IdFrom When not &IdFrom.IsEmpty()
7   Where CategoryId <= &IdTo When not &IdTo.IsEmpty()
8   Where CountryId = &CountryId When not &CountryId.IsEmpty()
9   Print PBAttractions
10 Endfor
11

```

ここまでの実装を行ってきた PDF 出力の機能を Web パネルでも実装することができます。

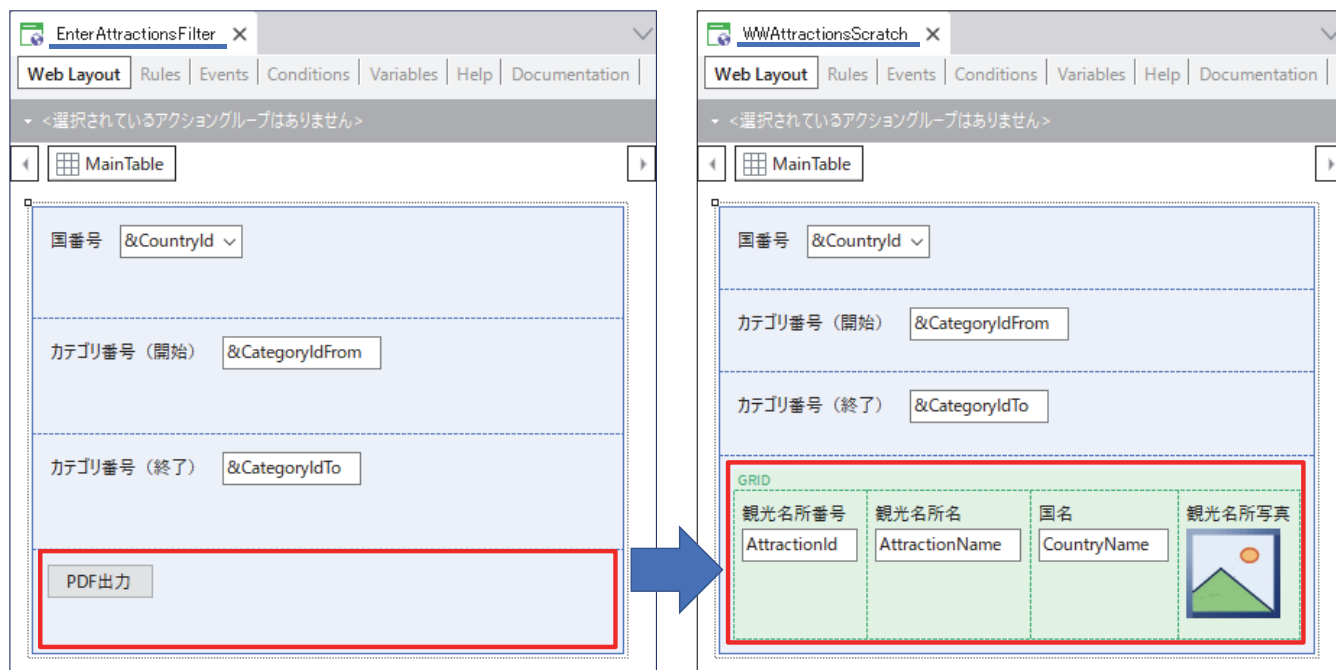
プロシージャオブジェクトの [Layout] エLEMENTと同等のインターフェースを Web パネルでデザインし、[Source] ELEMENTに記述された For each コマンドと同等の処理が行われるように実装します。

もちろん PDF 出力の実装もアプリケーションには必要です。

ユーザーから印刷可能なフォーマットでリストを出力したいという要望があれば、PDF 出力で実装するべきです。

そうではなく、動的に登録されたデータを表示したいという要望の場合、Web パネルで実装します。

対話形式：一覧の表示



「対話形式」とは、Web アプリケーションで変数に値を入力すると、その値に基づき、処理が行われることを意味しています。

この点も含め、Web パネルの実装について説明を続けます。

Web パネルで、データを一覧表示するためには、グリッドコントロールを利用します。グリッドコントロールを配置する場合、列として含める項目属性や変数を選択するダイアログが表示されるため、対象をすべて選択します。

項目属性を選択した場合、Web パネルの [Web Layout] エlementでは、グリッドの内外にかかわらず、読み取り専用のコントロールとして配置されます。

項目属性に値を入力し、データ操作が行える画面は、トランザクションオブジェクトからのみ生成することができます。

対話形式：グリッドコントロールの実行結果

国番号

カテゴリ番号 (開始)

カテゴリ番号 (終了)


観光名所番号	観光名所名	国名	観光名所写真
15	ルーブル美術館	フランス	
16	エッフェル塔	フランス	
17	サグラダ・ファミリア	スペイン	
18	スミソニアン博物館	アメリカ	
19	マティス美術館	フランス	
20	ノートルダム寺院	フランス	
21	万里の長城	中国	
22	紫禁城	中国	

Web Panel WWAttractionsScratch Navigation Report

Name:  WWAttractionsScratch
 Description: WWAttractions Scratch

Environment:  Default (.NET)
 Spec. Version:  18_0_10-184442
 Form Class: HTML
 Program Name: WWAttractionsScratch

FILL &CountryId WITH CountryId, CountryName IN

=Country (CountryId) INTO CountryId CountryName
 ORDER CountryName

Event Load

Order: AttractionId
 Index: IAttraction
 Navigation Start from: FirstRecord
 filters: Loop while: NotEndOfTable
 Join location: Server

=Attraction (AttractionId)
=Country (CountryId)

グリッドコントロールを配置しただけの状態でも、Web アプリケーションを実行した場合、データが一覧で表示されます。

この時、表示されるデータは、すべてのデータが対象となり、主キー項目の値で昇順に並べられています。

この点は、ナビゲーション表示からも確認できます。

ナビゲーション表示では、すべてのデータベースへの参照が表示されるため、ダイナミックコンボボックスに関する記述も含まれています。

対話形式：グリッドのベーステーブル

The screenshot shows the GeneXus IDE with a web layout named 'WWAttractionsScratch'. The layout contains a grid control named 'Grid1'. The grid has four columns: '観光名所番号' (AttractionId), '観光名所名' (AttractionName), '国名' (CountryName), and '観光名所写真' (AttractionImage). The grid is connected to a data source named 'Attraction'.

The 'プロパティ' (Properties) window for 'Grid1' is open, showing the 'Base Trn' property set to 'Attraction'.

The code snippet for the grid is as follows:

```

For each Attraction
  Order CountryName, AttractionName
  Where CategoryId >= @IdFrom When not @IdFrom.IsEmpty()
  Where CategoryId <= @IdTo When not @IdTo.IsEmpty()
  Where CountryId = @CountryId When not @CountryId.IsEmpty()
  Print PBAAttractions
Endfor

```

GeneXus は、ベーステーブルの指定が明確に行われていない場合、自動で対象となる項目属性のセットに基づき、ベーステーブルを決定します。

この場合、セットに含まれる項目属性の値すべてを参照可能なテーブルとその関係性を検証し、すべてのテーブルから最も N の関係となるテーブルをベーステーブルとして扱います。

もちろんベーステーブルが決定した場合、拡張テーブルのデータも参照することができます。

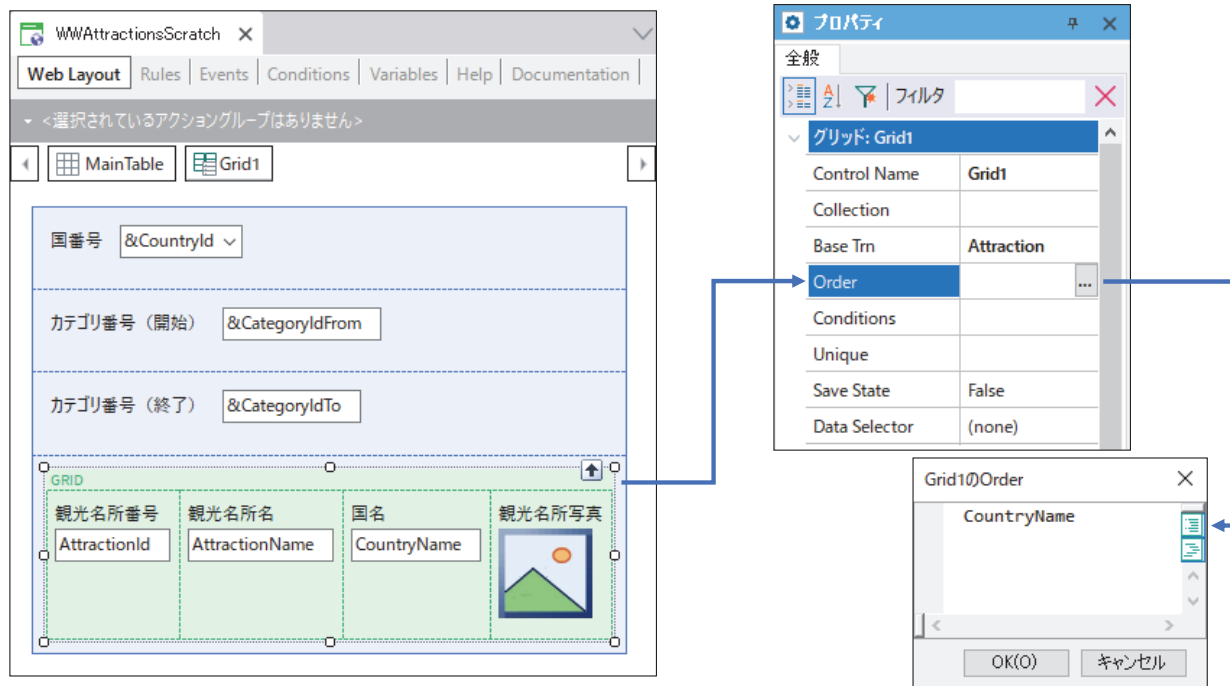
今回のグリッドコントロールは、このケースに該当し、列として配置された項目属性に基づき、ベーステーブルが決定され、データを取得しました。

これは、プロシージャオブジェクトで、For each コマンドを記述した場合と同様の結果でした。

グリッドコントロールに対し、明確にベーステーブルを指定したい場合、グリッドコントロールの [Base Trn] プロパティに、トランザクションオブジェクト名を記述します。

もし、第 2 レベルを対象にする場合には、トランザクションオブジェクト名に続き、ピリオドとレベル名を記述します。

対話形式：並び替えの指定



For each コマンドを記述した場合と同様の結果を出力できるグリッドコントロールには、他にも For each コマンドと同様の定義を行うためのプロパティが用意されています。例えば、表示するデータの並び替えを指定する [Order] プロパティがあります。このプロパティで、並び替えの基準としたい項目属性を指定します。

これは、For each コマンドにおける Order 節と同じ役割をします。

対話形式：並び替えの実行結果

国番号	(なし) ▼
カテゴリ番号 (開始)	0
カテゴリ番号 (終了)	0

観光名所番号	観光名所名	国名	観光名所写真
18	スミソニアン博物館	アメリカ	
17	サグラダ・ファミリア	スペイン	
15	ルーブル美術館	フランス	
16	エッフェル塔	フランス	
19	マティス美術館	フランス	
20	ノートルダム寺院	フランス	
21	万里の長城	中国	
22	紫禁城	中国	

Web Panel WWAttractionsScratch Navigation Report

Name: [WWAttractionsScratch](#)
Description: WWAttractions Scratch

Environment: [Default \(.NET\)](#)
Spec. Version: [18_0_10-184442](#)
Form Class: HTML
Program Name: WWAttractionsScratch

Warnings

⚠ **spc0038** There is no index for order [CountryName](#); poor performance may be noticed in grid 'Grid1'.

FILL &CountryId WITH [CountryId](#), [CountryName](#) IN
[Country](#) ([CountryId](#)) INTO [CountryId](#) [CountryName](#)
 ORDER [CountryName](#)

Event Load

Order: [CountryName](#)
 No index!

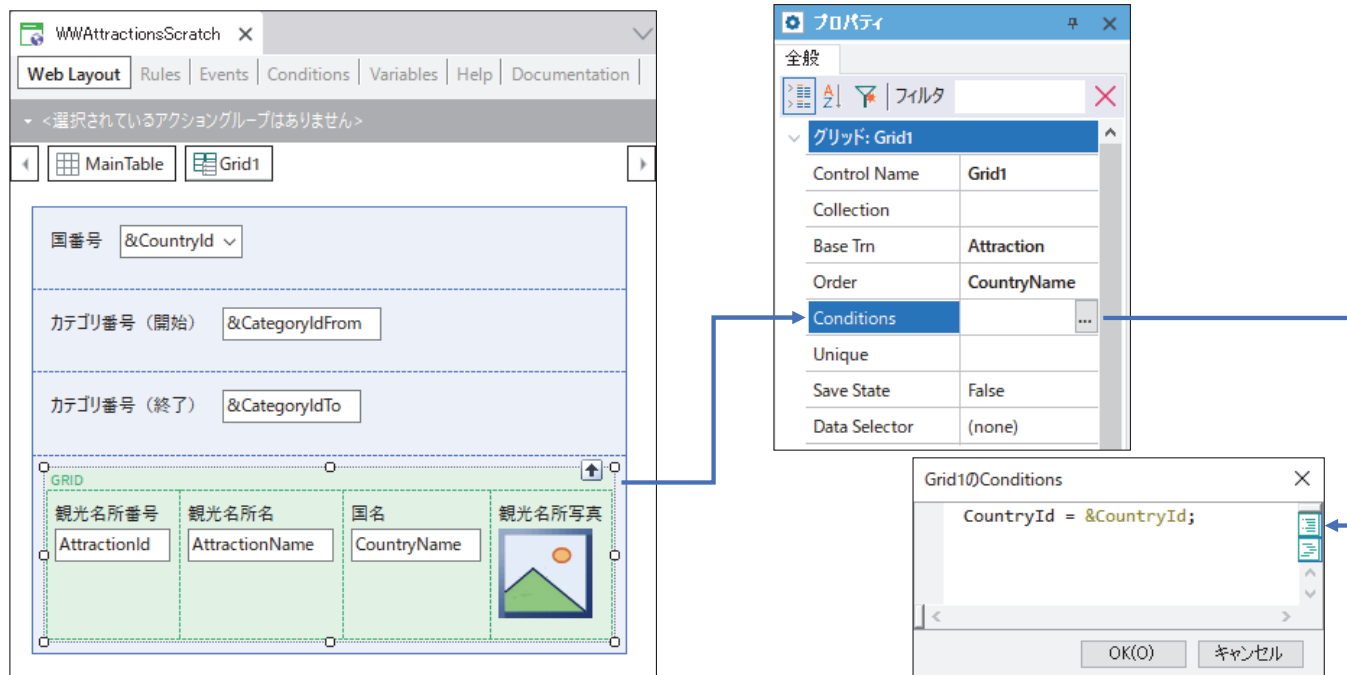
Navigation Start from: FirstRecord
filters: Loop while: NotEndOfTable
Join location: Server

[Attraction](#) ([AttractionId](#))
[Country](#) ([CountryId](#))

[Order] プロパティを設定したことで、実行時の表示順が変更できました。
 この場合、ナビゲーション表示にも対応した表示に切り替わります。

また、改めて確認を行うと、グリッドコントロールにデータを読み込むための
 ベーステーブルなどの情報も確認できます。
 ここには、[Base Trn] プロパティに設定したトランザクションに関連するテーブルが
 ベーステーブルとなったことを示しています。

対話形式：絞り込みの実装



対話形式の実装について、前述の通り、Web アプリケーションで変数に値を入力すると、その値に基づき、処理が行われるということを意味していました。具体的なケースとしては、入力された値を、グリッドに表示するデータの絞り込みに利用するケースが想定されます。

このような利用のためには、グリッドコントロールが持つ [Conditions] プロパティに条件式を記述します。

1 つの条件の記述が完了したら、セミコロンを入力する必要があります。

これは、For each コマンドにおける Where 節と同じ役割をします。

対話形式：絞り込みの実行結果

国番号

カテゴリ番号 (開始)

カテゴリ番号 (終了)

観光名所番号	観光名所名	国名	観光名所写真
--------	-------	----	--------


国番号


カテゴリ番号 (開始)

カテゴリ番号 (終了)

観光名所番号	観光名所名	国名	観光名所写真
15	ルーブル美術館	フランス	
16	エッフェル塔	フランス	
19	マティス美術館	フランス	
20	ノートルダム寺院	フランス	

Web Panel WWAttractionsScratch Navigation Report

Name:  WWAttractionsScratch
 Description: WWAttractions Scratch

Environment:  Default (.NET)
 Spec. Version:  18_0_10-184442
 Form Class: HTML
 Program Name: WWAttractionsScratch

Warnings

 spc0038 There is no index for order CountryName; poor performance may be noticed in grid 'Grid1'.

FILL &CountryId WITH CountryId, CountryName IN

=Country (CountryId) INTO CountryId CountryName
 ORDER CountryName

Event Load

Order: CountryName
 No index
 Navigation Start from: FirstRecord
 filters: Loop while: NotEndOfTable
 Constraints: CountryId = &CountryId
 Join location: Server

=Attraction (AttractionId)
=Country (CountryId)

[Conditions] プロパティを設定したことで、実行時に絞り込みを行うことができました。対話形式となるため、条件に設定した変数の値が変更されると、その瞬間にデータの再読み込みが行われます。

また、ナビゲーション表示にも対応した表示が含まれます。

対話形式：絞り込みのトリガー条件

The screenshot shows the GeneXus Web Layout editor for a project named 'WWAttractionsScratch'. The main window displays a form layout with the following elements:

- A dropdown menu for '国番号' (Country Code) with the value '&CountryId'.
- A text input for 'カテゴリ番号 (開始)' (Category Number (Start)) with the value '&CategoryIdFrom'.
- A text input for 'カテゴリ番号 (終了)' (Category Number (End)) with the value '&CategoryIdTo'.
- A grid control labeled 'GRID' containing four columns:
 - 観光名所番号 (Attraction Id) with the value 'AttractionId'.
 - 観光名所名 (Attraction Name) with the value 'AttractionName'.
 - 国名 (Country Name) with the value 'CountryName'.
 - 観光名所写真 (Attraction Photo) with a placeholder image.

On the right, the 'プロパティ' (Properties) window is open for 'グリッド: Grid1'. The 'Conditions' property is set to 'CountryId = &CountryId'. Below this, a dialog box titled 'Grid1のConditions' shows the condition: 'CountryId = &CountryId When not &CountryId.IsEmpty();'. The 'OK(O)' and 'キャンセル' (Cancel) buttons are visible at the bottom of the dialog.

For each コマンドにおける Where 節と同様に、絞り込みのトリガー条件を指定することができます。

[Conditions] プロパティに指定した条件に、For each コマンドにおける Where 節同様に、When 節を追加し、絞り込みを有効にする条件を指定します。

対話形式：複数条件の指定



[Conditions] プロパティでは、1 つの条件を記述後、セミコロンを入力しました。これは、セミコロンで区切り、複数の条件を記述できることを意味します。

そのため、必要な分の条件を記述することができます。
この実装は、For each コマンドにおける複数の Where 節を記述した場合と同様に、すべて And 結合の条件となります。

グリッドの読み込み



For each Attraction
 <メインのコード> ← **Print PBAttractions**
 Endfor

PBAttractions			
AttractionId	AttractionName	CountryName	AttractionImage

グリッドコントロールの機能についての説明を進めましたが、列として項目属性を含めることで、自動でデータが読み込まれることを説明しました。
 つまり、グリッドコントロールの配置以外に処理の記述は行っていませんでした。

プロシージャオブジェクトによる PDF 出力の場合、データベースを参照し、該当するレコードを出力する処理を記述しました。

検討事項：変数による表示の追加

WWAttractionsScratch

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

<選択されているアクショングループはありません>

MainTable | Grid1

国番号 &CountryId

カテゴリ番号（開始） &CategoryIdFrom

カテゴリ番号（終了） &CategoryIdTo

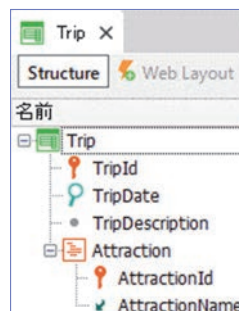
GRID

観光名所番号	観光名所名	国名	観光名所写真
AttractionId	AttractionName	CountryName	



```
For each Attraction
  &Trips = Count(TripDate)
  Print PBAttractions
Endfor
```

AttractionName	CountryName	AttractionId	Trip



プロシージャオブジェクトによる PDF 出力の場合、読み込み処理に対する任意のカスタマイズが可能です。

例えば、プリントブロックに変数が配置された場合、For each コマンド内で変数に値を代入することで、PDF 出力時には、変数に格納された値が表示されます。

では、グリッドコントロールの場合の実装方法について説明を進めます。

グリッド読み込みの処理 : Load イベント

WWAttractionsScratch

Web Layout Rules Events Conditions Variables Help Documentation

<選択されているアクショングループはありません>

MainTable Grid1 Trips

国番号 &CountryId

カテゴリ番号 (開始) &CategoryIdFrom

カテゴリ番号 (終了) &CategoryIdTo

GRID

観光名所番号	観光名所名	国名	観光名所写真	旅行数
AttractionId	AttractionName	CountryName		&Trips

For each Attraction
 &Trips = Count(TripDate)
 Print PBAttractions
 Endfor

Event Load
 <コード>
 Endevent

PBAttractions

AttractionName	CountryName	Trips

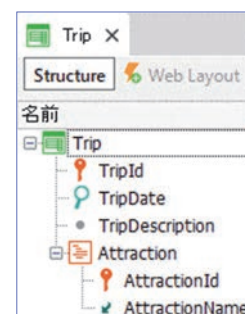
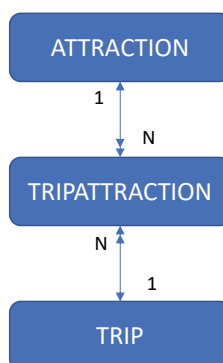
グリッドコントロールで表示する一覧も変数を含めることができます。
 配置済みのグリッドコントロールに列を追加したい場合、2 つの方法から選択できます。
 1 つは、グリッドコントロールを右クリックし、「項目属性/変数を挿入」を選択し、表示されるダイアログから追加したい項目属性や変数を選択します。
 もう 1 つは、[ツールボックス] ウィンドウより「項目属性/変数」をグリッドコントロール内へドラッグし、表示されるダイアログから追加したい項目属性や変数を選択します。

この操作によって、前のページで例として挙げたようにプリントブロックに変数を追加する場合と同様に、インタフェースへの変数が追加できます。

追加した列が変数の場合、値を代入しなければ、表示される値はなく、空の値となります。
 プロシージャオブジェクトの場合、For each コマンド内で割り当ての処理を記述することで、各表示レコードごとに異なる値を表示することが出来ました。
 Web パネルの場合、ここまでにグリッドコントロールに関連するコーディングは行っていません。

このような場合には、システムイベントの Load イベントを利用します。

Load イベントの実装



Event Load
&Trips = Count(TripDate)
Endevent

Load イベントは、グリッドコントロールの定義に基づき、1 行の出力を行う直前に実行される処理を記述します。

この処理の内容がグリッドコントロールの表示対象となるデータ分繰り返し実行されます。

これは、For each コマンド内に記述された処理と似た実行タイミング、実行回数となります。

また、実行される処理は、グリッドコントロールに対し、決定されたベーステーブルを処理のベーステーブルとして利用します。

Load イベントの実装結果

国番号

カテゴリ番号 (開始)

カテゴリ番号 (終了)

観光名所番号	観光名所名	国名	観光名所写真	旅行数
18	スミソニアン博物館	アメリカ		0
17	サグラダ・ファミリア	スペイン		0
15	ルーブル美術館	フランス		2
16	エッフェル塔	フランス		2
19	マティス美術館	フランス		1
20	ノートルダム寺院	フランス		2
21	万里の長城	中国		3
22	紫禁城	中国		2

```

WWAttractionsScratch
Web Layout | Rules | Events | Conditions | Variables
Events
1 Event Load
2   &Trips = Count(TripDate)
3 Endevent
4

```

```

Trip_DataProvider
Source | Rules | Variables | Help | Documentation
1 TripCollection{
2   Trip
3   {
4     TripDate = &Today
5     TripDescription = "フランス周遊"
6     Attraction
7     {
8       Attraction
9       {
10        AttractionId = Find(AttractionId,AttractionName = "ルーブル美術館")
11      }
12      Attraction
13      {
14        AttractionId = Find(AttractionId,AttractionName = "エッフェル塔")
15      }
16    }
17  }
18 }

```

Load イベントを実装したことで、行の読み込み時の処理を実装出来ることが確認できました。

これにより、各行の変数に値を代入し、表示することが出来ました。

Load イベントは、グリッドコントロールの行を表示するタイミングで実行されますが、処理の対象は、グリッド内ではありません。

グリッド外の変数に総数を表示

The screenshot shows the GeneXus web layout editor for a project named 'WWAttractionsScratch'. The 'Web Layout' tab is active. Below the menu, there is a message: '<選択されているアクショングループはありません>'. The layout includes a 'MainTable' and a 'TotalTrips' variable. The form contains several input fields: '国番号' (Country Number) with a dropdown '&CountryId', 'カテゴリ番号 (開始)' (Category Number (Start)) with '&CategoryIdFrom', and 'カテゴリ番号 (終了)' (Category Number (End)) with '&CategoryIdTo'. A 'GRID' section contains five columns: '観光名所番号' (Attraction Number) with 'AttractionId', '観光名所名' (Attraction Name) with 'AttractionName', '国名' (Country Name) with 'CountryName', '観光名所写真' (Attraction Photo) with a placeholder image, and '旅行数' (Number of Trips) with '&Trips'. At the bottom, there is a '合計旅行数' (Total Number of Trips) field with '&TotalTrips'.

Event Load
 $\&Trips = \text{Count}(\text{TripDate})$
 $\&TotalTrips = \&TotalTrips + \&Trips$
 Endevent

行	&Trips	&TotalTrips
3	2	2
4	2	4
5	1	5

グリッドコントロール外の変数に読み込んだ行の値に基づき、計算結果を表示したい場合、Load イベント内にこの計算式も追加することができます。
 ここでは、変数の値を加算する処理を実装しています。
 つまり、Load イベントでは、グリッド内の変数だけでなく、外部に配置した変数も変更することが出来ました。

再描画時の処理

国番号 (なし) ▼

カテゴリ番号 (開始) 0

カテゴリ番号 (終了) 0

観光名所番号	観光名所名	国名	観光名所写真	旅行数
18	スミソニアン博物館	アメリカ		0
17	サグラダ・ファミリア	スペイン		0
15	ルーブル美術館	フランス		2
16	エッフェル塔	フランス		2
19	マティス美術館	フランス		1
20	ノートルダム寺院	フランス		2
21	万里の長城	中国		3
22	紫禁城	中国		2

合計旅行数 12

国番号 中国 ▼

カテゴリ番号 (開始) 0

カテゴリ番号 (終了) 0

観光名所番号	観光名所名	国名	観光名所写真	旅行数
21	万里の長城	中国		3
22	紫禁城	中国		2

合計旅行数 17

Event Refresh
<コード>
Endevent

実行の結果、初期表示の時点では、正しく計算され、グリッド内の変数の合計値が表示されました。

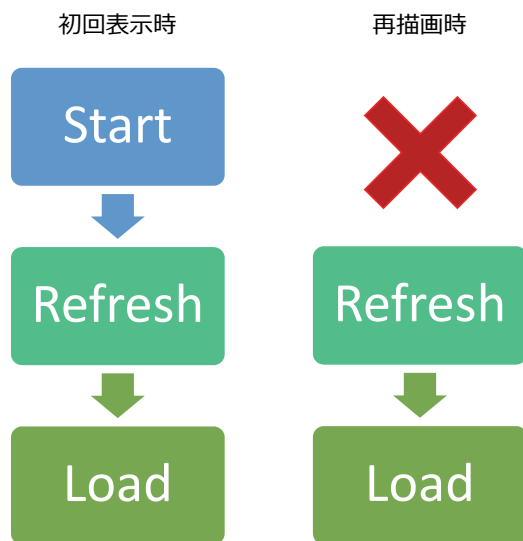
しかし、変数の値を変更し、絞り込みを行った場合、値は正しくありません。

この理由は、Load イベントが絞り込みを行った結果、再度実行されたためです。グリッド内の変数の場合、「代入」を行っているため、毎回適切な値のみが格納されますが、外部の変数に対しては、「加算」を行っていました。そのため、絞り込み後の再実行で、初期表示時の値を維持し、加算してしまったため、正しい値になりません。

グリッドコントロールの再読み込みに伴う再描画時に、グリッド外の変数を初期化します。このためには、システムイベントの 1 つである Refresh イベントを利用します。

Web パネルにおけるシステムイベント

システムイベント：既定のタイミングで自動で実行されるイベント



The screenshot shows the 'Events' tab in the GeneXus IDE for a project named 'WWAttractionsScratch'. The event list contains two events:
 1. 'Event Load' (lines 1-4):
 2 &Trips = Count(TripDate)
 3 &TotalTrips = &TotalTrips + &Trips
 4 Endevent
 2. 'Event Refresh' (lines 6-8):
 6 &TotalTrips = 0
 8 Endevent

Web パネルで利用可能なシステムイベントについて説明します。

システムイベントは、あらかじめ GeneXus として、生成されたアプリケーションにおいて、特定のタイミングで実行されるように用意されたイベントです。

Web パネルに用意されているシステムイベントのうち、画面描画に関連するイベントを説明します。

該当するイベントは、次の 3 つです。

- Start イベント
このオブジェクトに基づく画面をはじめて描画する際にのみ実行されるイベント
- Refresh イベント
画面の描画時に実行されるイベント。初回描画時も再描画時も実行されるイベント
- Load イベント
グリッドの行を出力する場合に実行されるイベント。出力する行 1 行ごとに 1 回実行されるイベント

このような実行タイミングから、Load イベントで利用している変数などが再描画のたびに初期化される必要がある場合、Refresh イベントで実装します。

画面の初期表示時のみで問題ない場合は、Start イベントを利用します。

また、各システムイベントの実行タイミングは、あらかじめ決められているため、[Events] エlementにおける記述順序は影響しません。

パターンが自動生成した WWAttraction

The screenshot displays the GeneXus IDE interface for the WWAttraction pattern. At the top, a table titled '観光名所s' (Attractions) lists data with columns for attraction ID, name, country, city, category, and image. Below this, a diagram shows the flow from the table to an 'Attraction' entity and then to a 'TrnMode' domain. The 'TrnMode' domain is defined as an enum with values: Insert, Update, Delete, and Display. A code editor shows a rule snippet for the 'AttractionId' property, using the 'TrnMode' enum to determine the action (insert, update, or delete).

観光名所番号	観光名所名	国番号	国名	都市番号	都市名	カテゴリ番号	カテゴリ名	観光名所写真	更新	削除
16	エッフェル塔	2	フランス	1	パリ	8	モニュメント		更新	削除
17	サグラダ・ファミリア	9	スペイン	1	バルセロナ	8	モニュメント		更新	削除
18	スミソニアン博物館	10	アメリカ	2	ワシントンD.C.	6	ミュージアム		更新	削除

```

1 /* Generated by Work With Pattern [Start] - Do not change */
2 [web]
3 {
4   parm(in &Mode, in:&AttractionId);
5
6   AttractionId = &AttractionId if not &AttractionId.IsEmpty();

```

ドメイン: TrnMode

Enum Values	Insert, Insert	INS; Update, Update	UPD; Delete, Delete	DLT; Display, Display	DSP

Web パネルにグリッドコントロールを配置し、データの一覧を表示する画面を実装してきました。

この画面は、Work With for Web パターンをトランザクションオブジェクトに適用し、自動生成された一覧画面と似ています。

実際に、トランザクションオブジェクトへパターンを適用し、自動生成される画面の多くは、Web パネルを利用した実装です。

パターンによって生成された一覧画面を確認すると、表示するデータの絞り込みだけでなく、データ操作を開始するためのアクションも含まれていました。

具体的には、データの新規登録、更新、削除です。

これらのアクションを実現するために、グリッド外にボタンコントロールが配置され、グリッド内に 2 つのコントロールが列として追加されています。

すべてのコントロールには、アクションとして、イベントが紐づけられ、イベント内でパターンを適用したトランザクションの呼び出しが実装されます。

呼び出しの処理では、必ず 1 つ目のパラメーターにモードを指定する変数が指定されます。

この変数の値によって、どのデータ操作（新規登録、更新、削除）を開始するために呼び出されたかを認識します。

それ以降に指定されるパラメーターは、トランザクションの主キー項目です。

更新や削除の場合、クリックした行の値が利用され、グリッド外にある新規登録の場合、空の値になります。

呼び出されるトランザクションオブジェクトにも値を受け取るための Parm ルールが定義されています。

モードの指定は、ナレッジベース作成時に自動で作成される TrnMode 列挙型ドメインの値に基づきます。

このドメインの [Enum Values] プロパティでは、4 つ値が用意され、新規登録、更新、削除に加え、参照というモードも用意されています。

更新アクションの追加

The screenshot displays the GeneXus IDE interface. On the left, the 'Web Layout' panel shows a form with the following elements:

- 国番号 (Country):
- カテゴリ番号 (開始) (Category Number (Start)):
- カテゴリ番号 (終了) (Category Number (End)):
- GRID (Table):

観光名所番号 (AttractionId)	観光名所名 (AttractionName)	国名 (CountryName)	観光名所写真 (Attraction Photo)	旅行数 (&Trips)	更新 (Update)
- 合計旅行数 (Total Trips):

On the right, the 'Events' panel shows the following code:

```

1  Event ブロック (Load)
5
6  Event ブロック (Refresh)
9
10 Event Start
11   &UpdateImage.FromImage(edit)
12 Endevent
13
14 Event &UpdateImage.Click
15   Attraction(TrnMode.Update, AttractionId)
16 Endevent
17
  
```

A blue arrow points from the 'edit' variable in the 'Images' panel to the '&UpdateImage.FromImage(edit)' action in the 'Event Start' event.

パターンによって生成された画面に実装されたアクションは、すべて Web パネルで手動で実装可能です。

例えば、更新のアクションを追加するため、Image タイプの変数をグリッドに追加することが可能です。

もし、この変数に表示する画像は、すべての行で統一した画像にする場合、画面を表示した際に 1 度だけ設定し、再読み込み時に変更する必要もありません。

そのため、Start イベントでの処理を実装することが最適です。

FromImage メソッドを利用することで、ナレッジベース内の画像オブジェクト名を指定し、変数に画像を設定できます。

また、更新のための画面を呼び出す処理についても様々な実装方法があります。

その中で、変数のコントロールをクリックした場合の実装をスライドでは行っています。

新たなイベントを定義し、イベントの名前には、<変数名>.Click と指定します。

画像をクリック時に実行する処理を記述できるため、トランザクションオブジェクトに必要なパラメーターとともに呼び出す実装を記述することで、更新のアクションが追加できます。

更新アクションの実行確認

国番号

カテゴリ番号 (開始)

カテゴリ番号 (終了)

観光名所番号	観光名所名	国名	観光名所写真	旅行数	更新
18	スミソニアン博物館	アメリカ		<input type="text" value="0"/>	
17	サグラダ・ファミリア	スペイン		<input type="text" value="0"/>	
15	ルーブル美術館	フランス		<input type="text" value="2"/>	
16	エッフェル塔	フランス		<input type="text" value="2"/>	
19	マティス美術館	フランス		<input type="text" value="1"/>	
20	ノートルダム寺院	フランス		<input type="text" value="2"/>	
21	万里の長城	中国		<input type="text" value="3"/>	
22	紫禁城	中国		<input type="text" value="2"/>	

合計旅行数 12

観光名所

観光名所番号 16

観光名所名

国番号

国名 フランス

都市番号

都市名 パリ

カテゴリ番号

カテゴリ名 有名なランドマーク

観光名所写真

更新アクションを追加した画面を確認すると、追加した画像はクリックすることができ、クリックした行に表示されているデータの更新画面が表示されます。

更新画面で「実行」をクリックすると、更新画面を呼び出した一覧画面へ自動で遷移します。

この挙動は、トランザクションに Work With for Web パターンが適用されていることにより、登録処理が完了した際に、呼び出し元の画面に戻るように Return コマンドの実行が実装されているためです。

一覧画面に戻ってきた場合、画面をはじめて表示する場合と同様の処理が行われます。つまり、システムイベントの Start、Refresh、Load イベントが順に実行されました。

ここまでに触れていませんでしたが、グリッド内に配置された変数は、更新アクションの実装まで、読み取り専用で表示されていました。

しかし、更新アクションが追加されたことで、変更可能なコントロールとして表示されました。

Webパネルの説明を開始した際に、配置される変数は入力可能で配置されると説明していましたが、実態としてグリッドコントロールの場合、異なります。

ある条件を満たした場合のみ既定で変更可能で実装され、そうでない場合、読み取り専用で実装されます。

変更可能となるケースの 1 つは、今回のような行単位のイベントが定義された場合です。

列の非表示

The screenshot displays the GeneXus IDE interface for a project named 'WWAttractionsScratch'. The 'Web Layout' tab is active, showing a form with several input fields and a grid. The grid is titled 'GRID' and contains columns for '観光名所番号' (AttractionId), '観光名所名' (AttractionName), '国名' (CountryName), '観光名所写真' (AttractionImage), and '旅行数' (&Trips). The 'AttractionId' column is highlighted with a green border, indicating it is the selected item for the properties window.

On the right, the 'プロパティ' (Properties) window is open, showing the '項目属性/変数: AttractionId' (Item Properties/Variable: AttractionId) section. The 'Visible' property is set to 'True', and the 'Invite Message' property is set to 'False'. The 'Event &UpdateImage.Click' section shows the code: `Attraction(TrnMode.Update, AttractionId)`.

グリッドコントロールの特定行で実行するアクション（イベント）で、項目属性の値を利用する場合、すべての項目属性は、列として存在している必要があります。この理由としては、アクションが実行されるタイミングで、新たにデータベースへの接続は行われません。

画面の描画（グリッドの読み込み）時に取得された値で処理が行われます。

そのため、もしも処理に必要な項目属性があり、画面には表示させたくないという場合、列として表示している項目属性の [Visible] プロパティを [False] に変更します。

GeneXus[™]