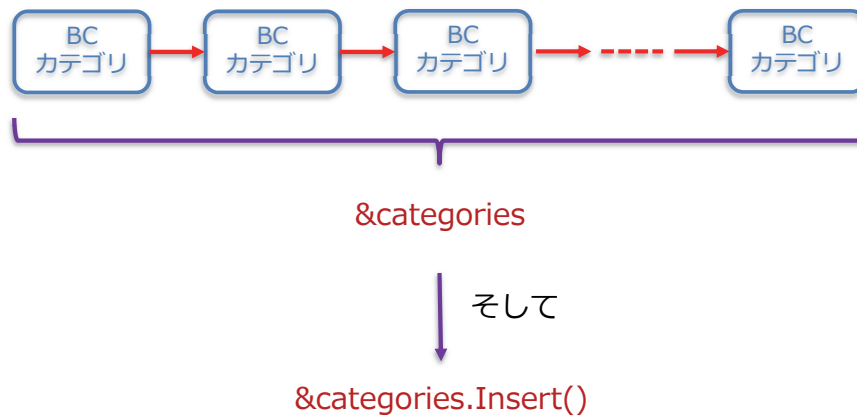


ビジネスコンポーネントと データプロバイダーを用いて データを一斉処理する方法

GeneXus[™]

目的: Category と Attraction テーブルを初期化してデータを入力する

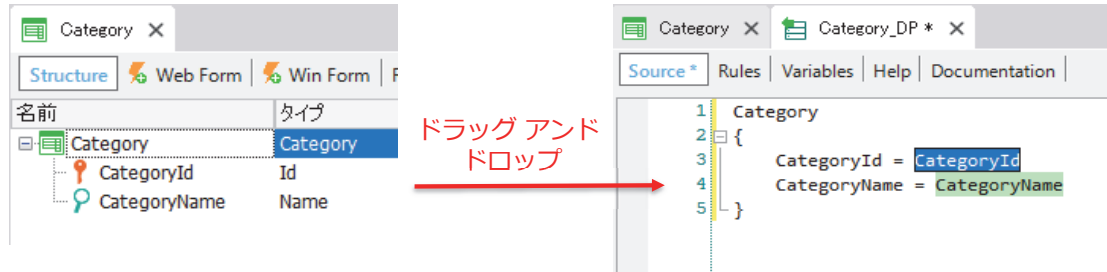
必要な作業:



データベースに追加するカテゴリアイテムを呼び出す、Category に対応するビジネスコンポーネントのコレクション変数を取得できる場合、必要なことは、このコレクション変数に Insert() メソッドを適用するだけです。これにより、前の章で説明した通り、このメソッドでコレクションに含まれるすべてのビジネスコンポーネントの挿入が可能になります。

この方法について考えてみましょう。

コレクションを取得する場合もデータプロバイダーを使用



既に説明したとおり、データプロバイダーによって、単純型とコレクション型両方の構造化データを返すことができます。今回の場合、Category コレクションを返すように設定する必要があります。ただし、今回の各カテゴリは、構造化データタイプではなく、**ビジネスコンポーネント**です。

ビジネスコンポーネントの構造は、SDT とまったく同じです。そのため、データプロバイダーによって、単純型とコレクション型両方のビジネスコンポーネントをロードし、返すことができます。

Category トランザクションをデータプロバイダーの [Source] にドラッグすると、トランザクション構造が記述されます。左辺には、戻り値に格納されるビジネスコンポーネントのメンバーが表示されます。メンバーは項目属性そのものではありませんが、項目属性と同じ名前になります。一方で、対応するテーブルの項目属性が右辺に表示されます。データプロバイダーは、戻り値の BC にロードするデータを取得します。

テーブルには多数のレコードが保持されているため、要件に応じて、データプロバイダーは、このビジネスコンポーネントのコレクションを返すようにする必要があります。

データプロバイダーによってコレクションを返すには [Collection] プロパティを使用

Data Provider: Category_DP	
Name	Category_DP
Description	Category_DP
Expose as Web Service	False
Module/Folder	Root Module
Qualified Name	Category_DP
Object Visibility	Public
Output	
Infer Structure	No
Output	Category
Collection	True
Collection Name	CategoryCollection
Network	

プロパティで、[Output] プロパティにビジネスコンポーネント値が入力されています。ただし、[Collection] プロパティは [True] には設定されておらず、[True] に設定する必要があります。

設定を変更すると、新しく [Collection Name] プロパティが表示されます。既定では、データプロバイダー名になります。それを「CategoryCollection」に変更します。

新しい値を手動で入力して割り当て

```
Category
{
    CategoryName = "美術館"
}
Category
{
    CategoryName = "モニュメント"
}
Category
{
    CategoryName = "有名なランドマーク"
}
```

=

```
CategoryCollection
{
    Category
    {
        CategoryName = "美術館"
    }
    Category
    {
        CategoryName = "モニュメント"
    }
    Category
    {
        CategoryName = "有名なランドマーク"
    }
}
```

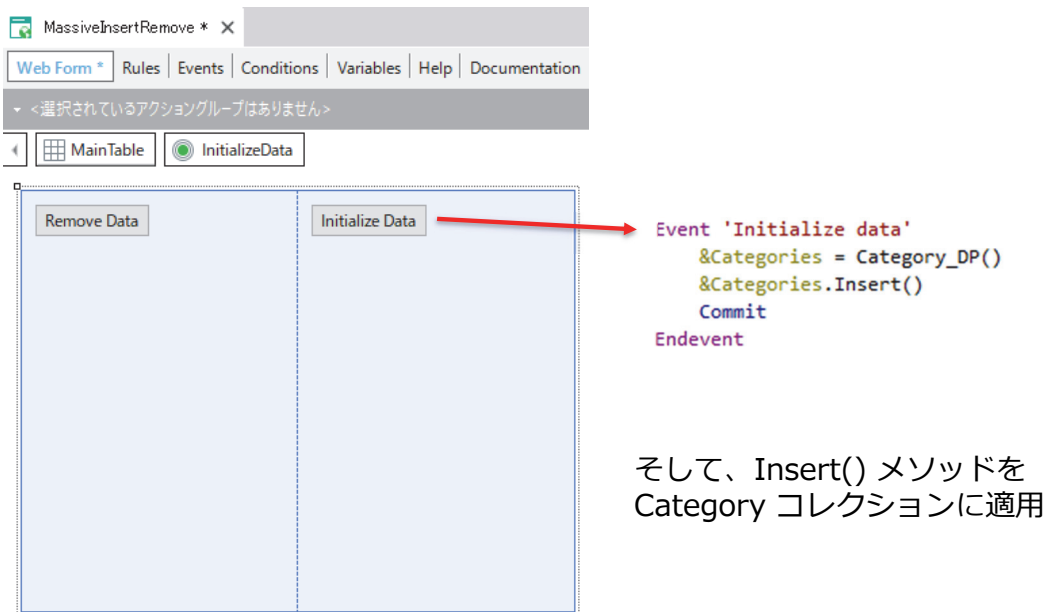
データプロバイダーの記述を明確化

なお、このコレクションには、データベースのデータをロードしません。新しい値を手動で入力して割り当てます。

そのため、コレクションアイテムに関連付けられるグループを 1 つずつ入力します。CategoryId 項目属性は自動採番されるため、レコードを挿入する際、値を割り当てる必要はありません。この点については次回扱いますので、ここでは、この代入式を削除します。

CategoryCollection という名のコレクションを返す必要があります。[Collection] プロパティを [True] に設定したため、データプロバイダーには、コレクションを返すよう求めていることが指定されていますが、コードを明確化することで、GeneXus の推論内容を明示することができます。そのためには、すべての Category グループを、CategoryCollection コレクションに対応する CategoryCollection グループ内に入れます。

Web パネルからデータプロバイダーを呼び出し



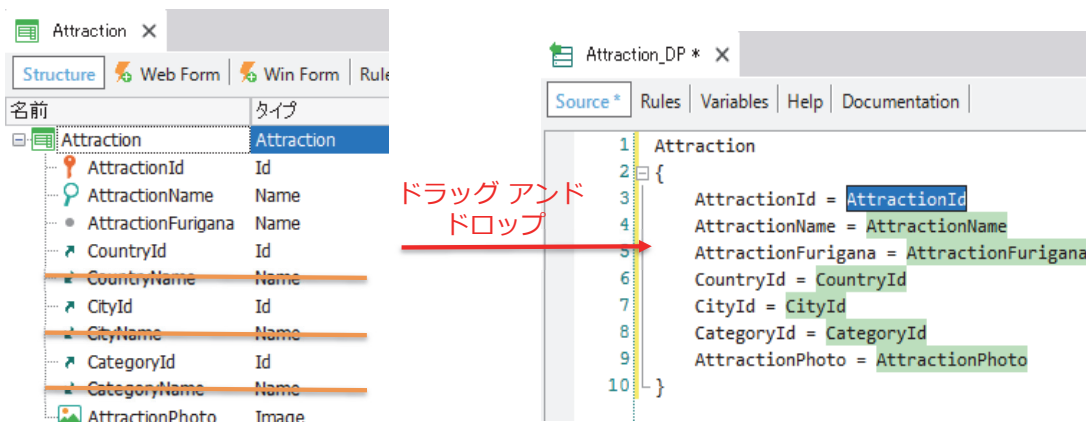
The screenshot shows the GeneXus IDE interface. At the top, there's a tab labeled 'Web Form *' and a menu bar with 'Rules', 'Events', 'Conditions', 'Variables', 'Help', and 'Documentation'. Below the menu bar, a message says '<選択されているアクショングループはありません>'. In the main workspace, there are two buttons: 'MainTable' and 'InitializeData'. A red arrow points from the 'InitializeData' button to a code snippet on the right. The code snippet is as follows:

```
Event 'Initialize data'  
  &Categories = Category_DP()  
  &Categories.Insert()  
  Commit  
Endevent
```

そして、Insert() メソッドを
Category コレクションに適用

ここで必要な作業は、このデータプロバイダーを、Web パネルのボタンに関連付けられたイベントから呼び出し、データベースに入力することだけです。

Attraction テーブルを初期化



次に、Attraction テーブルを初期化する必要があります。また、「Attraction_DP」という名のデータプロバイダーを作成します。

ビジネスコンポーネントをすでに取得したトランザクションからデータプロバイダーに項目属性をドラッグして、ビジネスコンポーネントの各メンバーのうち、物理テーブルの対応する項目属性によって、既定値が初期化されることを確認します。

ここでも、テーブル内に物理的に存在する項目属性だけが対象であり、トランザクションや式で推論された項目属性や式は対象ではありません。

新しい値を割り当て

```

Attraction
{
  AttractionName = "ルーブル美術館"
  AttractionFurigana = "るーぶるびじゅつかん"
  CountryId = 2
  CityId = 1
  CategoryId = 1
}
Attraction
{
  AttractionName = "万里の長城"
  AttractionFurigana = "ばんりのちやうじやう"
  CountryId = 3
  CityId = 1
  CategoryId = 2
}
Attraction
{
  AttractionName = "エッフェル塔"
  AttractionFurigana = "えっふえるとう"
  CountryId = 2
  CityId = 1
  CategoryId = 2
}

```

```

Attraction
{
  AttractionName = "ルーブル美術館"
  AttractionFurigana = "るーぶるびじゅつかん"
  CountryId = find(CountryId , CountryName="フランス")
  CityId = find(CityId , CityName="パリ")
  CategoryId = find(CategoryId , CategoryName="美術館")
}
Attraction
{
  AttractionName = "万里の長城"
  AttractionFurigana = "ばんりのちやうじやう"
  CountryId = find(CountryId , CountryName="中国")
  CityId = find(CityId , CityName="北京")
  CategoryId = find(CategoryId , CategoryName="モニュメント")
}
Attraction
{
  AttractionName = "エッフェル塔"
  AttractionFurigana = "えっふえるとう"
  CountryId = find(CountryId , CountryName="フランス")
  CityId = find(CityId , CityName="パリ")
  CategoryId = find(CategoryId , CategoryName="モニュメント")
}

```

ミスを防ぐため、Find() メソッドを使用

既存の観光名所をロードする必要はないため (このデータプロバイダーを実行して初期データをロードするため)、右辺にあるすべての項目属性を削除して、手動で右辺の値を入力します。また、ID は自動採番されるため、値を割り当てる必要はありません。観光名所の写真は、後で割り当てるため、この項目属性も削除します。

スライド左側では、覚えておいた CountryId、CityId、CategoryId の値を割り当てます。この値は、該当テーブルに存在しないことがあります。値が 1 つでも存在しないと、ビジネスコンポーネントでレコードを挿入しようとしたときに、該当する参照整合性チェックがトリガーされ、挿入に失敗します。

存在しない値を割り当てることがないようにするために、Find 式を使用して、国、都市、カテゴリの名前に基づく正しい ID を見つけます。

Find 式が行うのは、データベースにアクセスして、使用されている名前に対応する ID を検索することだけであり、ビジネスコンポーネントに割り当てられているほかの値は静的に指定されています。

[Collection] プロパティを [True] に設定

Data Provider: Attraction_DP

Name	Attraction_DP
Description	Attraction_DP
Expose as Web Service	False
Module/Folder	Root Module
Qualified Name	Attraction_DP
Object Visibility	Public
Output	
Infer Structure	No
Output	Attraction
Collection	True
Collection Name	AttractionCollection
Network	

```
AttractionCollection
{
  Attraction
  {
    AttractionName = "ルーブル美術館"
    AttractionFurigana = "ルーブルびじゅつかん"
    CountryId = find(CountryId , CountryName="フランス")
    CityId = find(CityId , CityName="パリ")
    CategoryId = find(CategoryId , CategoryName="美術館")
  }
  Attraction
  {
    AttractionName = "万里の長城"
    AttractionFurigana = "ばんりのちやうじやう"
    CountryId = find(CountryId , CountryName="中国")
    CityId = find(CityId , CityName="北京")
    CategoryId = find(CategoryId , CategoryName="モニュメント")
  }
  Attraction
  {
    AttractionName = "エッフェル塔"
    AttractionFurigana = "えっふえるとう"
    CountryId = find(CountryId , CountryName="フランス")
    CityId = find(CityId , CityName="パリ")
    CategoryId = find(CategoryId , CategoryName="モニュメント")
  }
}
```

構文を調整して、コレクションが返されるように指定

Category データプロバイダーの場合と同様、多数の観光名所が返されるように、[Collection] プロパティを [True] に設定します。また、[Source] 内の宣言を調整して、各グループが AttractionCollection グループに入るようにして、[AttractionCollection] にします。

観光名所の写真をロードするには

The screenshot shows the 'ImagesToolWindow' with a '新規画像' (New Image) button highlighted. A red arrow points from this button to the '新規画像ウィザード' (New Image Wizard) dialog box. The dialog box has two options: 'ファイルから画像を作成' (Create image from file) and '外部画像を作成' (Create external image). The '外部画像を作成' option is selected. Below the dialog, the 'AttractionCollection' data structure is shown in the 'Source' tab. The 'Attraction' object has properties like 'AttractionName', 'AttractionFurigana', 'CountryId', 'CityId', 'CategoryId', and 'AttractionPhoto'. The 'AttractionPhoto' property is highlighted with a red box, and a red arrow points from the 'AttractionPhoto' property to the 'AttractionPhoto' property in the 'Attraction' object. The text 'すべての画像を挿入' (Insert all images) is written next to the arrow.

新規画像ウィザード

新規画像ウィザードへようこそ

このウィザードは新しい画像オブジェクトの作成を手助けします。

☒ ファイルから画像を作成
ソース画像が PC またはネットワーク上に保存されているファイルの場合、このオプションを選択してください。

☐ 外部画像を作成
Web 上の画像を参照する場合、このオプションを選択してください (例: GeneXus Web サイトにある画像)

キャンセル < 戻る 次へ > 終了

すべての画像を挿入

AttractionCollection

```

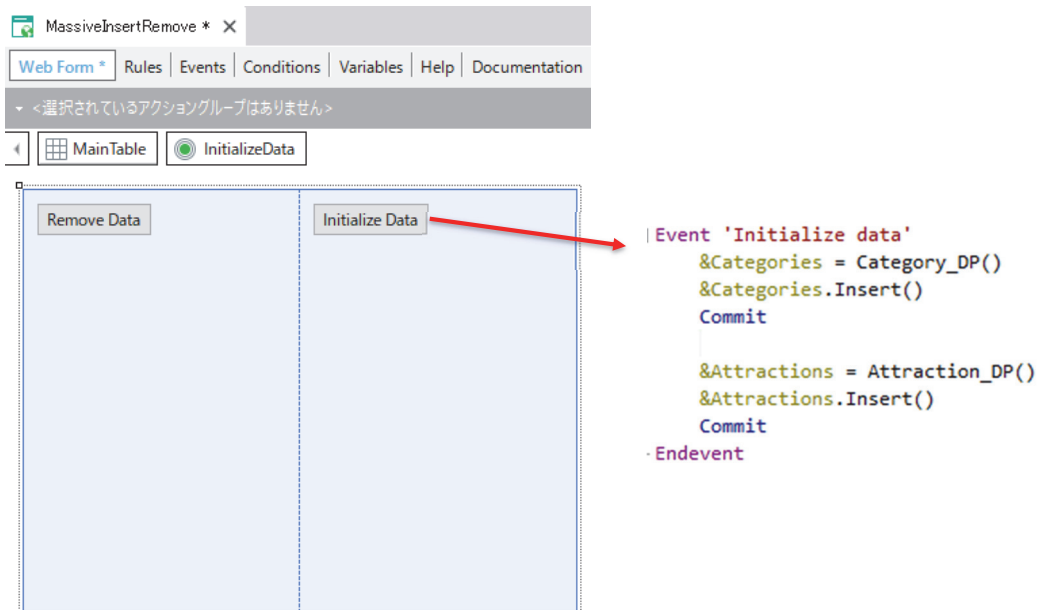
1 {
2   Attraction
3   {
4     AttractionName = "ルーブル美術館"
5     AttractionFurigana = "るーぶるびじゅつかん"
6     CountryId = find(CountryId, CountryName="フランス")
7     CityId = find(CityId, CityName="パリ")
8     CategoryId = find(CategoryId, CategoryName="美術館")
9     AttractionPhoto = louvre.Link()
10  }
11 }
12
13 {
14   Attraction
15   {
16     AttractionName = "万里の長城"
17     AttractionFurigana = "ばんりのちようじょう"
18     CountryId = find(CountryId, CountryName="中国")
19     CityId = find(CityId, CityName="北京")
20     CategoryId = find(CategoryId, CategoryName="モニュメント")
21     AttractionPhoto = GreatWall.Link()
22  }
23 }

```

観光名所の写真もロードするには、まず、KB 内の Image オブジェクトとして挿入します。

次に、各データ プロバイダー グループで、AttractionPhoto に画像の名前を割り当てます (.Link)。

Web パネルイベントにデータプロバイダーの呼び出しを追加



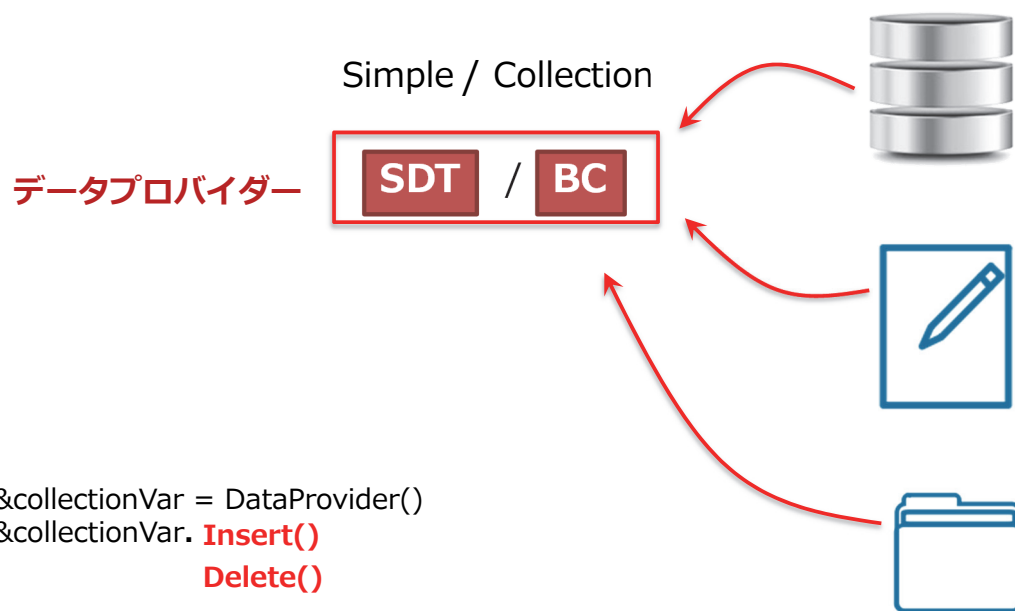
The screenshot shows the GeneXus IDE interface for a web form named 'MassiveInsertRemove'. The 'Events' tab is selected, and the 'InitializeData' event is highlighted. The event code is as follows:

```
Event 'Initialize data'  
  &Categories = Category_DP()  
  &Categories.Insert()  
  Commit  
  
  &Attractions = Attraction_DP()  
  &Attractions.Insert()  
  Commit  
Endevent
```

ここで必要な作業は、ロードしたコレクションを受け取るように、データプロバイダーを呼び出すだけです。

なお、観光名所を挿入するには、先にカテゴリを作成しておく必要があります。そのため、順序はイベントコードで使用しているものになります。

まとめ



データプロバイダーは、データベースのデータの構造をロードできるだけでなく、定数などの静的なデータの構造もロードできます。

ほかの外部ソースからもロードすることが可能ですが、この点については、あとの章で説明します。

また、データプロバイダーは、SDT の構造だけでなく、ビジネスコンポーネントの構造もロード可能で、単純型とコレクション型の両方に対応します。

最後にコレクション型の構造の場合、Insert() や Delete() など、単一の操作ですべてのコレクションアイテムに影響を与えるメソッドを適用できます。