

データベースの更新

ビジネスコンポーネントの使用

GeneXus[™]

ビジネスコンポーネントの有効化

名前	タイプ
Attraction	Attraction
AttractionId	Id
AttractionName	Name
CountryId	Id
CountryName	Name
CityId	Id
CityName	Name
CategoryId	Id
CategoryName	Name
AttractionPhoto	Image

プロパティ	
BusinessComponent: Attraction	
Name	Attraction
Description	観光名所
Module/Folder	Root Module
Business Component	True
Qualified Name	Attraction
Object Visibility	KnowledgeBase

Type Definition	
Based on	(none)
Data Type	Numeric
Length	
Decimals	
Business Components	Attraction
External Objects	

Attraction x ナビゲーション表示 x

フィルタ:

☒ Attraction
☒ Attraction_BC

Transaction Attraction_BC Navigation Report

Name: [Attraction_BC](#) Environment: [Default \(.NET\)](#)
 Description: 観光名所 Spec. Version: [18_0_10-18442](#)
 Form Class: [HTML](#)
 Program Name: [Attraction_BC](#)

LEVELS

Level Attraction

-Attraction ([AttractionId](#))
 -Country ([CountryId](#))
 -CountryCity ([CountryId](#), [CityId](#))
 -Category ([CategoryId](#))

INSERT INTO [Attraction](#) ([AttractionName](#), [AttractionPhoto](#),
[AttractionPhoto.Uri](#), [CountryId](#), [CityId](#), [CategoryId](#))

UPDATE [Attraction](#) ([AttractionName](#), [AttractionPhoto](#), [AttractionPhoto.Uri](#),
[CountryId](#), [CityId](#), [CategoryId](#))

DELETE FROM [Attraction](#)

Referential integrity controls on delete:

• [SupplierAttraction](#) ([AttractionId](#))

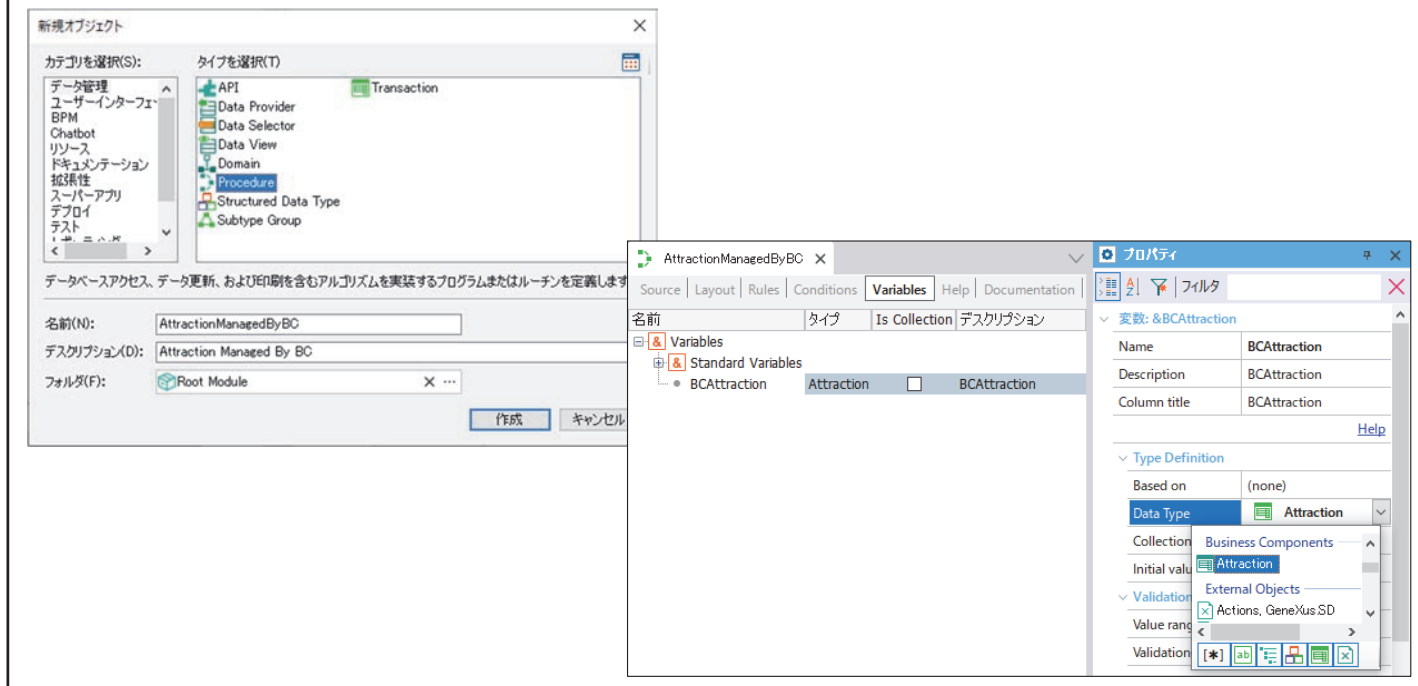
エラー: 0 警告: 0 成功: 2 すべて

トランザクションの [Business Component] プロパティを True に変更し、保存した場合、ナレッジベース内の変数で利用可能なデータタイプが追加されます。

変数の [Data Type] プロパティを参照すると、「Business Components」というグループ内に、プロパティを変更したトランザクションオブジェクトと同名のデータタイプが追加されていることが確認できます。

また、ナビゲーション表示では、トランザクションオブジェクト名末尾に「_BC」という文字列が追加された表示が増えます。
これは、ビジネスコンポーネントに対し、GeneXusが実施するデータ操作について確認できる表示です。

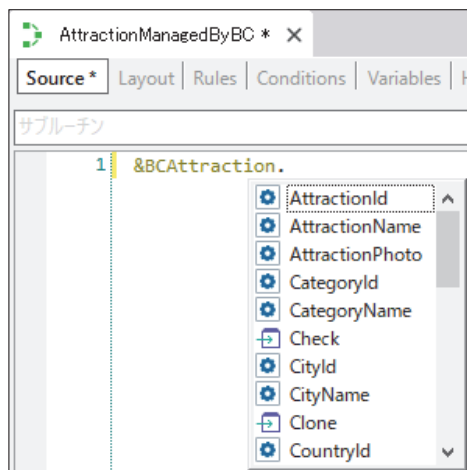
ビジネスコンポーネントを利用するオブジェクト



ビジネスコンポーネントは、変数のデータタイプとして利用するため、ナレッジベース内で定義可能な多くのオブジェクトで利用可能です。

この資料では、プロシージャオブジェクトを利用し、機能の利用を説明します。

ビジネスコンポーネントのメンバー



```
&BCAttraction.AttractionId =
&BCAttraction.AttractionName =
&BCAttraction.CountryId =
&BCAttraction.CountryName =
&BCAttraction.CityId =
&BCAttraction.CityName =
&BCAttraction.CategoryId =
&BCAttraction.CategoryName =
&BCAttraction.AttractionPhoto =
```

ビジネスコンポーネントタイプの変数は、SDT タイプの変数と同様に、構造データを扱うことが可能です。

変数名を記述し、ピリオドを入力して表示される候補には、値を格納するメンバーが表示されます。

この点も既に説明していますが、対象としたトランザクションで定義された項目属性名と同名のメンバーが用意されます。

もちろん、このメンバーが管理できるデータタイプも項目属性の定義に基づきます。

そして、このメンバーに値を代入し、メソッドを利用し、データの操作を行うことができます。

この点について引き続き説明を進めます。

ビジネスコンポーネントによる新規登録

AttractionManagedByBC * X

Source * | Layout | Rules | Conditions | Variables | Help | Documentation

サブルーチン

```

1 &BCAttraction.AttractionName = "紫禁城"
2 &BCAttraction.CountryId = Find(CountryId, CountryName = "中国")
3 &BCAttraction.CityId = Find(CityId, CityName = "北京")
4 &BCAttraction.CategoryId = Find(CategoryId, CategoryName = "遺跡")
5
6 &BCAttraction.|

```

Insert

InsertOrUpdate

Load

Mode

Save

SetValue

Success

ToJson

ToXml

Update

Insert(): Boolean

AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	ルーブル美術館	2	1	1	...
2	万里の長城	3	1	2	...
3	エッフェル塔	2	1	3	...
4	サグラダ・ファミリア	9	1	3	...
5	スミソニアン博物館	10	2	1	...
6	マティス美術館	2	2	1	...
0	紫禁城	3	1	2	...

&BCAttraction.Insert()

&BCAttraction	
AttractionId	0
AttractionName	紫禁城
CountryId	3
CountryName	
CityId	1
CityName	
CategoryId	2
CategoryName	
AttractionPhoto	

Attraction	
AttractionId	8
AttractionName	紫禁城
CountryId	3
CountryName	中国
CityId	1
CityName	北京
CategoryId	2
CategoryName	遺跡
AttractionPhoto	

ビジネスコンポーネントタイプの変数で、データを登録するためには、メンバーに登録する値を代入します。

必須となる項目すべてに値の代入が完了したら、Insert メソッドの実行を記述します。このメソッドが実行されると、変数内に格納された値をテーブルに登録するための処理が実行されます。

ただし、この処理は単純な登録処理ではなく、トランザクションオブジェクトから生成された画面から登録する場合と同じように、ルールの実行や参照整合性のチェックが行われたうえで、登録処理が実施されます。

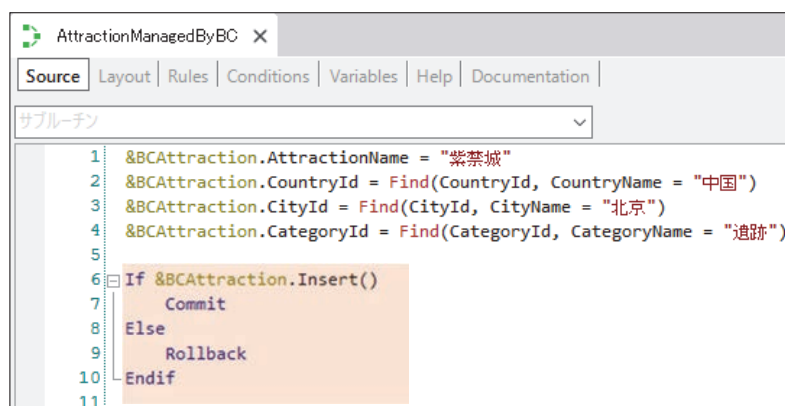
Insert メソッドは、Boolean 型の戻り値が設定されていて、登録が成功した場合には、True、失敗した場合には、False を返します。

また、成功した場合には、構造データ内の自動採番項目や、外部キーの従属項目の値も格納された状態となります。

登録したい値に外部キーがある場合、該当しない値を指定すると、上記の通り、参照整合性のチェックでエラーとなります。

もし、何らかの条件で該当のキーの値を想定できる場合、ローカル式の集計関数「Find」の利用を推奨します。

ビジネスコンポーネントによる新規登録の確定



ビジネスコンポーネントタイプによるデータの新規登録、更新、削除では、データベースの操作を確定する Commit が自動的に発行されません。

そのため、この機能を利用する場合、処理の最後にコミットが行われるように実装する必要があります。

GeneXus には、そのための Commit コマンドが用意されています。

このコマンドによる処理が実行された場合、それまでのデータ操作内容が確定します。

もし、前回のコミット以降に実施されたデータ操作内容を破棄したい場合には、Rollback コマンドが用意されています。

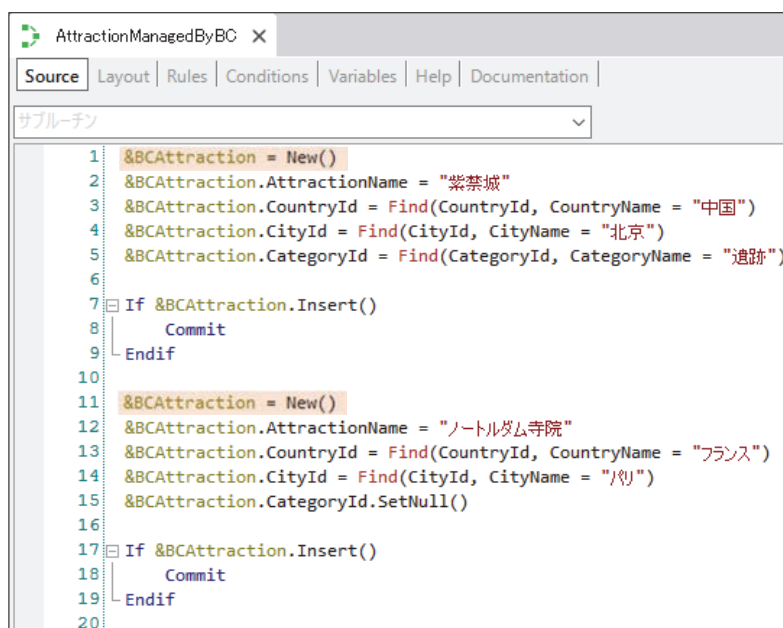
直前の処理の成否によって処理を分岐したい場合、If コマンドが利用できます。

「If」と記述し、スペースの後に条件を記述することで、条件を満たす (True) 場合の処理を続けて記述します。

もし、満たさない (False) 場合の処理を記述する必要がある場合、「Else」を記述し、満たさない場合の処理を記述します。

If コマンドは、「Endif」で閉じる必要があります。

ビジネスコンポーネントによる連続登録①



New コマンドを利用しない場合

AttractionId	0
AttractionName	紫禁城
CountryId	3
CountryName	
CityId	1
CityName	
CategoryId	2
CategoryName	
AttractionPhoto	



AttractionId	9
AttractionName	紫禁城
CountryId	3
CountryName	中国
CityId	1
CityName	北京
CategoryId	2
CategoryName	遺跡
AttractionPhoto	



AttractionId	9
AttractionName	ノートルダム寺院
CountryId	2
CountryName	中国
CityId	1
CityName	北京
CategoryId	NULL
CategoryName	遺跡
AttractionPhoto	

ビジネスコンポーネントタイプの変数を利用し、連続で複数のデータ操作を行うことが可能です。

ただし、この場合に重要となる点が、New 演算子の利用です。

ビジネスコンポーネントタイプで扱うことができる値は、SDT タイプと同様に、1 つの変数内に複数ある状態です。

また、データ操作が完了した場合、変数内には、対象としたデータの情報が格納されています。

これを破棄せずに処理が進められた場合、一部の値は、直前の操作で格納された値のままとなり、意図した挙動になりません。

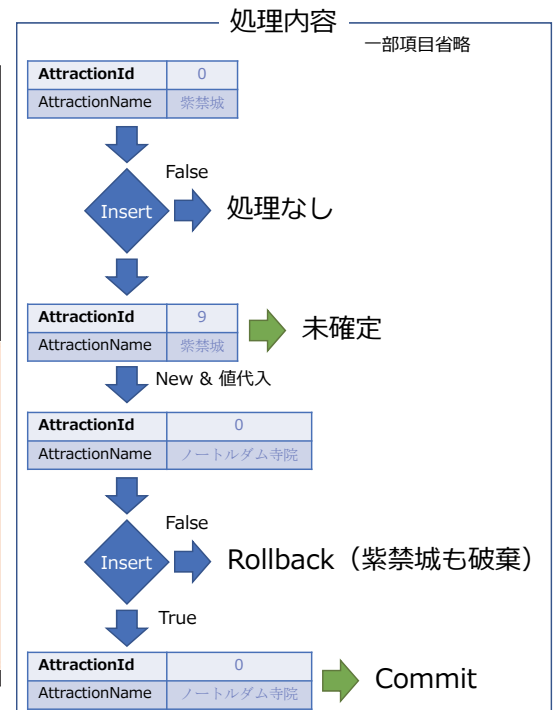
そのため、再利用の際には、New 演算子を利用し、新しいメモリ領域を確保することを推奨します。

ビジネスコンポーネントによる連続登録②

```

AttractionManagedByBC
Source | Layout | Rules | Conditions | Variables | Help | Documentation
サブルーチン
1  &BCAttraction = New()
2  &BCAttraction.AttractionName = "紫禁城"
3  &BCAttraction.CountryId = Find(CountryId, CountryName = "中国")
4  &BCAttraction.CityId = Find(CityId, CityName = "北京")
5  &BCAttraction.CategoryId = Find(CategoryId, CategoryName = "遺跡")
6
7  If &BCAttraction.Insert()
8      &BCAttraction = New()
9      &BCAttraction.AttractionName = "ノートルダム寺院"
10     &BCAttraction.CountryId = Find(CountryId, CountryName = "フランス")
11     &BCAttraction.CityId = Find(CityId, CityName = "パリ")
12     &BCAttraction.CategoryId.SetNull()
13
14     If &BCAttraction.Insert()
15         Commit
16     Else
17         Rollback
18     Endif
19 Endif
20

```



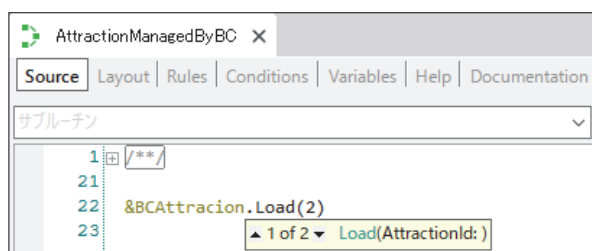
もし、ビジネスコンポーネントタイプの変数を利用し、連続して登録するデータで、双方が成功しなければ、データを確定したくないという実装も可能です。

その場合、1 件目のデータ操作の結果で、処理を分岐し、成功した場合の処理として、2 件目のデータ操作を記述します。
2 件目のデータ操作に成功した場合には、Commit コマンドを実行し、失敗した場合には、Rollback コマンドを実行します。

この実装により、1 件目が失敗した場合、後続の処理は実施されず、1 件目も失敗しているため、データ操作は一切行われません。

1 件目が成功し、2 件目が失敗した場合、Rollback コマンドにより、まだ確定されていない 1 件目のデータ操作も破棄されます。

ビジネスコンポーネントによる更新の開始



AttractionId	2
AttractionName	
CountryId	
CountryName	
CityId	
CityName	
CategoryId	
CategoryName	
AttractionPhoto	



AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	ルーブル美術館	2	1	1	...
2	万里の長城	3	1	2	...
3	エッフェル塔	2	1	3	...
4	サグラダ・ファミリア	9	1	3	...
5	スミソニアン博物館	10	2	1	...
6	マティス美術館	2	2	1	...
10	ノートルダム寺院	2	1	NULL	...



AttractionId	2
AttractionName	万里の長城
CountryId	3
CountryName	中国
CityId	1
CityName	北京
CategoryId	2
CategoryName	遺跡
AttractionPhoto	<画像データ>

ビジネスコンポーネントタイプの変数で、データを更新するためには、更新対象となるデータをロード（読み込み）する必要があります。

このためには、Load メソッドを利用します。

このメソッドには、引数が必要となり、その値は、更新対象となる主キーを特定できるものを指定します。

入力値のヒントが表示されるツールチップ内には、どの項目の値を指定するかが表示されます。

また、もしも対象のトランザクションが複合主キーで定義されている場合、カンマ区切りで指定する必要があります。

指定された値のデータが読み込まれると、変数内のすべてのメンバーに値が格納されます。この挙動は、トランザクションオブジェクトによって生成された画面で、主キーのフィールドに値を入力し、フォーカスを移動した場合と同様です。

ビジネスコンポーネントによる更新

AttractionManagedByBC

Source | Layout | Rules | Conditions | Variables | Help | Documentation

サブルーチン

```

1 田 /**/
21
22 &BCAttraction.Load(2)
23 &BCAttraction.AttractionName = "長城"
24 &BCAttraction.CategoryId = Find(CategoryId, CategoryName = "モニュメント")
25
26 If &BCAttraction.Update()
27     Commit
28 Endif
29

```

AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	ルーブル美術館	2	1	1	...
2	長城	3	1	3	...
3	エッフェル塔	2	1	3	...
4	サグラダ・ファミリア	9	1	3	...
5	スミソニアン博物館	10	2	1	...
6	マティス美術館	2	2	1	...
10	ノートルダム寺院	2	1	NULL	...

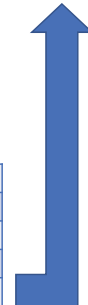
AttractionId	2
AttractionName	
CountryId	
CountryName	
CityId	
CityName	
CategoryId	
CategoryName	
AttractionPhoto	



AttractionId	2
AttractionName	万里の長城
CountryId	3
CountryName	中国
CityId	1
CityName	北京
CategoryId	2
CategoryName	遺跡
AttractionPhoto	<画像データ>



AttractionId	2
AttractionName	長城
CountryId	3
CountryName	中国
CityId	1
CityName	北京
CategoryId	3
CategoryName	遺跡
AttractionPhoto	<画像データ>



AttractionId	2
AttractionName	長城
CountryId	3
CountryName	中国
CityId	1
CityName	北京
CategoryId	3
CategoryName	モニュメント
AttractionPhoto	<画像データ>



Load コマンドで読み込んだデータを更新する場合、直接値を割り当てることで、データの変更を行います。

そして、テーブルへの更新を行うためには、Update メソッドを利用します。

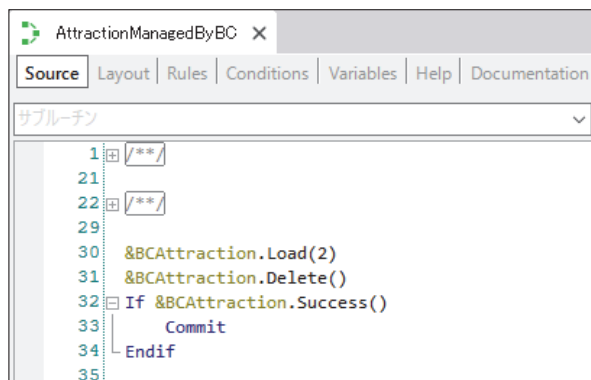
このメソッドも新規登録の Insert メソッド同様に、トランザクションオブジェクトから生成された画面から更新する場合と同じように、ルールの実行や参照整合性のチェックが行われたうえで、データの更新が行われます。

更新が完了すると、そのレコードの値に基づき、変数内の値も最新の状態となるように再読み込みが行われます。

また、前述の通り、更新時も自動でコミットは行われないため、成否判定の結果、明確に Commit コマンドを実行するように実装が必要です。

Update メソッドも、Boolean 型の戻り値が設定されていて、更新が成功した場合には、True、失敗した場合には、False を返すため、条件分岐に利用できます。

ビジネスコンポーネントによる削除



AttractionId	2
AttractionName	長城
CountryId	3
CountryName	中国
CityId	1
CityName	北京
CategoryId	3
CategoryName	モニュメント
AttractionPhoto	<画像データ>



AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	ルーブル美術館	2	1	1	...
2	長城	3	1	3	...
3	エッフェル塔	2	1	3	...
4	サグラダ・ファミリア	9	1	3	...
5	スミソニアン博物館	10	2	1	...
6	マティス美術館	2	2	1	...
10	ノートルダム寺院	2	1	NULL	...

ビジネスコンポーネントタイプの変数で、データを削除する場合も、対象となるデータをロード（読み込み）する必要があります。

この点は、トランザクションオブジェクトによって生成された画面で、データを削除する場合も同様でした。

削除を実行するためには、Delete メソッドを利用します。

このメソッドもトランザクションオブジェクトから生成された画面から削除する場合と同じように、ルールの実行や参照整合性のチェックが行われたうえで、データの削除が行われます。

Delete メソッドの場合、実行結果が Boolean 型で返されません。

そのため、ビジネスコンポーネントタイプで利用可能なすべてのメソッドの成否を参照することができる Success メソッドおよび Fail メソッドを利用する必要があります。Success メソッドは、直前のメソッドの実行が成功した場合に True、失敗した場合に False を返します。

Fail メソッドは、逆に失敗した場合に True、成功した場合に False を返します。

もし、削除が成功していた場合、Commit コマンドを実行することで、削除が確定されます。

エラーメッセージの取得

エラー発生ケース

トランザクションオブジェクトから生成された画面からデータ操作を行った場合、ルールや参照整合性のチェックによって、エラーが発生した場合、画面上にメッセージが表示されました。

同様の状況がビジネスコンポーネントタイプの変数による登録でも発生した場合、どうなるでしょうか。

ここまでで説明した実装では、成否判定によって、失敗した場合に登録を破棄することは実装していましたが、なぜ失敗したかを表示していません。

このようなエラーメッセージを取得する方法も用意されているため、その実装方法について説明を続けます。

エラーメッセージの取得

メッセージのデータタイプ

The screenshot displays the KB Explorer window on the left, showing the project structure. The 'Messages' object is highlighted under the 'Common' module. On the right, the 'Messages' structure window is open, showing a collection of 'Message' objects. The 'Message' object has three properties: 'Id' (VarChar(128)), 'Type' (MessageTypes, GeneXus), and 'Description' (VarChar(256)). A 'Values' window is also shown, indicating the status: Warning:0, Error:1, Info:2, Debug:3.

ビジネスコンポーネントから出力されるエラーメッセージは、あらかじめ GeneXus 上で定義されています。

[KB エクスプローラー] ウィンドウの「参照モジュール」ノードに含まれる「GeneXus」モジュール配下の「Common」モジュールに含まれる SDT オブジェクト「Messages」に基づき出力されます。

「参照モジュール」ノード内のモジュールやオブジェクトは、新規ナレッジベースを作成した際に必ず含まれ、GeneXus があらかじめ持つ機能を利用するためのものとなります。

また、このノード内のオブジェクトは、すべて [読み取り専用] となり、変更することはできません。

SDT オブジェクト Messages は、メイン構造がコレクションとなり、アイテムとなるサブ構造の名前は、Message です。

Message サブ構造内には、Id、Type、Description の 3 つのメンバーが含まれています。Id は、トランザクションオブジェクトのロジックに基づくエラーの場合、例外名が格納されます。

また、開発者によって定義されたメッセージの場合、指定された文字列がある場合、格納されます。

Type は、発生したメッセージの種類を示し、格納される値は、列挙型ドメインに基づき、番号で「0: 警告」、「1: エラー」、「2: 情報」、「3: デバッグ」を意味します。

Description に、実際に発生したメッセージが格納されます。

エラーメッセージの取得

取得と出力

```
&BCAttraction = New()
&BCAttraction.AttractionName = "エッフェル塔"
&BCAttraction.CountryId = Find(CountryId, CountryName = "フランス")
&BCAttraction.CityId = Find(CityId, CityName = "パリ")
&BCAttraction.CategoryId = Find(CategoryId, CategoryName = "モニュメント")
```

```
If &BCAttraction.Insert()
    Commit
Else
    &Messages = &BCAttraction.GetMessages()
    For &Message in &Messages
        Print PBMesage
    Endfor
Endif
```

名前	タイプ	Is Collection	デスクリプション
Variables			
Standard Variables			
• BCAttraction	Attraction	<input type="checkbox"/>	BCAttraction
• Messages	Messages, GeneXus.Common	<input type="checkbox"/>	Messages
• Message	Messages.Message, GeneXus.Common	<input type="checkbox"/>	Message

Source	Layout	Rules	Conditions	Variables	Help	Documentation
PBMessage						
Id	&Message.Id					
Type	&M					
Description	&Message.Description					

Values Warning:0,Error:1,Info:2,Debug:3

Id	
Type	1
Description	観光名所名は既に存在しています。

ビジネスコンポーネントのメソッドを利用した場合、そのメソッドによって、メッセージが生成される場合、GetMessages メソッドを利用することで、メッセージのコレクションを取得できます。

メソッドの戻り値は、直前に説明した Messages タイプです。

変数を定義し、戻り値を格納し、コレクション内のアイテムを参照することで、出力されたメッセージを確認できます。

プロシージャオブジェクトを利用し、実装していたため、PDF を出力するように定義を変更し、プリントブロックのデザインを行うことで、エラーが発生した際のメッセージを確認できるように実装が可能です。

エラーメッセージの取得

Id メンバーの値

```

Attraction X
Structure Web Layout Rules Events Variables Help Documentation Patterns
1 /* Generated by Work With Pattern [Start] - Do not change */
2 [web]
3 Code ブロック
16 /* Generated by Work With Pattern [End] - Do not change */
17
18 Error("観光名所名は必須です。", EmptyValue) If AttractionName.IsEmpty();
19 Msg("観光名所画像が未指定です。", EmptyValue) If AttractionPhoto.IsEmpty();
20

&BCAttraction = New()
//&BCAttraction.AttractionName = "エッフェル塔"
//&BCAttraction.CountryId = Find(CountryId, CountryName = "フランス")
&BCAttraction.CountryId = 22
&BCAttraction.CityId = Find(CityId, CityName = "パリ")
&BCAttraction.CategoryId = Find(CategoryId, CategoryName = "モニュメント")

If &BCAttraction.Insert()
    Commit
Else
    &Messages = &BCAttraction.GetMessages()
    For &Message in &Messages
        Print PMessage
    Endfor
Endif

```

Id	EmptyValue
Type	1
Description	観光名所名は必須です。
Id	ForeignKeyNotFound
Type	1
Description	'国'の該当レコードはありません。
Id	ForeignKeyNotFound
Type	1
Description	'都市'の該当レコードはありません。
Id	EmptyValue
Type	0
Description	観光名所画像が未指定です。

Id メンバーに格納される文字列について少し補足します。

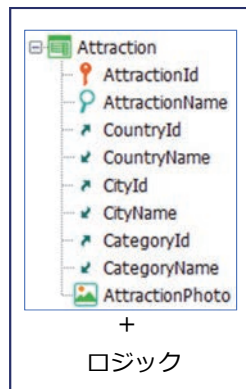
トランザクションのロジックに基づくエラーの場合、事前に決められた例外名が格納されます。

開発者による Error ルールや、Msg ルールの場合、ルールの第 2 引数に指定した文字列が格納されます。

ただし、指定できる文字列に注意が必要です。

第 2 引数に指定できる文字列は、シングルバイトの連続した文字列となり、メッセージとして表示したり、代入する文字列とは異なり、「"」（または「'」）でくくる必要がありません。

ビジネスコンポーネントの利用補足



&BCAttraction

AttractionId	7
AttractionName	紫禁城
CountryId	3
CountryName	中国
CityId	1
CityName	北京
CategoryId	2
CategoryName	遺跡
AttractionPhoto	<画像データ>

+

```

&BCAttraction.Load(Pk)
&BCAttraction.Insert()
&BCAttraction.Update()
&BCAttraction.Delete()

&BCAttraction.Success()
&BCAttraction.Fail()
&BCAttraction.GetMessages()

&BCAttraction.Mode()
&BCAttraction.Save()
&BCAttraction.InsertOrUpdate()
  
```

ビジネスコンポーネントタイプの変数を利用するためには、対象のトランザクションオブジェクトで [Business Component] プロパティを True に設定する必要があります。

この結果、トランザクションのロジックを含むデータ操作が可能なデータタイプが利用可能となりました。

また、[Rules] エlementや、[Events] Elementに記載された内容もビジネスコンポーネントには含まれます。

ただし、すべてがビジネスコンポーネントタイプの変数を通したデータ操作で利用されるものではありません。

インターフェースに関する定義は、ビジネスコンポーネントには、含まれません。

例えば、Parm ルールなどです。

また、ここまでに説明していないその他のメソッドも用意されていました。

これらについて簡単に説明します。

Mode メソッドは、ビジネスコンポーネントタイプの変数を通して、新規登録、更新、削除のどのデータ操作を行おうとしているまたは、行ったかを取得できるメソッドです。

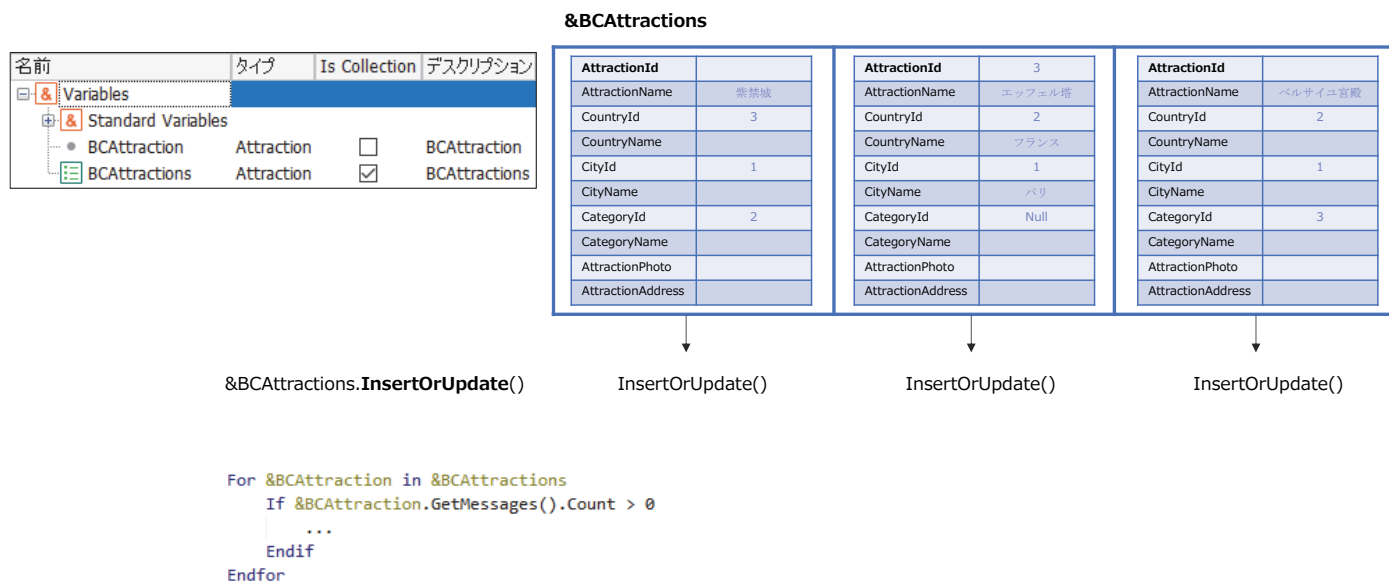
Save メソッドは、ビジネスコンポーネントタイプの変数を利用するモードに基づき、データの操作を新規登録するか、更新するかを決定し、データ操作を行います。

InsertOrUpdateメソッドは、新規登録をまず実施し、主キーの重複が無ければ、新規登録を行い、もし主キーの重複があった場合、既存レコードの更新を行います。

ビジネスコンポーネントタイプの変数は、コレクション型とすることもできます。

この時、Insert、Update、Delete、InsertOrUpdate メソッドがコレクション型のまま利用可能です。

ビジネスコンポーネントのコレクション



コレクション型のビジネスコンポーネントタイプの変数を定義した場合、これまでに説明したコレクション型同様アイテムは、同一データタイプのアイテム型となります。

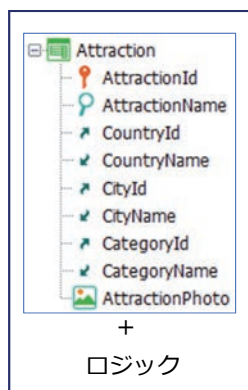
コレクション内に複数のアイテムが格納されている場合、任意のメソッドを利用すると、格納されたすべてのアイテムに対し、メソッドを実行するように指示することと、同意です。

ただし、コレクションに対し、メソッドを実行した場合、格納されたアイテムすべてが成功した場合にのみ、戻り値が True となります。
コレクション内のいずれかのアイテムで False であった場合、結果は False となります。

もし、失敗したアイテムの理由を確認したい場合、For in コマンドを利用し、格納されたアイテムを 1 つずつ参照し、GetMessages メソッドの内容を確認する必要があります。

ビジネスコンポーネントを利用可能なオブジェクト

プロシージャーのみか？



&BCAttraction

AttractionId	7
AttractionName	紫雲城
CountryId	3
CountryName	中国
CityId	1
CityName	北京
CategoryId	2
CategoryName	遺跡
AttractionPhoto	<画像データ>

+

&BCAttraction.**Load**(Pk)
 &BCAttraction.**Insert**()
 &BCAttraction.**Update**()
 &BCAttraction.**Delete**()

 &BCAttraction.**Success**()
 &BCAttraction.**Fail**()
 &BCAttraction.**GetMessages**()

 &BCAttraction.**Mode**()
 &BCAttraction.**Save**()
 &BCAttraction.**InsertOrUpdate**()

本資料では、プロシージャーオブジェクトで、ビジネスコンポーネントタイプの変数を利用し、データ操作の実装を行いました。

GeneXus において、ビジネスコンポーネントタイプの変数を利用したデータ操作は、手続き型のコードが記述可能なオブジェクトで利用可能です。

そのため、Web パネルオブジェクトや、トランザクションオブジェクトなどの [Events] エLEMENTで利用することも可能です。

ただし、このコースで詳細は取り扱いませんが、トランザクションオブジェクトで利用する場合には、いくつかの制限事項があります。

*GeneXus*TM

training.genexus.com
wiki.genexus.com