

オブジェクト間の呼び出し

基礎

GeneXus™

定数を使用したフィルタ

```

AttractionList X
Source | Layout | Rules | Conditions | Variables | Help | Documentation
サブルーチン
1 Print PBTitle
2 Print PBColumnTitles
3
4 For each Attraction
5   Order CountryName, AttractionName
6   Where CountryId = 2
7   Print PBAttractions
8 Endfor
9

```

```

AttractionList X
Source | Layout | Rules | Conditions | Variables | Help | Documentation
サブルーチン
1 Print PBTitle
2 Print PBColumnTitles
3
4 For each Attraction
5   Order CountryName, AttractionName
6   Where CountryName = "フランス"
7   Print PBAttractions
8 Endfor
9

```

一般的なリストを作成してフィルタに使用する国を「受け取る」には？

ここまでで説明してきた中では、パラメーターとともにオブジェクトを呼び出すべきと判断できるようなケースもありました。

例えば、PDF 出力を行うプロシージャオブジェクトで、データの絞り込みが必要となる場合、ここまでで説明した内容では、直接コード内で Where 節に定数を使用し、絞り込みを行っていました。

しかし、もしこの絞り込みのための値をユーザーが変更したいという場合、そのたびに開発者がこのコードを修正する必要があります。

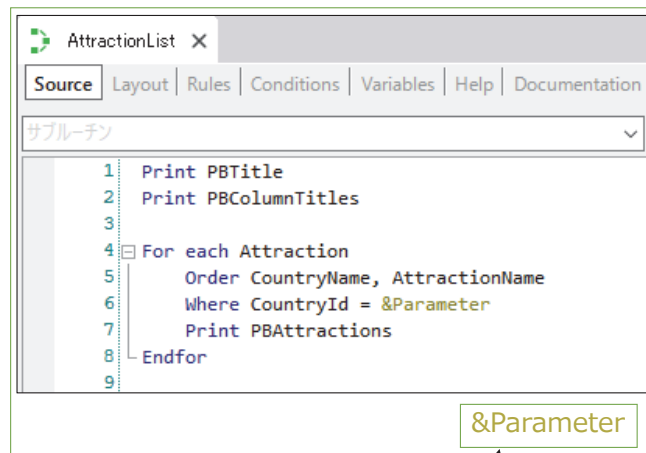
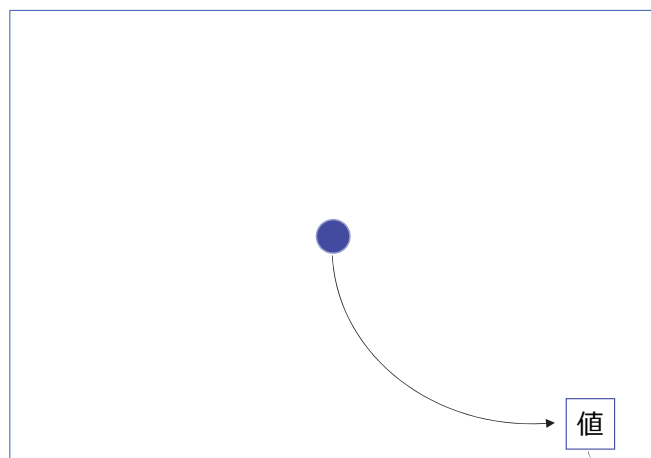
これは、稼働したアプリケーションとしては、最適な実装にはなりません。

これを解消するための方法について、この資料で説明を進めます。

想定される最適解

オブジェクト A

オブジェクト B
(AttractionsList)



それでは、このような要望に対応するために想定できる最適解について考えてみます。

この場合、プロシージャオブジェクトは、実行することで PDF が出力されるため、絞り込み用の値を用意することはできません。

そのため、別のオブジェクトを用意します。仮に「オブジェクト A」と呼びます。

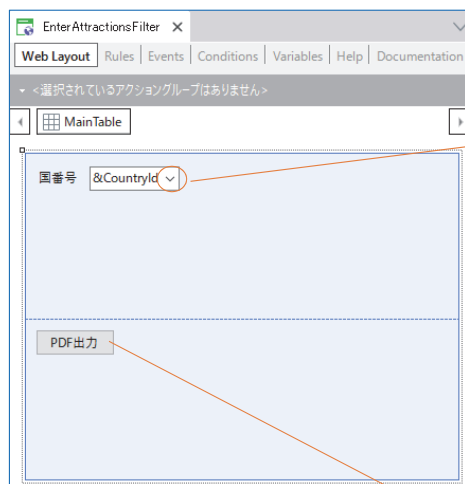
このオブジェクト A は、アプリケーションとしてのユーザーインターフェースがあり、値の入力や選択が可能なオブジェクトとして定義します。

オブジェクト A からは、PDF を出力するプロシージャオブジェクト（以降、「オブジェクト B」と記載）を呼び出します。

そして、この呼び出し時にオブジェクト A からオブジェクト B に値を受け渡します。

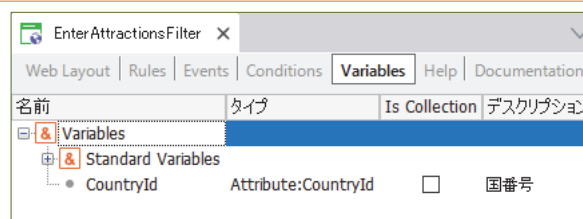
この受け渡された値をオブジェクト B においてフィルタに再利用することが最適解となります。

呼び出し画面の定義



1) 対象の国を指定するコントロールの追加

ダイナミック コンボ ボックス タイプのコントロールを使用した変数で、データベースにある国を選択可能とする。



2) AttractionsList プロシーチャーを呼び出すボタンコントロールの追加

[Events] エLEMENT :

```
Event 'CallAttractionList'
  AttractionList(&CountryId)
Endevent
```

CallAttractionList イベントが関連付けられたボタンコントロール

Web Layout で指定された国を保持する変数

ユーザーインターフェースがあり、値の入力や、別のオブジェクトを呼び出す機能を実装できるオブジェクトとして、「Web パネル」タイプのオブジェクトを利用します。このオブジェクトについての詳細は、別途説明するため、この時点では、様々な用途に活用できる画面を持つオブジェクトと認識してください。

画面上でユーザーが値を入力できるようにするためには、変数の定義が必要です。プロシーチャーオブジェクトと同様に [Variables] エLEMENTで定義することができます。

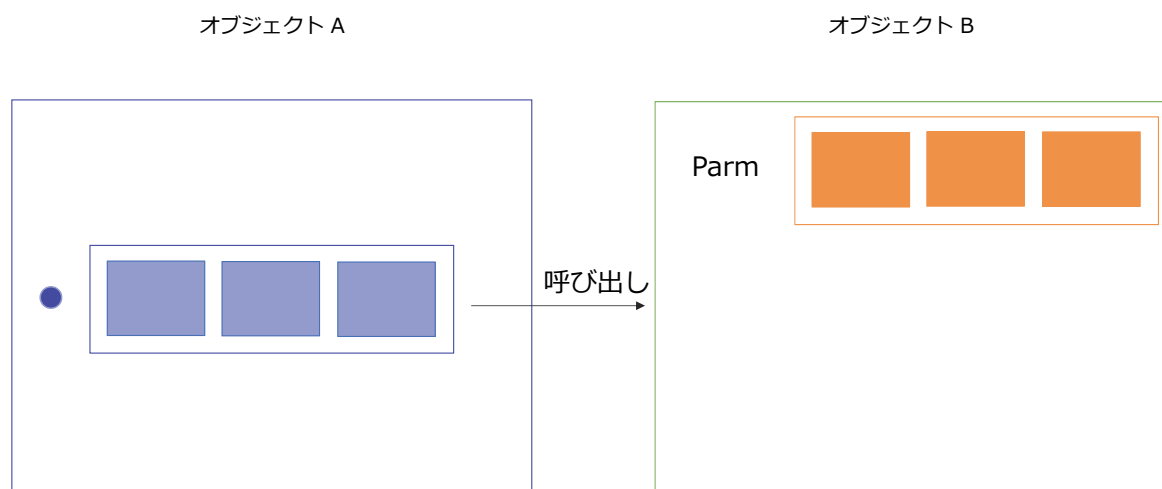
もし、入力させたい値が、データベースに登録済みの値から選択したいという要望の場合、「ダイナミック コンボ ボックス」というタイプのコントロール形式を活用できます。この形式を利用した場合、一般的な「コンボ ボックス」形式と同様に、一覧から値が選択可能となりますが、選択肢として表示される値は、データベースから自動で取得した値が表示されます。

また、PDF を出力するプロシーチャーオブジェクトを呼び出す必要があります。オブジェクトを呼び出すためには、呼び出したいオブジェクト名を記述し、そのあとに「()」を記述します。

もし、パラメーターの受け渡しが必要な場合、この括弧内にその値を指定します。指定する場合には、定数の記述や変数、項目属性などを指定することができます。

Web パネルオブジェクトのインタフェースにおいて、この呼び出しを実行したい場合、ボタンコントロールを配置し、クリックされた場合の処理で実装を行います。このため、[Events] ELEMENTにコードを記述する必要があります。

パラメーターの受領方法



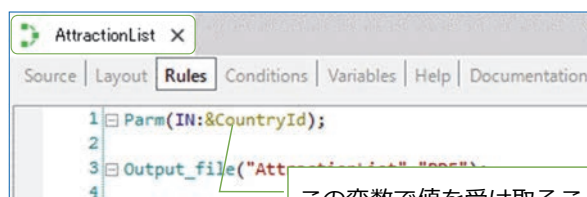
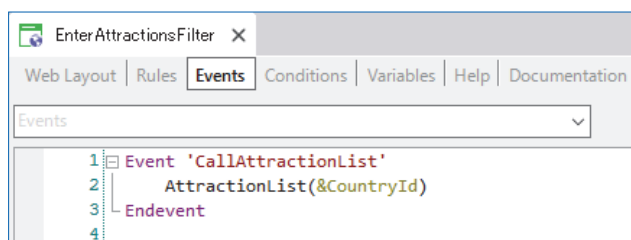
では、オブジェクト A からオブジェクト B を呼び出す方法について説明しました。呼び出したいオブジェクト名を記述し、その末尾に「()」を加える形でした。

そして、パラメーターを渡したい場合、この「()」の間に渡したい値を記述することで、渡すことができることについても説明しました。

では、渡した値をオブジェクト B で再利用するために、受け取る定義は必要でしょうか。もちろん必要となります。そのため、渡された値を受け取るためのルールとして、Parm ルールが用意されています。また、この Parm ルールは、呼び出し元に戻り値を渡すようなケースでは、戻り値を指定するためにも利用します。

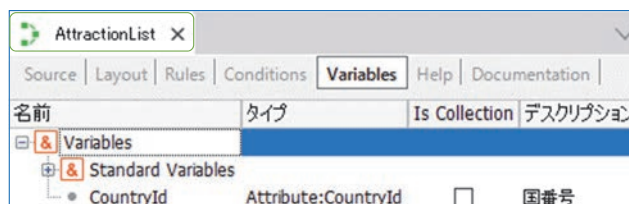
このルールを定義し、パラメーターを利用できるように受け取る定義について引き続き説明していきます。

パラメーターを受け取る定義



```
Print PBTitle
Print PBColumnTitles

For each Attraction
  Order CountryName, AttractionName
  Where CountryId = &CountryId
  Print PBAttractions
Endfor
```



パラメーターを受け取るため、呼び出されるオブジェクトでも値を受け取るための入れ物として、変数を定義する必要があります。

この時、定義する変数は、呼び出し元と互換性のあるタイプに設定されていれば、その他の設定は、呼び出されるオブジェクトに固有の定義で問題ありません。

なぜならば、変数の定義は、オブジェクトに固有であり、ここで受け渡ししている内容は、変数内に格納された値のみとなります。

そして、先ほどの図でも説明済みのように、定義した変数で値を受け取るためには、[Rules] エLEMENTで Parm ルールを定義する必要があります。

このルールは、ここまでに紹介したオブジェクトに限らず、GeneXus で定義し、呼び出すことができる多くのオブジェクトでパラメーターを受け取るために利用するルールです。

Parm ルールの括弧内へ定義した変数名を指定しますが、変数の接頭辞として、IN 演算子を記述できます。

この演算子は、Parm ルールにおいて、この変数が呼び出し元から値を受け取るために定義されたものを明確にします。

IN 演算子と変数の記述の間には、「:」が必要です。

この演算子について、詳細は次の資料で紹介しています。

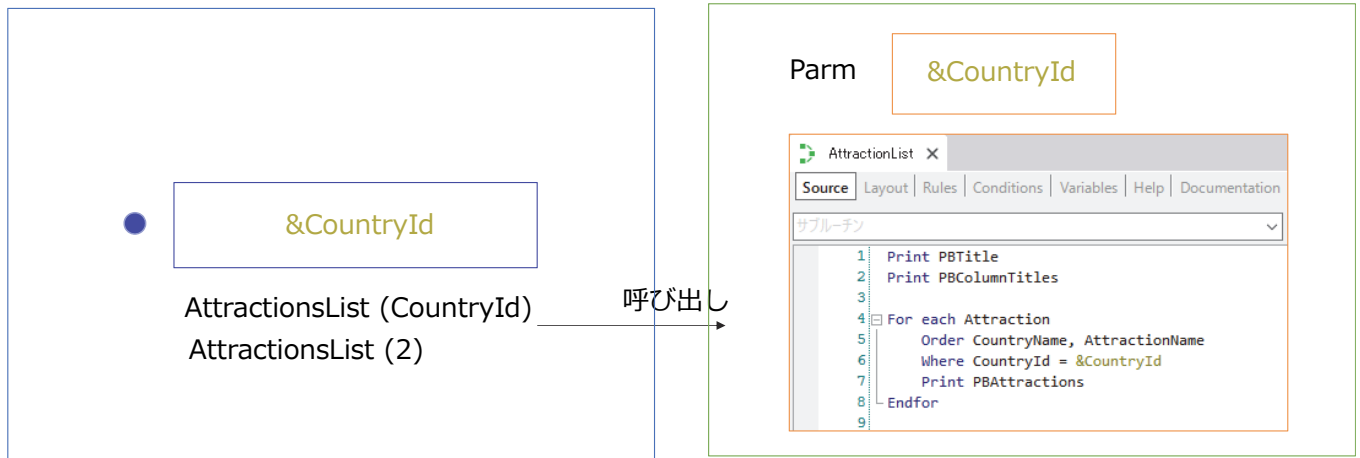
このルールの定義によって、呼び出し元からパラメーターを任意の変数へ受け取ることが出来ました。

呼び出されたオブジェクト側では、この変数を利用することで、受け取った値に基づく処理を行えます。

Parm ルールが定義されたオブジェクト

オブジェクト A

オブジェクト B
(AttractionsList)



Parm ルールが定義されたことにより、呼び出される側となるオブジェクト B は、呼び出し時にパラメーターを受け渡すことができるようになります。
 今回、オブジェクト B として説明に利用したオブジェクトが、プロシーチャーオブジェクトであるため、もともと Launchpad には表示されないオブジェクトでしたが、上記の通り呼び出し時（実行時）に、パラメーターを必要とするオブジェクトと、GeneXus は認識するため、Launchpad に表示されていたオブジェクトであれば、表示されなくなります。

また、今回の例では、呼び出し側となるオブジェクト A では、パラメーターとなる値を変数に格納し、オブジェクト呼び出しの記述で、この変数を引数に指定する形でパラメーターを受け渡していました。

もし、呼び出し元画面で、パラメーターとして受け渡したい値が、項目属性に格納されている場合には、項目属性を引数として指定することもできます。

もちろん、変数や項目属性でなく、このオブジェクトから固定の値、つまり定数を渡して、呼び出したい場合にも定数をそのまま指定することができます。


オブジェクト呼び出しの実行

国番号



フランス

アメリカ
アルゼンチン
イタリア
ウルグアイ
スペイン
パラグアイ
ブラジル
フランス
メキシコ
中国
日本

PDF出力



観光名所一覧

番号	名前	国	写真
3	エッフェル塔	フランス	
1	ルーブル美術館	フランス	

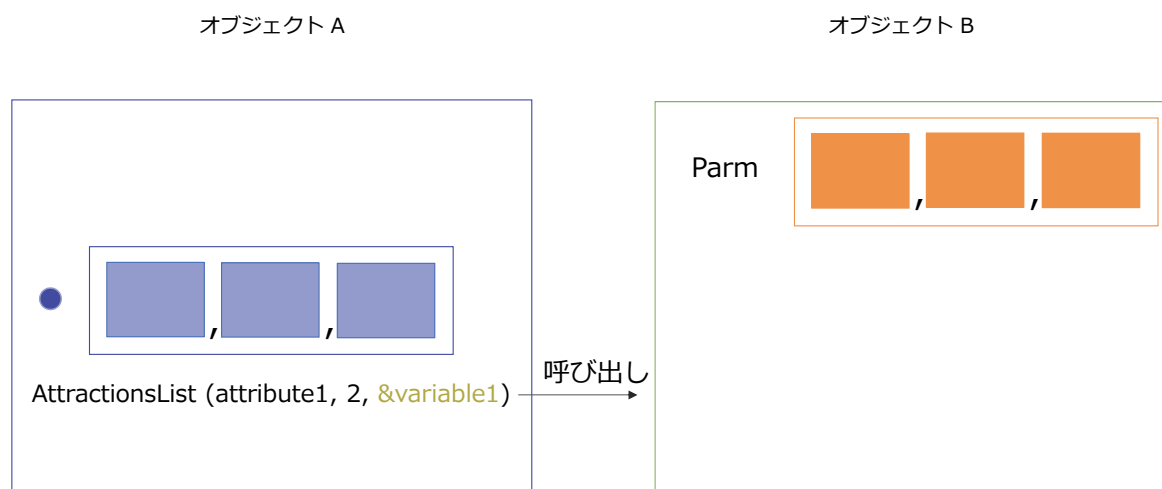
アプリケーションを実行し、結果を確認します。

用意した入力欄は、ダイナミック コンボ ボックスに設定していたため、データベースから自動で値が読み込まれ、選択肢が表示されます。

選択後、配置されているボタンをクリックすると、オブジェクトの呼び出しが発生し、PDF 出力が行われることが確認できます。

この時、表示された PDF は、選択した値によって絞り込みが行われていることが確認できます。

複数パラメーターの受け渡し



ここまでで説明したケースでは、受け渡すパラメーターは、単一のパラメーターでした。もし、複数のパラメーターを受け渡す必要がある場合、どうすればいいでしょうか。

このような場合、オブジェクトを呼び出す際の引数をカンマ区切りで必要なパラメーターとなる値すべてを指定します。そして、受け取る側も同様にカンマ区切りで値を受け取る定義を行います。つまり、Parm ルールの入れ物の定義をカンマ区切りで記述します。

特定のカテゴリ範囲に含まれる観光名所の表示

Event 'CallAttractionsByCategory'
AttractionsByCategory(&CategoryIdFrom,&CategoryIdTo)
Endevent

AttractionsByCategory X

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

```

1 Parm(IN:&IdFrom, IN:&IdTo);
2
3 Output_file("AttractionByCategory", "PDF");
4

```

Source | Layout | Rules | Conditions | Variables

名前	タイプ
& Variables	
Standard Variables	
IdFrom	Numeric(4.0)
IdTo	Numeric(4.0)

サブルーチン

```

1 Print PBTitle
2 Print PBColumnTitles
3
4 For each Attraction
5   Order CountryName, AttractionName
6   Where CategoryId >= &IdFrom
7   Where CategoryId <= &IdTo
8   Print PBAttractions
9 Endfor
10

```

では、どのような場合に複数のパラメーターが必要となるのでしょうか。
一般的に考えられるケースとしては、「範囲を指定する」ケースがあります。
例えば、特定の日付の範囲のデータが必要な場合や、特定の値の範囲というケースです。
範囲の指定には、「開始」と「終了」の2つの値が必要となります。

実際に実装を行う場合、先ほどまでの単一のパラメーターを利用するケースとの差異は、次の2点です。

- ・呼び出し側 : オブジェクトの呼び出し時に、引数をカンマ区切りで複数指定
- ・呼び出される側 : Parm ルールで、値を受け取る入れ物をカンマ区切りで複数指定

ただし、複数のパラメーターを利用する場合、注意すべき点があります。
パラメーターとして、受け渡す側の記述の順番と、受け取る側の順番は一致している必要があります。
つまり、1つ目の引数が「開始」の値、2つ目の引数が「終了」の値として利用することを想定している場合、呼び出し側も呼び出される側もこの順番に沿って定義しなければ、意図した値にならなくなります。

複数パラメーターを利用した呼び出しの実行

国番号	アメリカ ▼
カテゴリ番号 (開始)	1
カテゴリ番号 (終了)	4
<input type="button" value="PDF出力"/> <input type="button" value="PDF出力 (指定カテゴリ内)"/>	


観光名所一覧

番号	名前	国	写真
5	スミソニアン博物館	アメリカ	
4	サグラダ・ファミリア	スペイン	
3	エッフェル塔	フランス	
1	ルーブル美術館	フランス	
2	万里の長城	中国	

アプリケーションを実行し、結果を確認します。

新たに追加した 2 つの入力欄と、ボタンが表示されていることが確認できます。

これらを利用すると、入力した 2 つの値がパラメーターとして受け渡すことができ、PDF 出力の中で、この値が絞り込みに利用されていることが確認できます。

Where 節のオプション When 節

Enter AttractionsFilter ×

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

```
Event 'CallAttractions'
  AttractionsByCategoryAndCountry(&CountryId,&CategoryIdFrom,&CategoryIdTo)
Endevent
```

AttractionsByCategoryAndCountry ×

Source Layout Rules Conditions Variables Help Documentation

名前	タイプ	Is Collection	デスクリプション
Variables			
Standard Variables			
CountryId	Attribute:CountryId	<input type="checkbox"/>	国番号
IdFrom	Numeric(4.0)	<input type="checkbox"/>	Id From
IdTo	Numeric(4.0)	<input type="checkbox"/>	Id To

Source Layout Rules Conditions Variables Help Documentation

```
1 Param(IN:&CountryId, IN:&IdFrom, IN:&IdTo);
2
3 Output_file("AttractionByCategoryAndCountry", "PDF");
4
```

Source Layout Rules Conditions Variables Help Documentation

サブルーチン

```
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
```

パラメーターとして受け取った値をフィルタの条件として利用している場合、受け取った値が空の場合、条件として利用しないように定義することが可能です。

データを表示するために、For each コマンドを利用し、条件を Where 節で定義していた場合、Where 節に対するオプション節に When 節があります。When 節は、Where 節をトリガーする条件を指定するために利用できます。つまり、When 節の条件を満たさない場合、Where 節は、For each コマンドにおける絞り込みの条件として利用されません。

例えば、When 節で、パラメーターを受け取った変数が空の値の場合、条件として利用しないように定義することができます。

この実装により、空の値でデータを絞り込むケースを防ぐことができるようになります。

*GeneXus*TM

training.genexus.com
wiki.genexus.com