

はじめての

GeneXus™ その2

ローコードツールを利用したはじめてのアプリケーション作成

Copyright © GeneXus S.A. 1988-2017.

All rights reserved. 本書は、GeneXus S.A.の明示的許可なしにはいかなるメディアにも複写することはできません。本書の内容は個人的使用のみを目的として提供するものです。

登録商標:

GeneXus は GeneXus S.A.の商標または登録商標です。本書において取り上げているその他の商標はすべて、それぞれの所有財産です。

目次

目次	2
序論	3
既存の GENEXUS アプリケーションを開く	3
WEB PANEL の作成	4
PROCEDURE の作成(LAYOUT)	8
PROCEDURE の作成(内部処理)	12
PROCEDURE の呼び出し	15
[オプション]グリッドに罫線を引く方法	16
[オプション] WEB PANEL に検索機能を追加する方法	20
次のステップ	25

序論

GeneXus は、複数の言語で、異なるプラットフォーム用のソフトウェアアプリケーションを迅速に生成できるローコード開発スイートです。GeneXus には次のような特長があります。習得が簡単で、生産性が高く、クロスプラットフォームで、将来にわたって使用可能であるため、デジタル資産を守ることができ、新しい技術も容易に取り入れることができます。

たとえば、GeneXus では、開発者が選択したターゲットプラットフォーム (特定の言語、データベース、環境、およびプラットフォーム、さらにレスポンシブ Web デザインの有無など) に対応した、Web 用 / モバイルおよびスマートデバイス用のアプリケーションを生成できます。

本書は、GeneXus でアプリケーションを開発するためのビギナーズガイドのその 2 です。

別資料である「FirstStepsWithGeneXus17_JP.pdf」(以下「ファーストステップ」)をすでに実施されている方向けの資料ですのでご注意ください。

本書に関するご質問は次のアドレスにお問い合わせください。

customersupport@genexus.jp

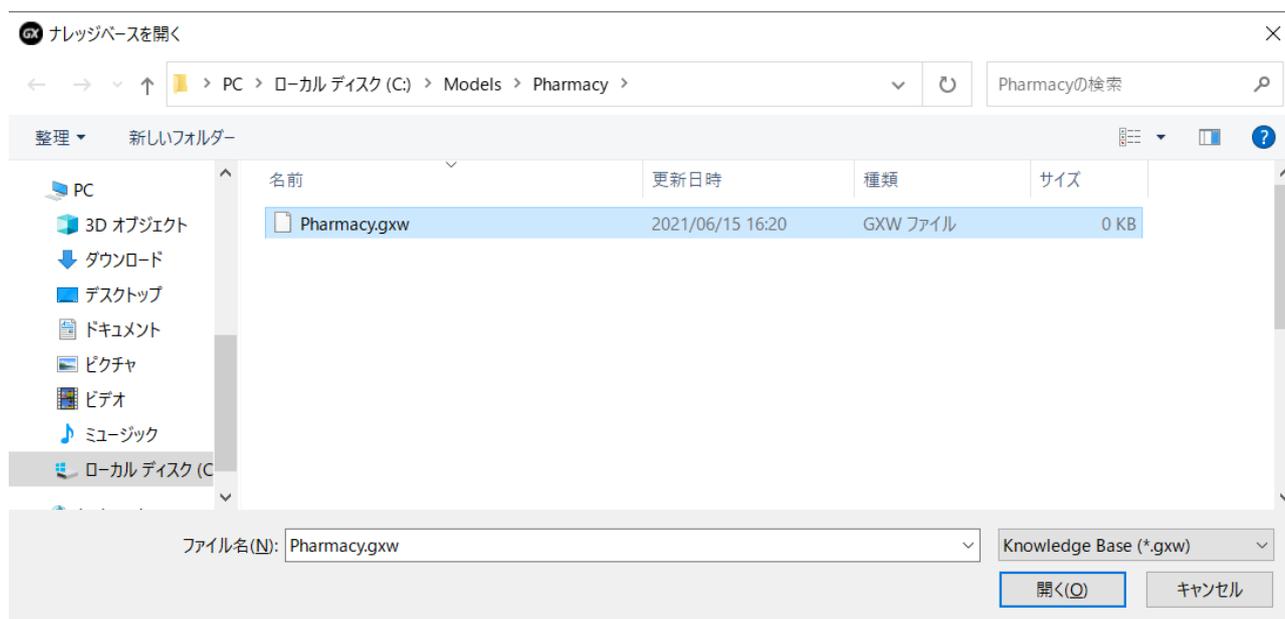
既存の GENEXUS アプリケーションを開く

既存の GeneXus アプリケーションの開発を続けるにあたって、既存**ナレッジベース**を開く必要があります。特にナレッジベースや IDE を閉じていない場合にはそのまま次の単元をご確認ください。

ツールバーで **[ファイル]** → **[開く]** → **[ナレッジベース]** を選択すると、ファイルを開くためのダイアログボックスが表示されます。

ファーストステップにて任意のフォルダに作成した「Pharmacy」(あるいは「PharmacySystem」)フォルダを開き、「Pharmacy.gxw」ファイルを選択し、「開く」を押下することでナレッジベースを IDE に表示することができます。

(デフォルトでは「C:¥Models¥Pharmacy¥Pharmacy.gxw」)



また、直近に作成したナレッジベースである場合には、ツールバーで **[ファイル]**→**[最近の KB]**から選択することも可能です。

WEB PANEL の作成

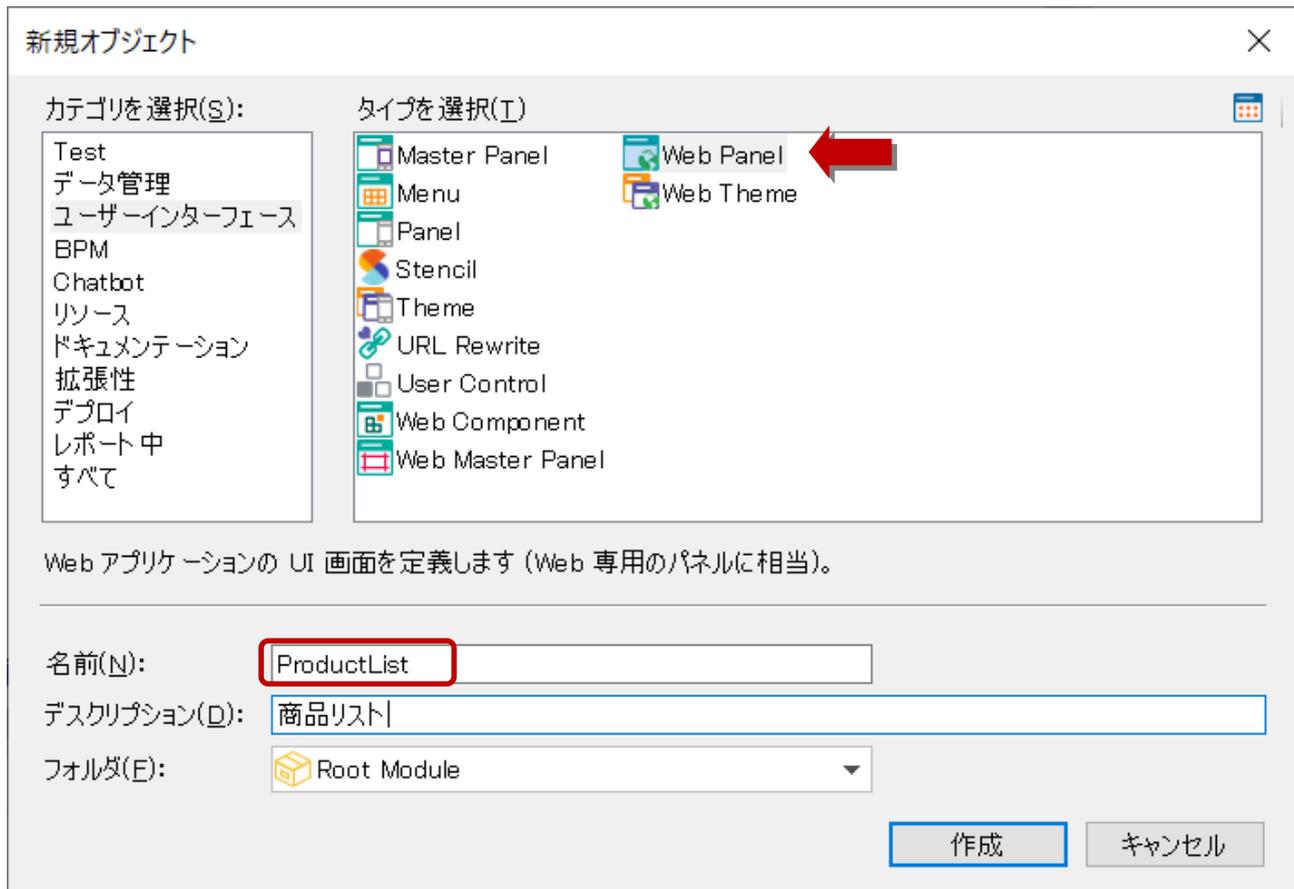
ナレッジベースを開いたら、今回は **Web Panel** を使用して製品(Product)の一覧画面を作成します。

Web Panel はよりカスタマイズ可能なインターフェイスを作成することや詳細な内部処理を行うことが可能です。

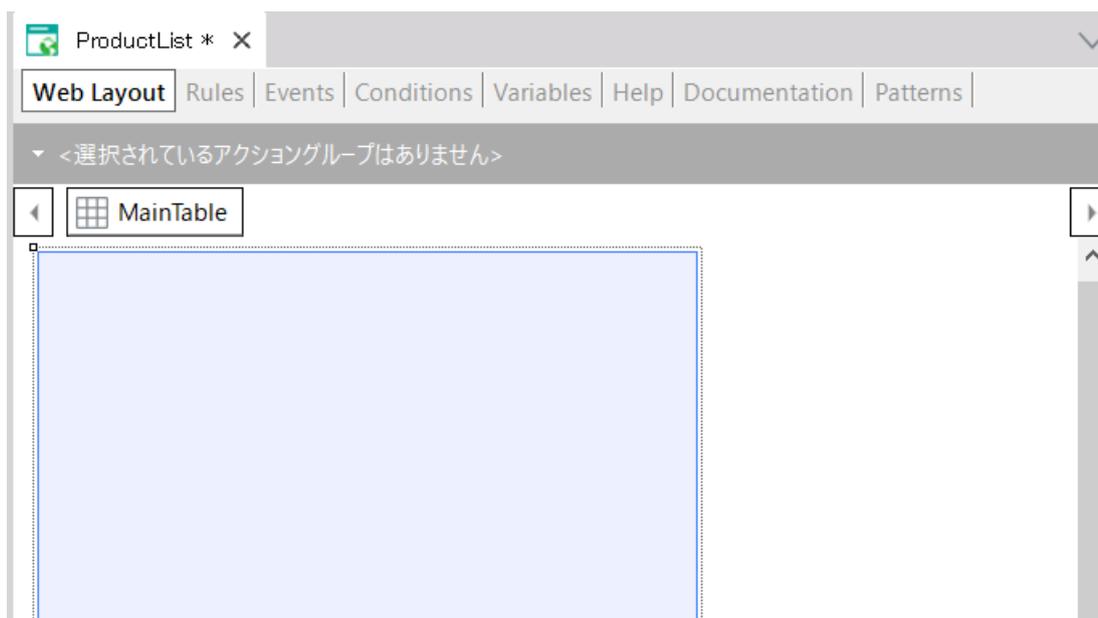
ツールバーで **[ファイル]**→**[新規]**→**[オブジェクト]** を選択すると、次のようなダイアログボックスが表示されます。

ここで、作成するオブジェクトのタイプを選択し、名前を入力します。カテゴリはユーザーインターフェース、タイプは

Web Panel を選択する必要があります。名前は次のように入力します: **ProductList**



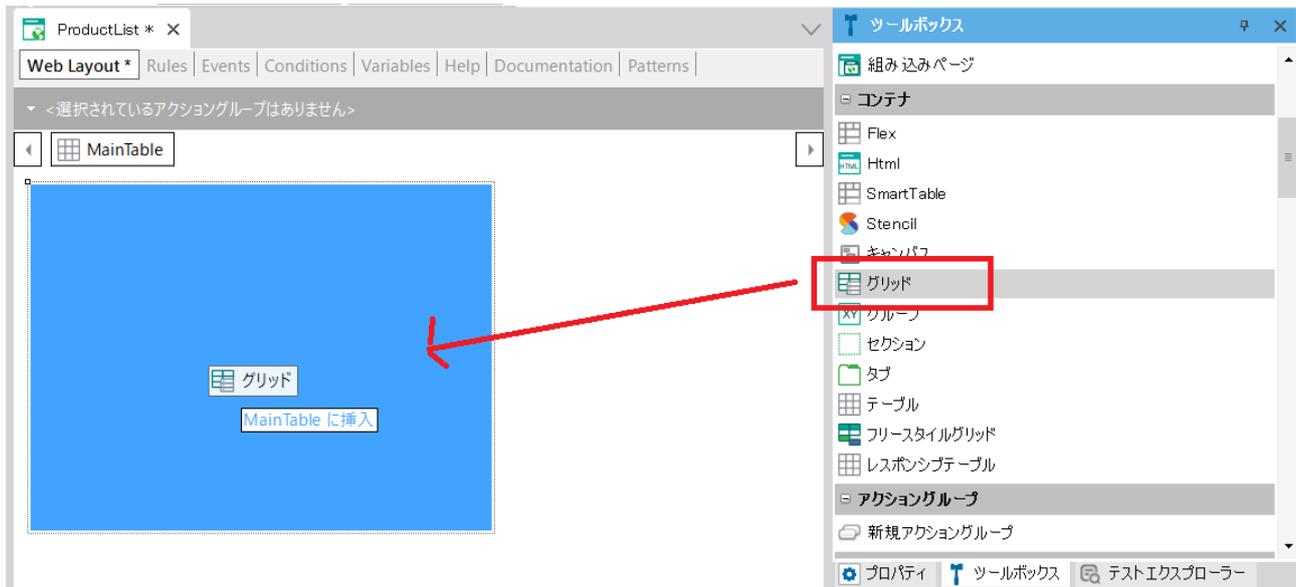
[作成] ボタンをクリックすると、**ProductList** Web Panel 作成され、Web Layout を定義できるように ProductList Web Panel が開きます。



各 Web Panel にはいくつかのセクションがあります。まず、Web Layout では、ドラッグアンドドロップでツールボックス内のコントロールを直感的に配置することが可能です。

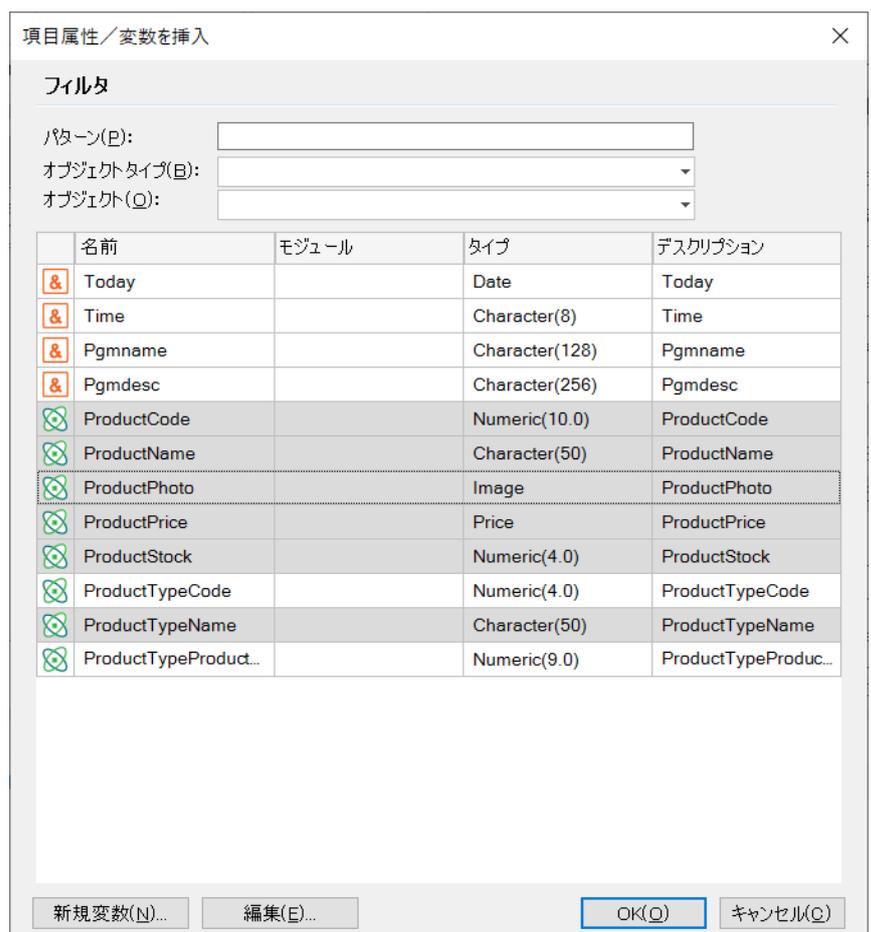
今回は商品のリストを表示する画面を作成します。

ツールボックスから**グリッド**をドラッグアンドドロップで Main Table に配置します。

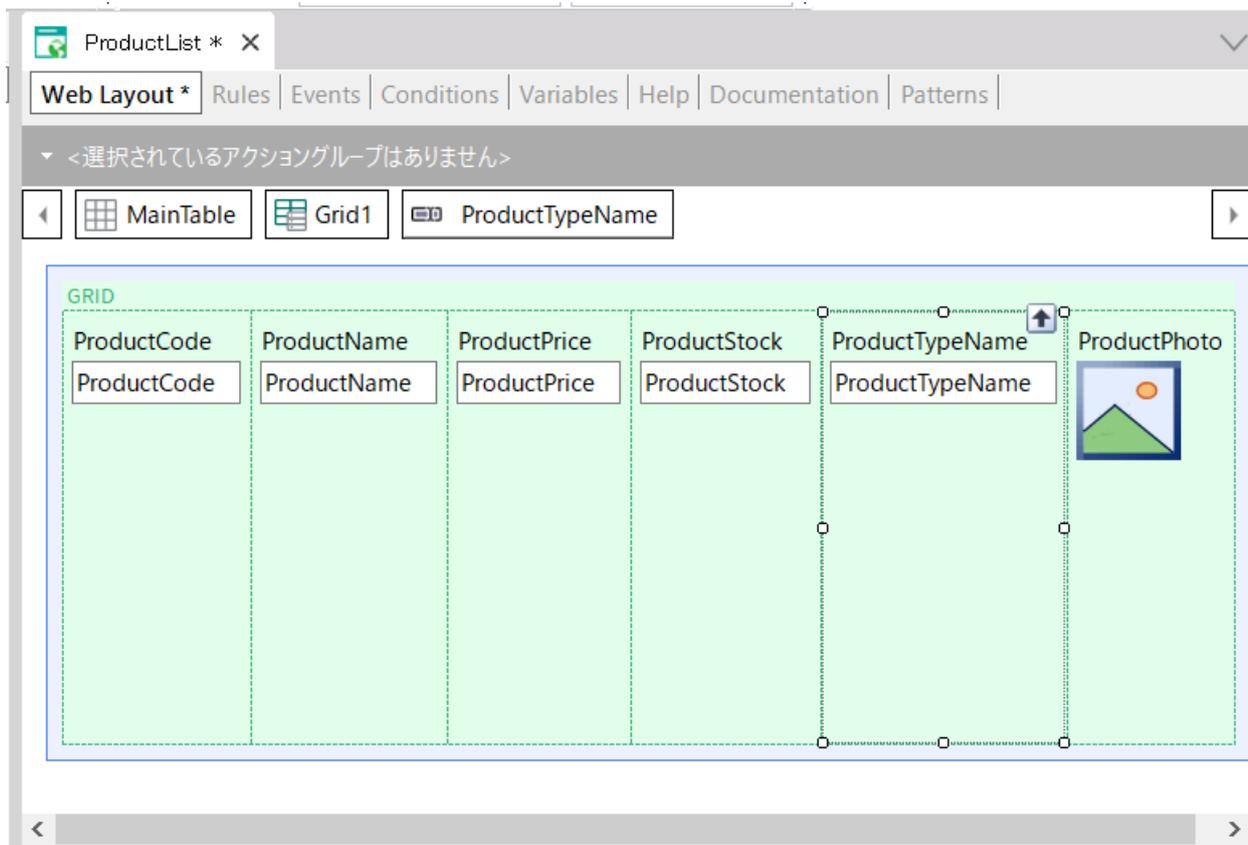


すると次のダイアログが表示されますので、次の項目属性を選択します。

- ProductCode
- ProductName
- ProductPrice
- ProductStock
- ProductTypeName
- ProductPhoto



「OK」を押下することで選択した項目属性がグリッド上に配置されます。



グリッド上の順序が期待した順と違う場合には、ドラッグアンドドロップで順序を変更可能です。

以上で実装完了です。

F5 を押下してビルドし、作成した ProductList Web Panel オブジェクトを確認しましょう。

次のように一覧画面が作成されています。

ProductCode	ProductName	ProductPrice	ProductStock	ProductTypeName	ProductPhoto
1	赤リンゴ	100.00	5	フルーツ	
2	青リンゴ	120.00	7	フルーツ	
3	黄リンゴ	140.00	4	フルーツ	

PROCEDURE の作成 (LAYOUT)

続いて、**Procedure** オブジェクトを使用して、製品(Product)一覧の帳票を作成します。

Procedure オブジェクトは内部処理や帳票出力を行うことが可能です。

もう一度ツールバーから **[ファイル]** → **[新規]** → **[オブジェクト]** を選択し、ダイアログボックスが表示します。今回は、カテゴリはデータ管理、タイプは Procedure を選択する必要があります。名前は次のように入力します：

ProductListOutput

新規オブジェクト

カテゴリを選択(S):

- Test
- データ管理
- ユーザーインターフェイス
- BPM
- Chatbot
- リソース
- ドキュメンテーション
- 拡張性
- デプロイ
- レポート中
- すべて

タイプを選択(T):

- API
- Data Provider
- Data Selector
- Data View
- Domain
- Procedure
- Structured Data Type
- Subtype Group
- Transaction

データベースアクセス、データ更新、および印刷を含むアルゴリズムを実装するプログラムまたはルーチンを定義します。

名前(N): ProductListListOutput

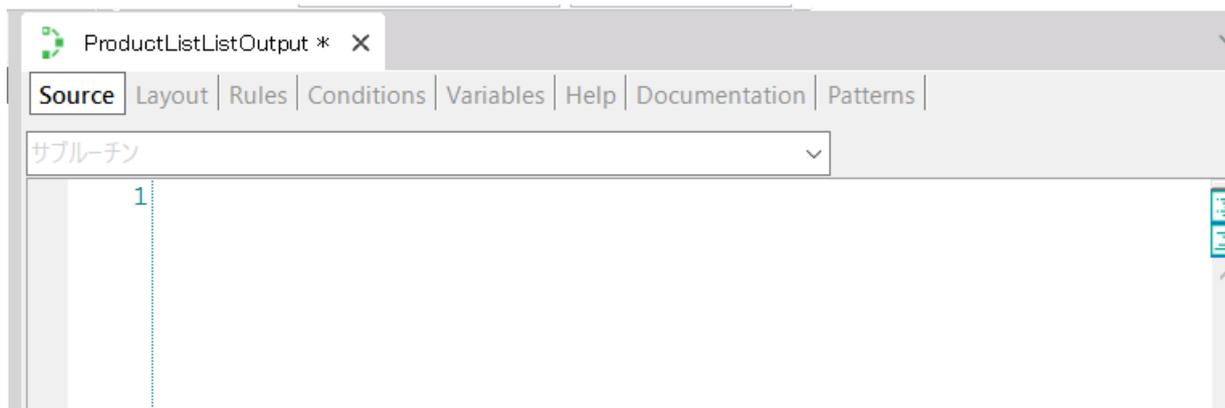
説明(D): 商品リスト 帳票出力

フォルダ(F): Root Module

作成 キャンセル

[作成] ボタンをクリックすると、**ProductListOutput** Procedure が作成され、Source を定義できるように

ProductListOutput Procedure が開きます：

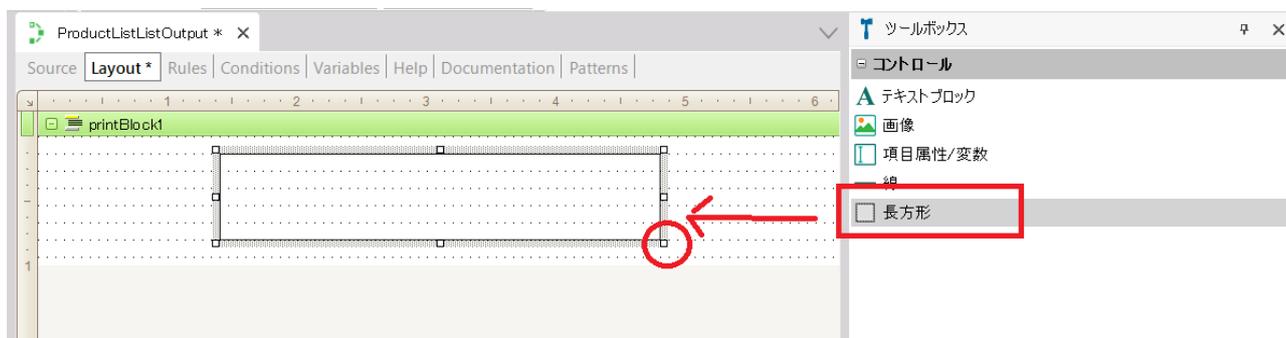


各 Procedure にはいくつかのセクションがあります。まず、Source では、内部処理を記述できます。Layout ではドラッグアンドドロップでツールボックス内のコントロールを直感的に配置することで帳票が作成可能です。

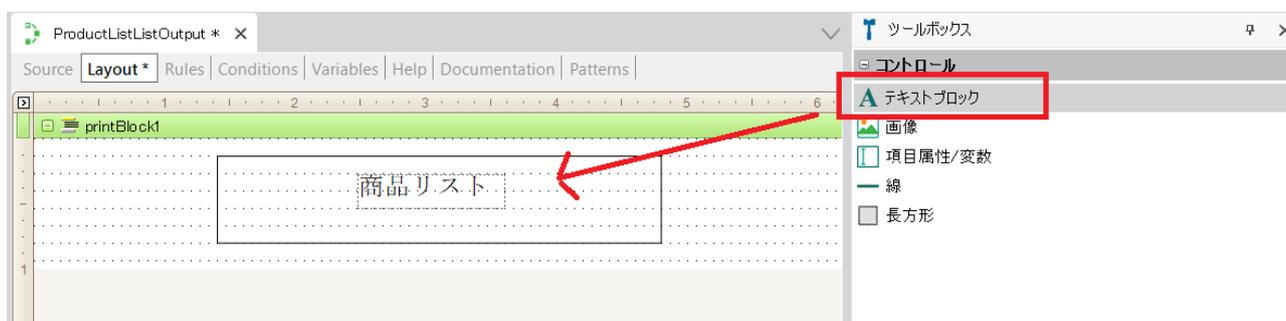
今回は商品のリストを表示する帳票を作成します。

まず、Layout のタブをクリックし、移動します。

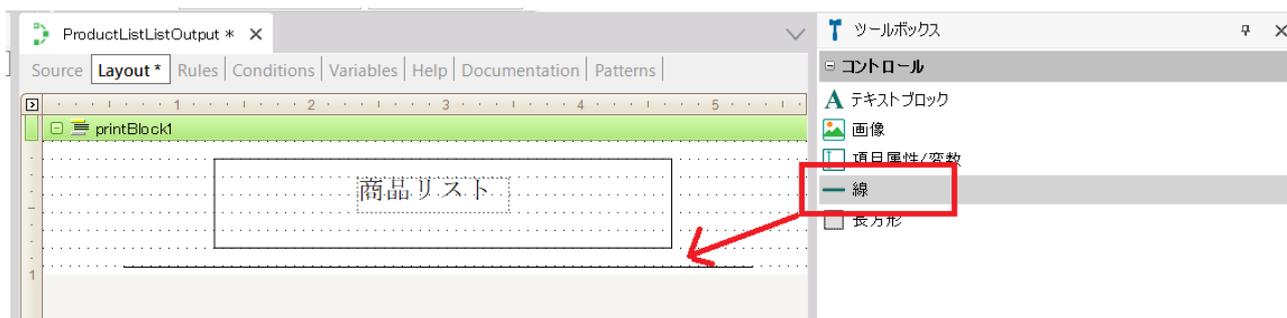
帳票のタイトルを作成するために、既定として最初から配置されている printBlock1 にツールボックスから**長方形**をドラッグアンドドロップで配置し、右下の四角をドラッグして任意のサイズに変更します。



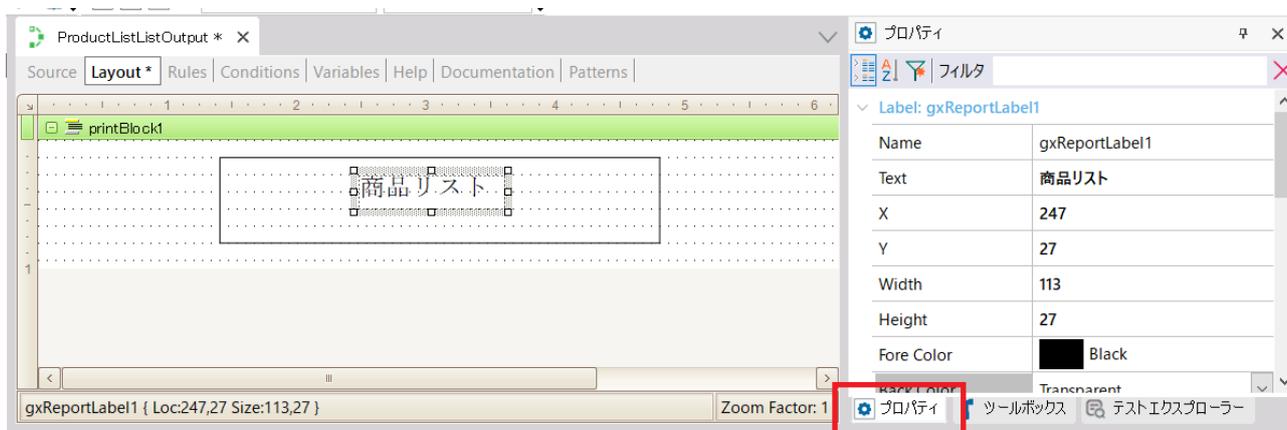
続いて**テキストブロック**をドラッグアンドドロップで長方形内に配置し、その後にテキストブロックをダブルクリックし、**商品リスト**と書き換えます。



さらに、タイトルと帳票データの境界線として**線**を配置し、ドラッグで任意のサイズに変更します。



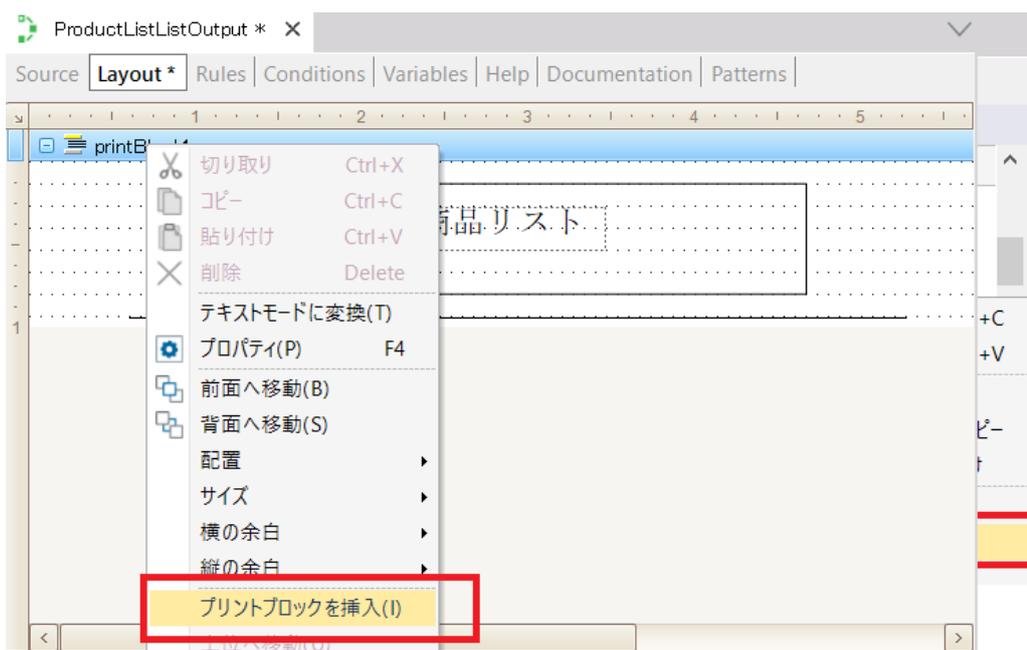
ここまで、ドラッグやダブルクリックなどで直感的に操作しましたが、各コントロールをクリックしてプロパティタブを選択することによってより詳細な設定をすることが可能です。



帳票のタイトルにつきましては完了しましたので、次は帳票のデータについて作成します。

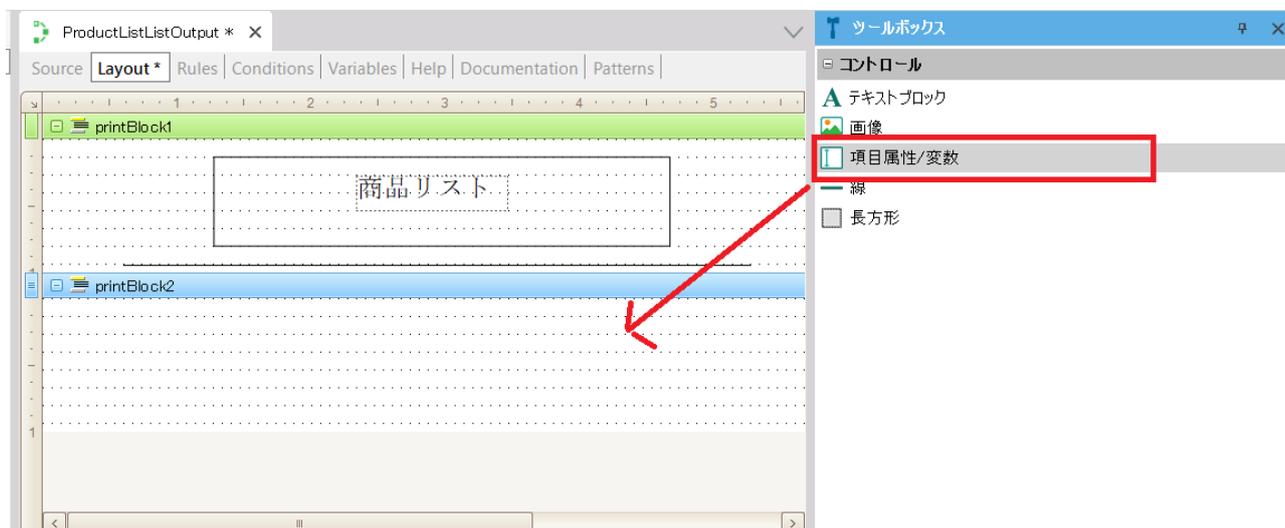
Layout 内の任意の場所を右クリックし、**プリントブロックを挿入**を選択します。

こちらの操作により、新規 printBlock が作成されます。



新規作成した printBlock2 がタイトルの printBlock1 よりも上に配置されてしまった場合には、printBlock2 を右クリックして、**下位へ移動**を選択することによって配置を変更できます。

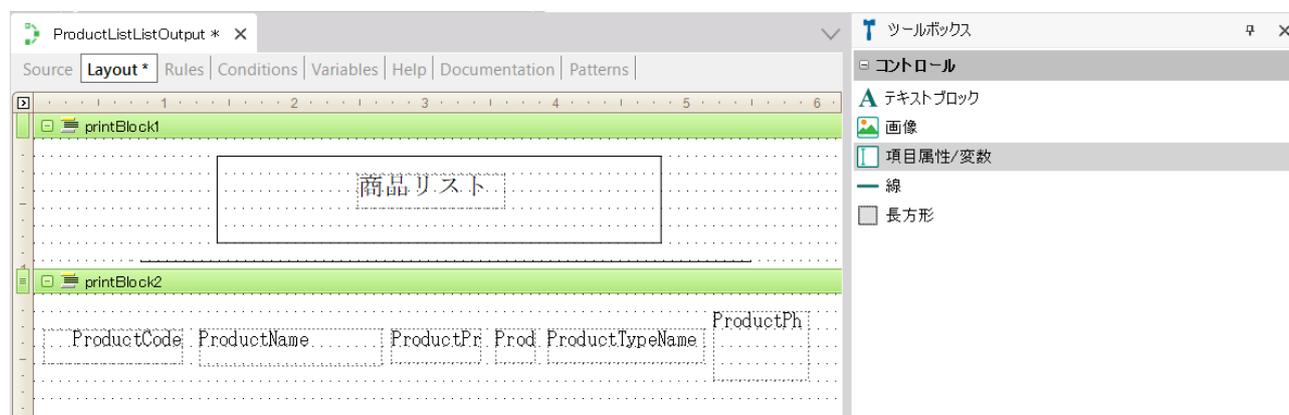
作成した printBlock2 にツールボックスから**項目属性/変数**をドラッグアンドドロップで配置します。



すると、項目属性選択のダイアログが表示されますので、次の項目属性を選択して配置します。

- ProductCode
- ProductName
- ProductPrice
- ProductStock
- ProductTypeName
- ProductPhoto

※Web Panel の時と違い、Procedure の Layout では項目属性は一つずつしか配置できません。



長方形やテキストブロック、線と同様に項目属性/変数も詳細な設定が可能ですので、任意で追加の設定を行ってください。

また、ここでも下部に各データの区切りとして線を配置します。



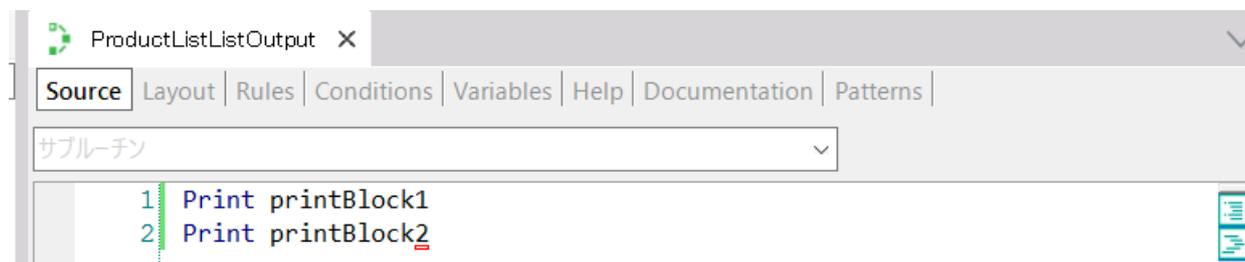
ここまででいったん Layout の設定は終了し、帳票出力の内部処理の設定を行います。

PROCEDURE の作成 (内部処理)

作成した Layout を帳票として出力するには、次の手順が必要となります。

まず、**Source** のタブを開き、次のように作成した printBlock を **Print** する処理を記述します。

- **Print** printBlockX



ただし、このままだとどのデータを出力するか GeneXus が認識できていないため、**For Each** を使用して DB にあるデータ毎に繰り返し出力を行います。

- **For Each** <Transaction 名>

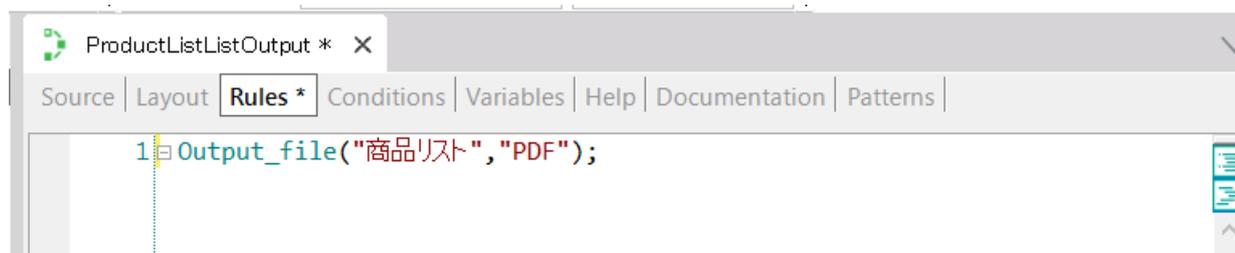
<処理>

EndFor



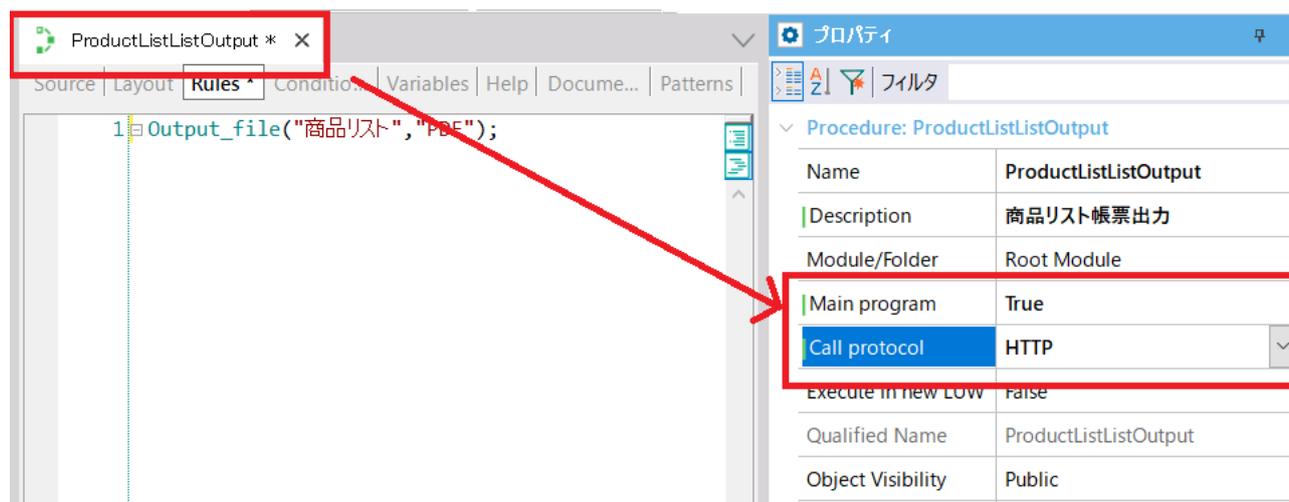
続いて、**Rules** のタブを開き、次のように **Output_File** ルールを使用して PDF として出力することを宣言します。

- **Output_file**("<帳票の名称>", "PDF");



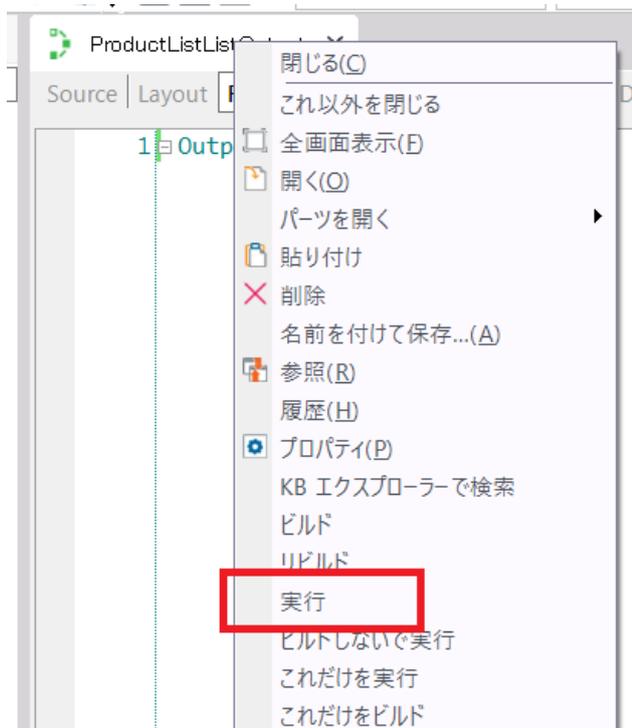
さらに、ProductListOutput をクリックして表示されるプロパティにて、次のように設定を行います。

- **Main program = True**
- **Call protocol = HTTP**



これで Procedure を帳票として出力する設定は完了です。

ProductListOutput を右クリックして**実行**を行うと実際にどのように出力されるか確認できます。

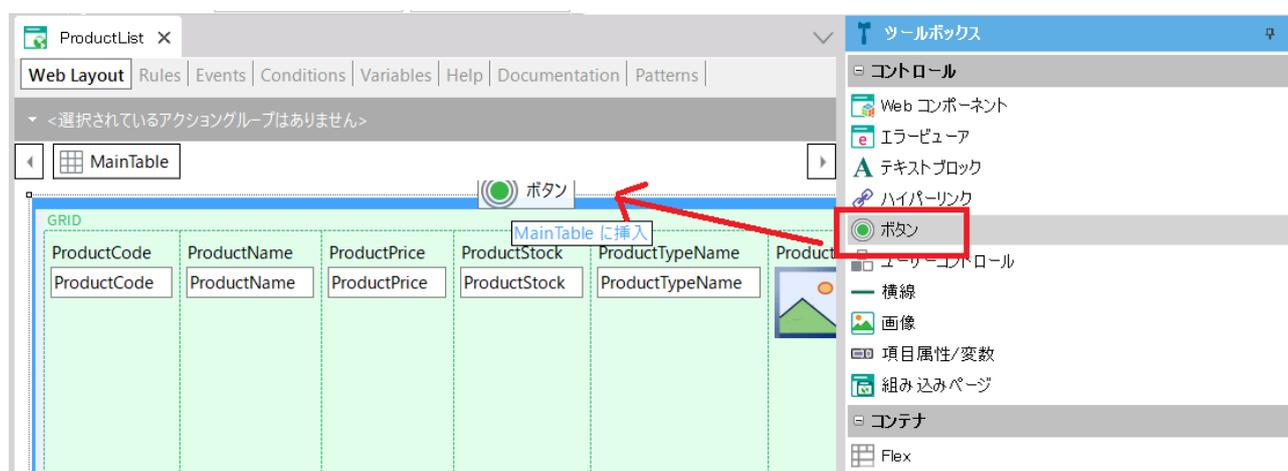


商品リスト			
1	赤リンゴ	100.00	5 フルーツ 
2	青リンゴ	120.00	7 フルーツ 
3	黄リンゴ	140.00	4 フルーツ 

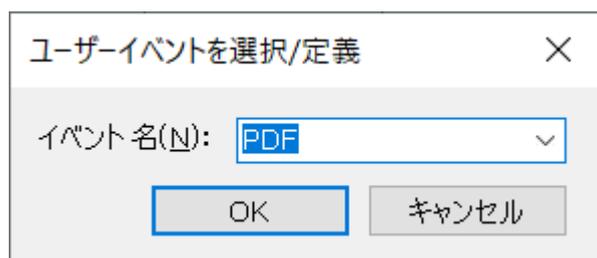
PROCEDURE の呼び出し

続いて、上記で作成した帳票を実際の Web アプリケーションで出力する処理を ProductList Web Panel に実装します。ボタンを押下した際に帳票が出力されるような実装を行います。

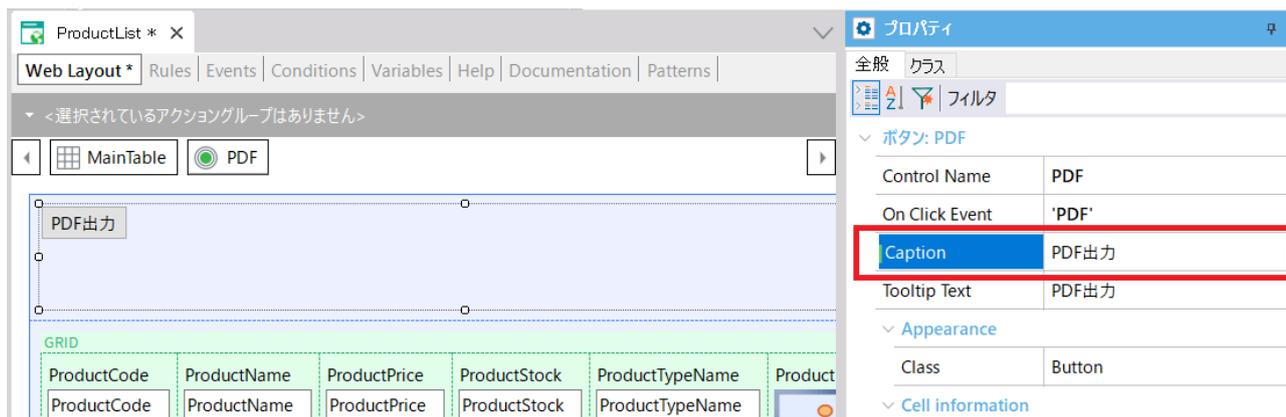
まず、作成した ProductList Web Panel を開き、ツールボックスから**ボタン**をドラッグアンドドロップで Web Layout に配置します。



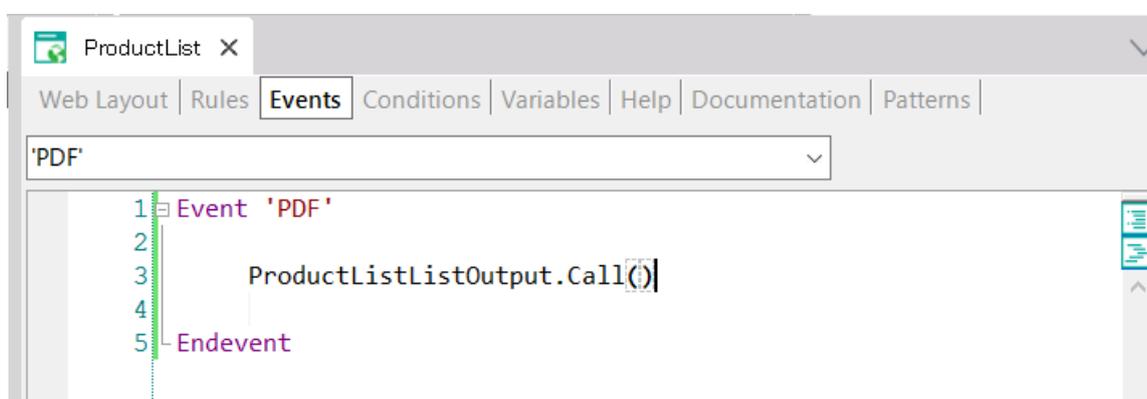
すると、ユーザーイベントを選択/定義のダイアログが表示されるので、イベント名に **PDF** を設定します。



Web Layout に配置された PDF ボタンを選択し、**Caption** プロパティからボタンに実際に表示される文字が設定可能です。



Web Layout に配置された PDF ボタンをダブルクリックすることで、自動的に Events タブへ移動し、“PDF”イベントが設定されますので、このイベント内に ProductListListOutput プロシージャを **Call** する記述をします。



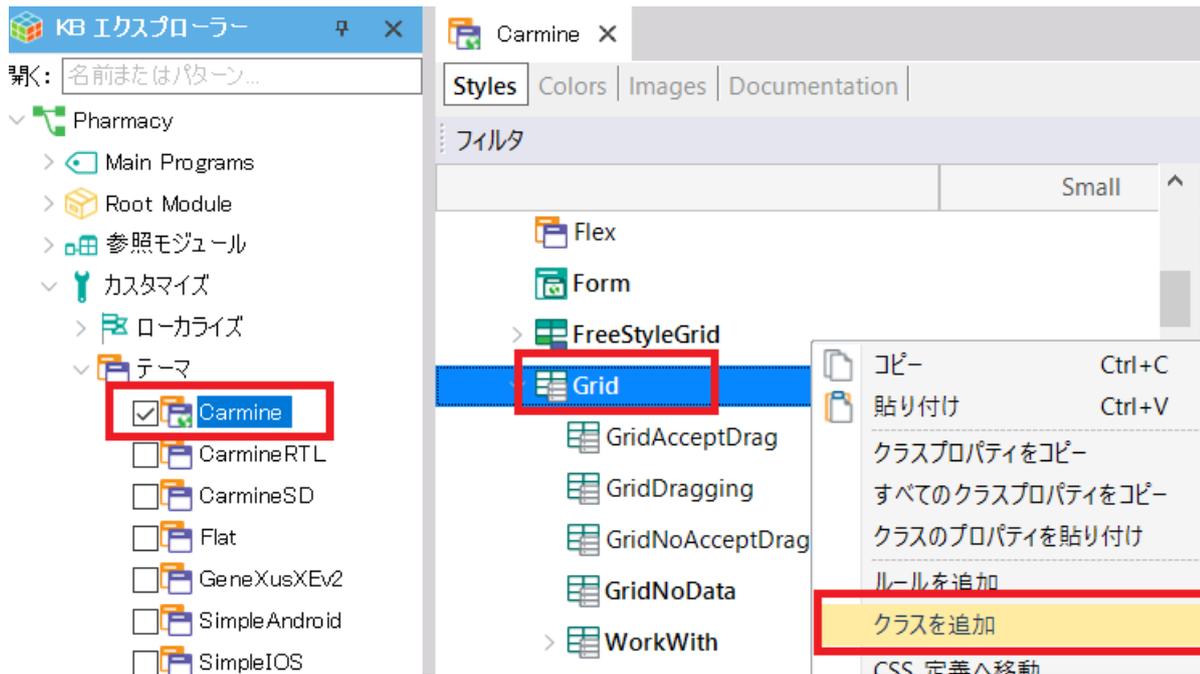
これだけで実装完了です。

開発者メニューから ProductList Web Panel を開き、PDF ボタンをクリックすることで帳票出力可能です。

[オプション]グリッドに罫線を引く方法

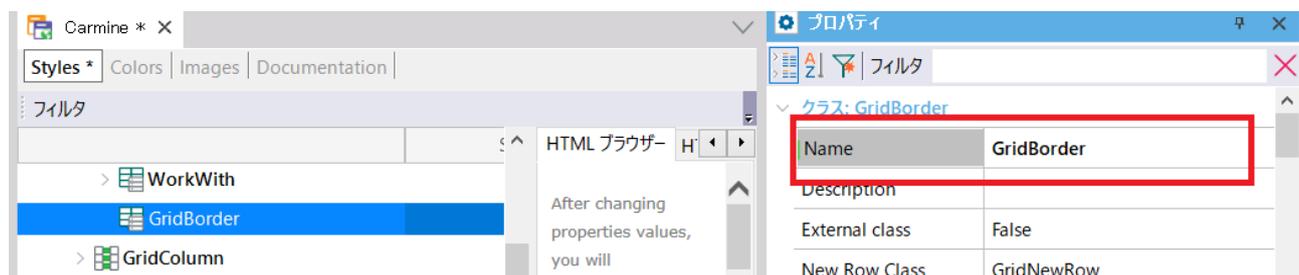
「WEB PANEL の作成」にて作成した一覧グリッドにて罫線を引きたいという要望が出た場合には次の方法が有効です。今回は罫線用のクラスを作成し、こちらを ProductList 内グリッド上のコントロールに設定します。

KB エクスプローラー内の **カスタマイズ** → **テーマ** → **Carmine** をダブルクリックで開き、**Grid** を右クリックして **クラスを追加** を選択します。



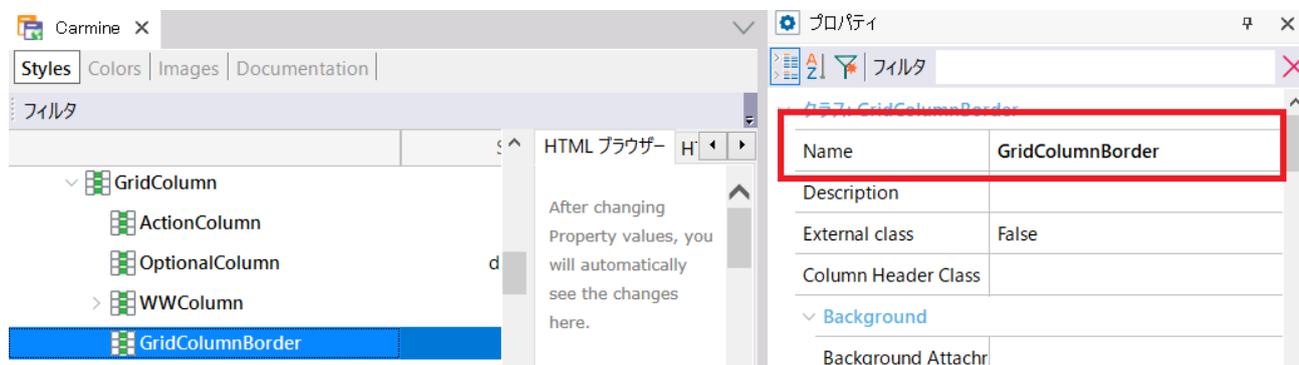
新規作成した Grid クラスに次の名前を設定します。

GridBorder



同様に、GridColumn クラス配下にクラスを追加し、次の名前を設定します。

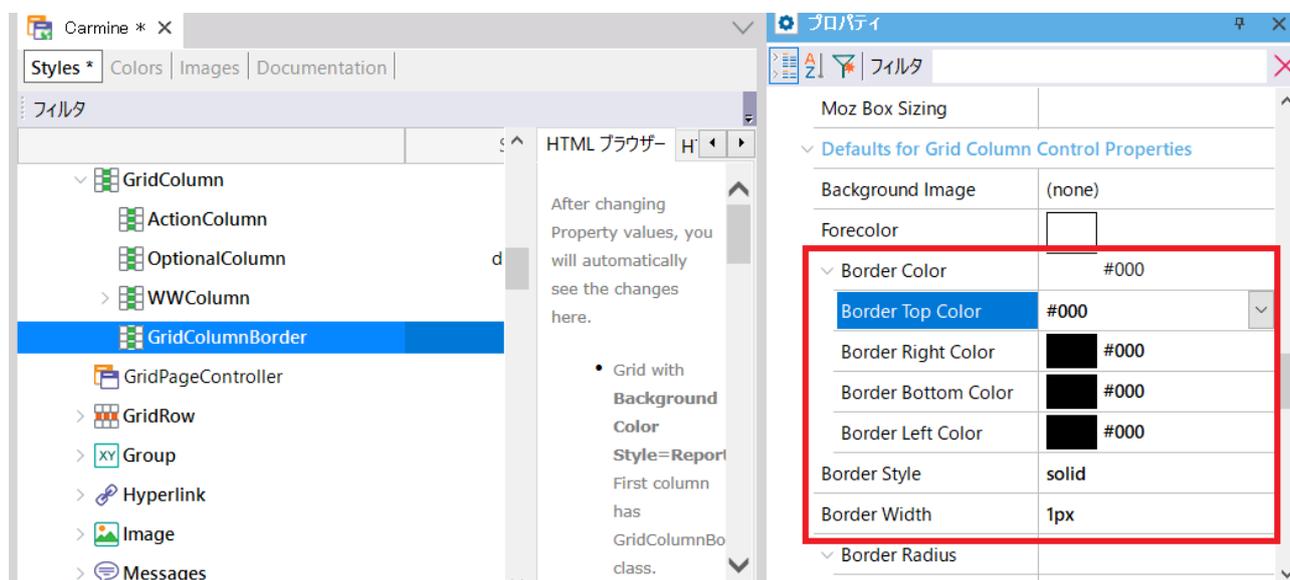
GridColumnBorder



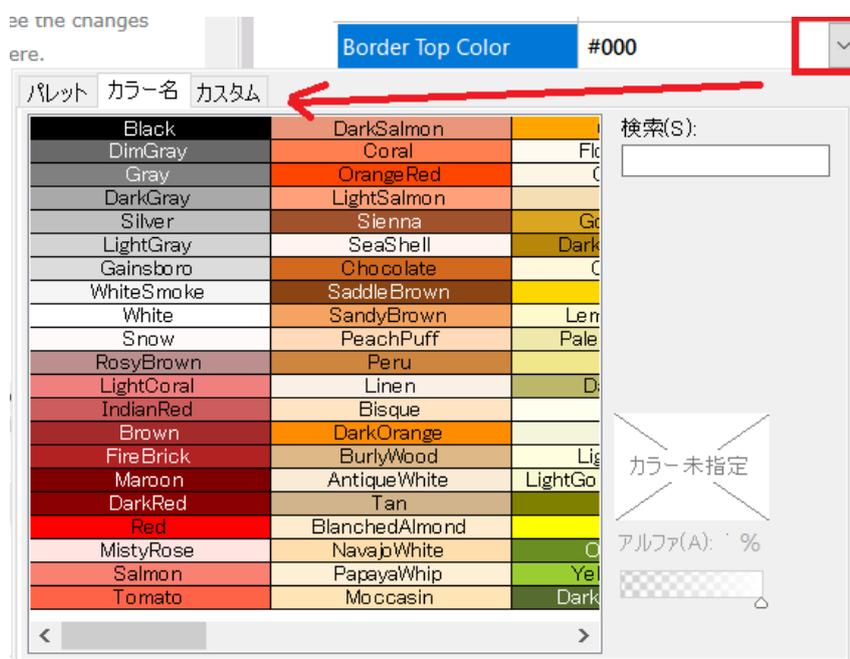
作成した GridColumnBorder クラスにて次のように各プロパティを設定します。

- **Border Top Color:** 任意の値(※)
- **Border Right Color:** 任意の値(※)

- **Border Bottom Color:** 任意の値(※)
- **Border Left Color:** 任意の値(※)
- **Border Style:** solid
- **Border Width:** 任意の値(単位は px)

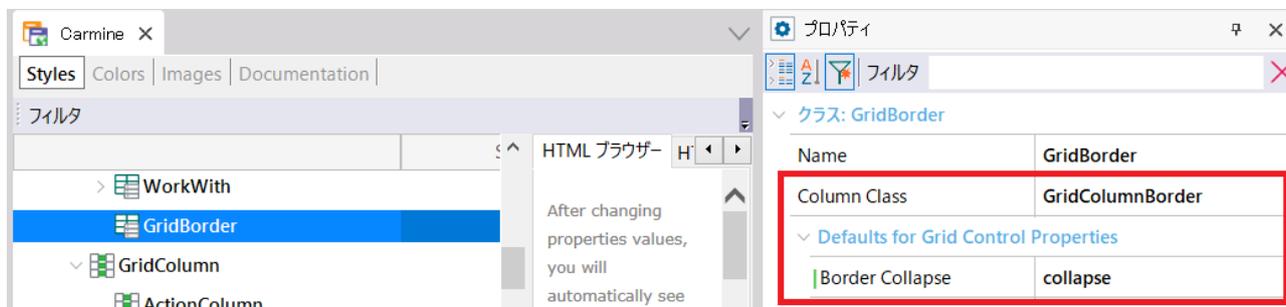


(※)各 Border Color プロパティの右側の下矢印アイコンをクリックすることでパレット、カラー名、カスタムから任意の色を設定可能です。



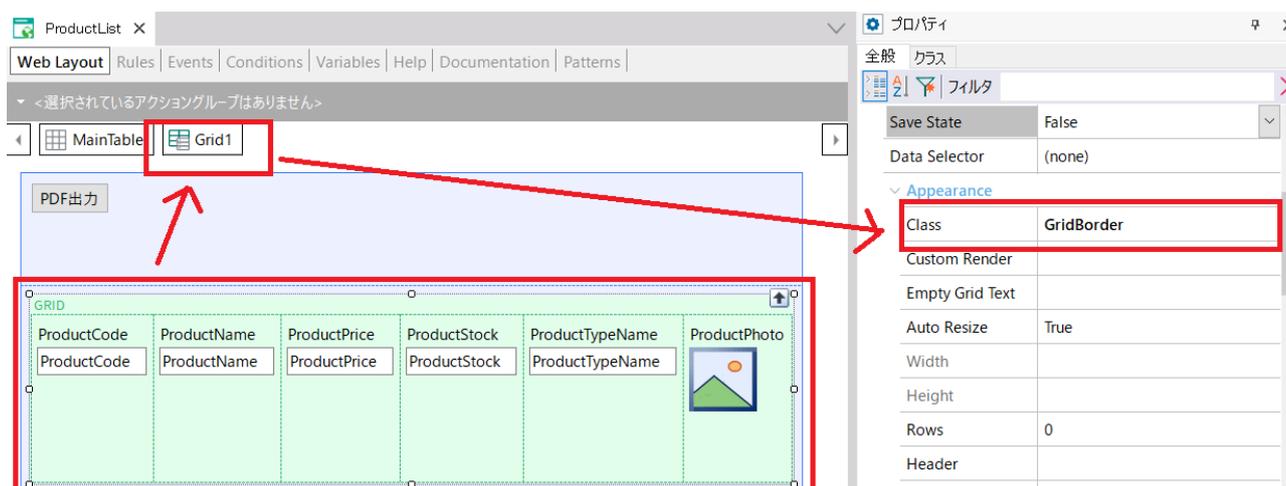
続いて、作成した GridBorder クラスにて次のように各プロパティを設定します。

- **Column Class:** GridColumnBorder
- **Border Collapse:** collapse

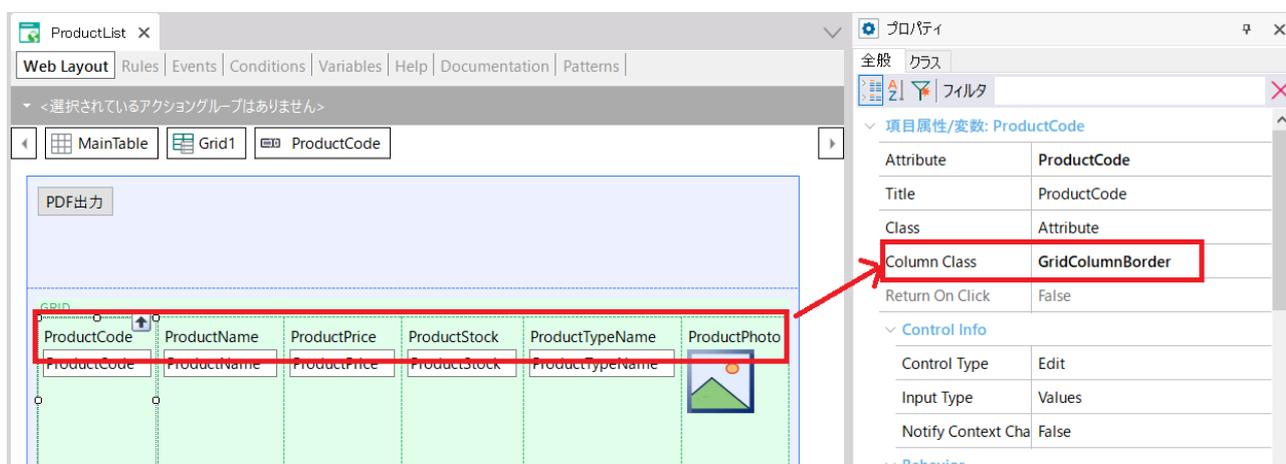


最後に、作成したクラスを Web Panel ProductList に設定します。

まず、Grid の Class プロパティに GridBorder を設定します。



次に、Grid 上の各項目属性の Column Class プロパティに GridColumnBorder を設定します。



以上で実装完了です。

PDF出力					
ProductCode	ProductName	ProductPrice	ProductStock	ProductTypeName	ProductPhoto
1	赤リンゴ	100.00	5	フルーツ	
2	青リンゴ	120.00	7	フルーツ	
3	黄リンゴ	140.00	4	フルーツ	

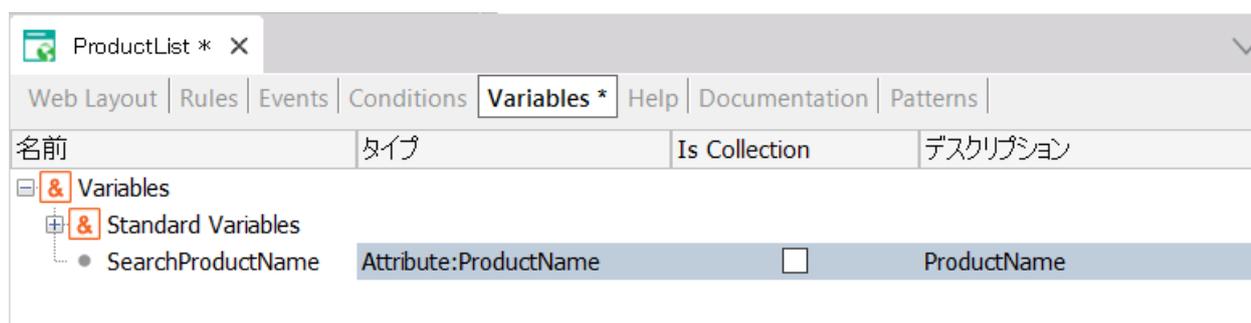
[オプション] WEB PANEL に検索機能を追加する方法

「WEB PANEL の作成」にて作成した一覧グリッドにて検索(フィルター)機能をつけたいという要望が出た場合には次の方法が有効です。

今回は **ProductName** にて検索を行う機能を作成します。

ProductList Web Panel の **Variables** タブにて検索ワード入力用の変数として次の変数を追加します。

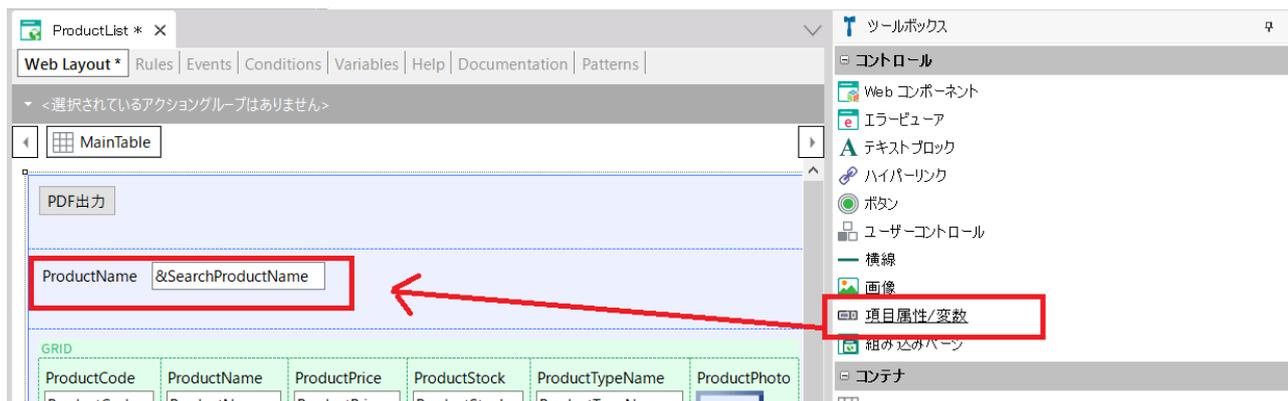
SearchProductName



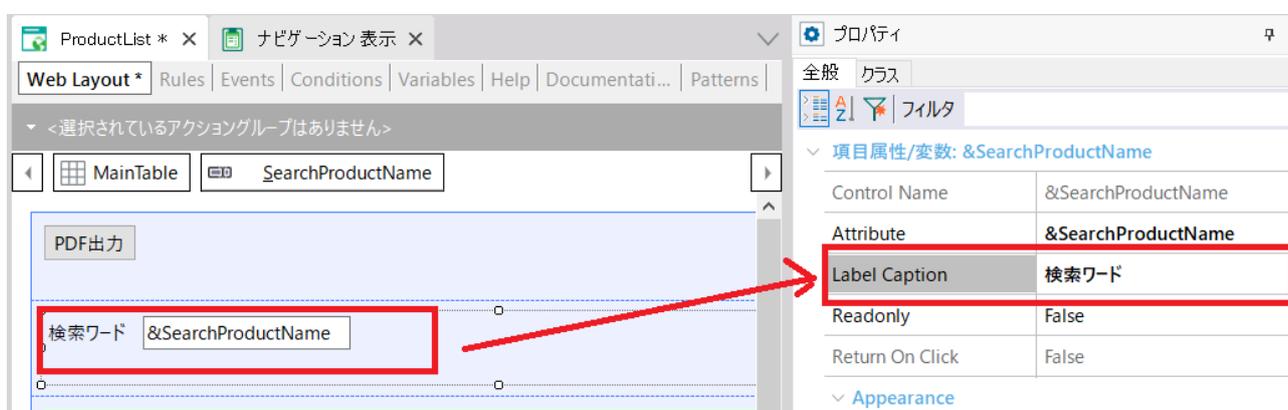
The screenshot shows the 'Variables' tab in the GeneXus IDE. A tree view on the left shows 'Variables' expanded to 'Standard Variables', with 'SearchProductName' selected. The main table displays the configuration for 'SearchProductName':

名前	タイプ	Is Collection	デスクリプション
SearchProductName	Attribute:ProductName	<input type="checkbox"/>	ProductName

Web Layout タブへ移行し、ツールボックスから項目属性/変数を Web Form 上にドラッグアンドドロップし、先ほど作成した SearchProductName を選択します。



配置した ProductName の **Label Caption** プロパティにて検索ワード入力フィールド用の文字列を設定可能です。



続いて **Conditions** タブへ移動し、次のようなコードを記述します。

- `ProductName like &SearchProductName when not &SearchProductName.IsEmpty();`



このコードにより、検索用フィールドに入力した文字から始まる ProductName を持つレコードのみが表示されます。



こちらの検索ワードとの一致の粒度を変更したい場合には「ProductName like &SearchProductName」の部分で、下記 Wiki を参考に变更してください。

参考 Wiki : Like 演算子

<http://wiki.genexus.jp/hwikibypageid.aspx?9991>

例：任意の場所に検索ワードの文字列を含む名前の場合は次のようなコードとなります。

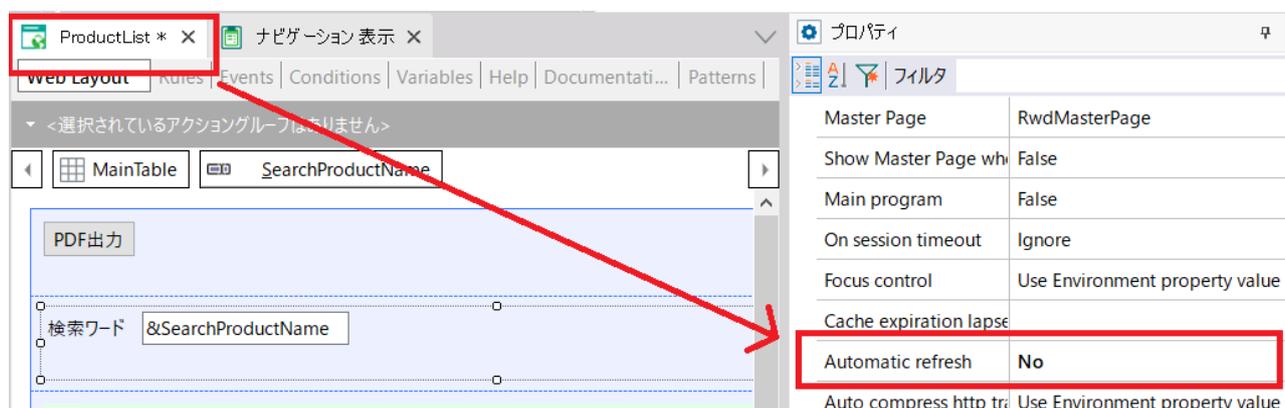
```
ProductName like "%" + &SearchProductName when not &SearchProductName.IsEmpty();
```

```
1 //ProductName like &SearchProductName when not &SearchProductName.IsEmpty();
2 ProductName like "%" + &SearchProductName when not &SearchProductName.IsEmpty();
```

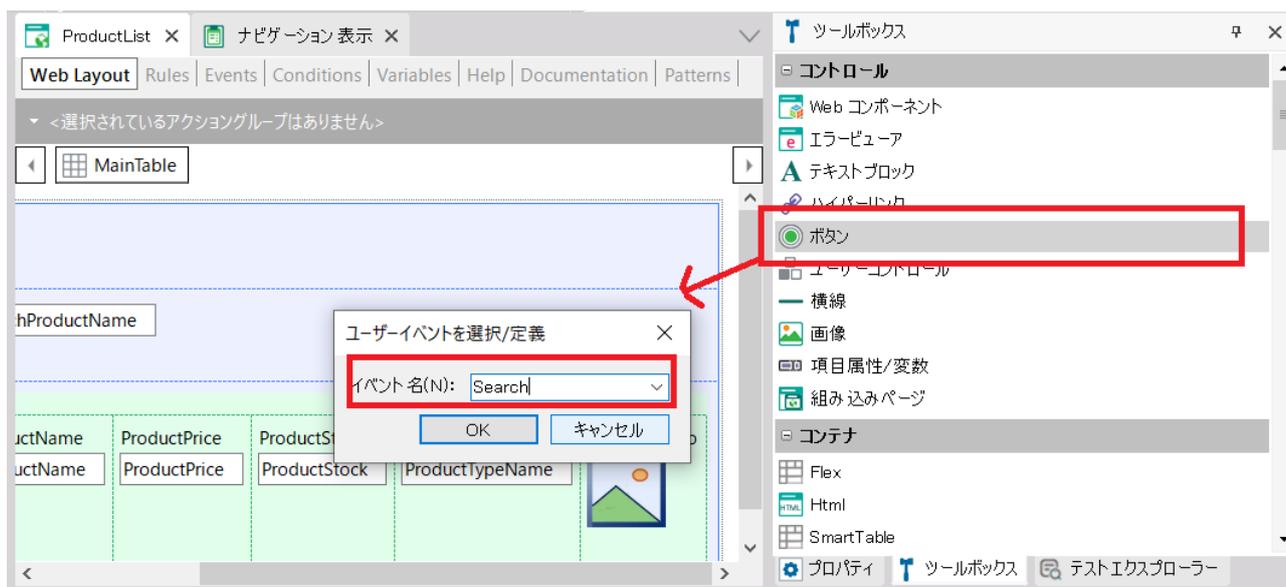
以上までで検索機能は実装可能ですが、現時点では GeneXus のデフォルトの機能にて、検索ワードを入力した瞬間にグリッド上に表示されるレコードが変更されます。

こちらの挙動を、検索ボタンを押下したタイミングで変更されるようにしたい場合には次の手順が有効です。

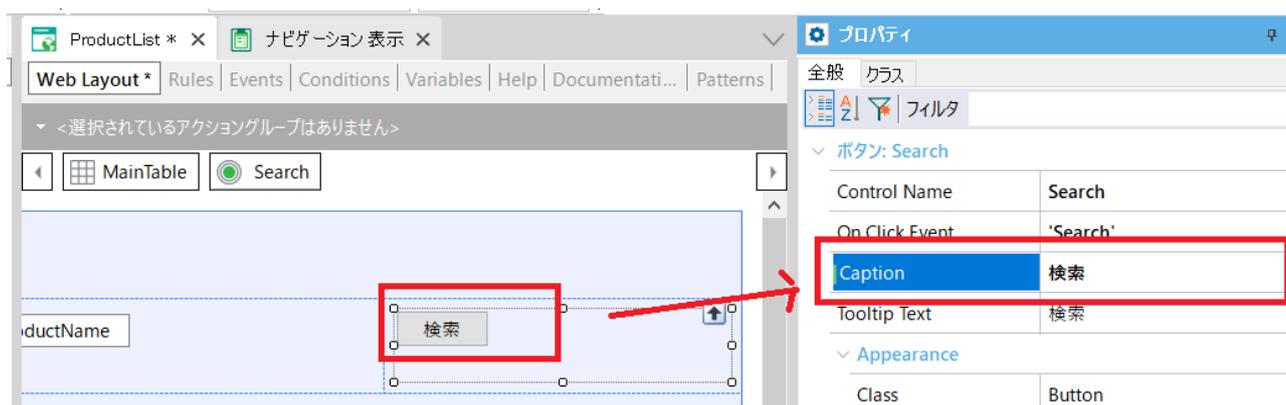
ProductList Web Panel の **Automatic refresh** プロパティを **No** に変更します。



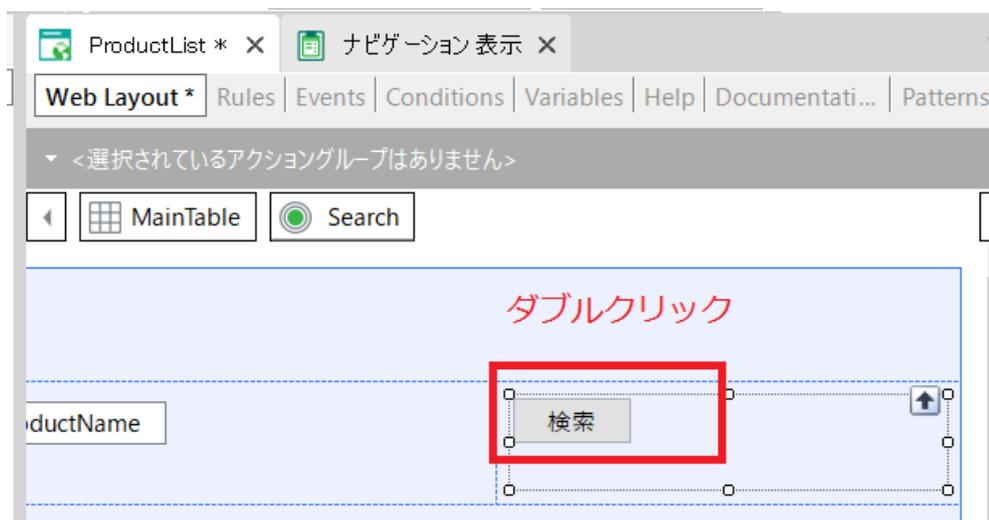
Web Layout タブにて、ツールボックスから **ボタン** をドラッグアンドドロップで Web Form に配置し、その後表示される **ユーザーイベント** を **選択/定義** ダイアログにてイベント名に **Search** を設定します。



配置した Search ボタンの Caption プロパティに実行時に表示されるボタンの文字を変更可能です。



配置した Search ボタンをダブルクリックすることで、自動的に Events タブに 'Search' Event が追加されます。



↓

```

1 | Event 'PDF'
2 |
3 |     ProductListListOutput.Call()
4 |
5 | Endevent
6 |
7 | Event 'Search'
8 |
9 | Endevent

```

追加された'Search' Event にて次のコードを入力します。

- Grid1.Refresh()

```

7 | Event 'Search'
8 |     Grid1.Refresh()
9 | Endevent

```

以上で検索ボタンを押下した際にのみグリッド上のレコード表示を更新することが可能です。

PDF出力

検索ワード

ProductCode	ProductName	ProductPrice	ProductStock	ProductTypeName	ProductPhoto
1	赤リンゴ	100.00	5	フルーツ	

次のステップ

- GeneXus はきわめて広範な開発プラットフォームであり、学ぶべきことが山ほどあります。

次の場所で深く掘り下げて学ぶことができます：

- o GeneXus トレーニングコース：

<https://www.genexus.jp/training>

- o GeneXus Wiki：

<http://wiki.genexus.jp/hwikibypageid.aspx?46071>

開発に役立つサイト

<https://www.genexus.jp/community-and-support-jp/support-overview#link05>