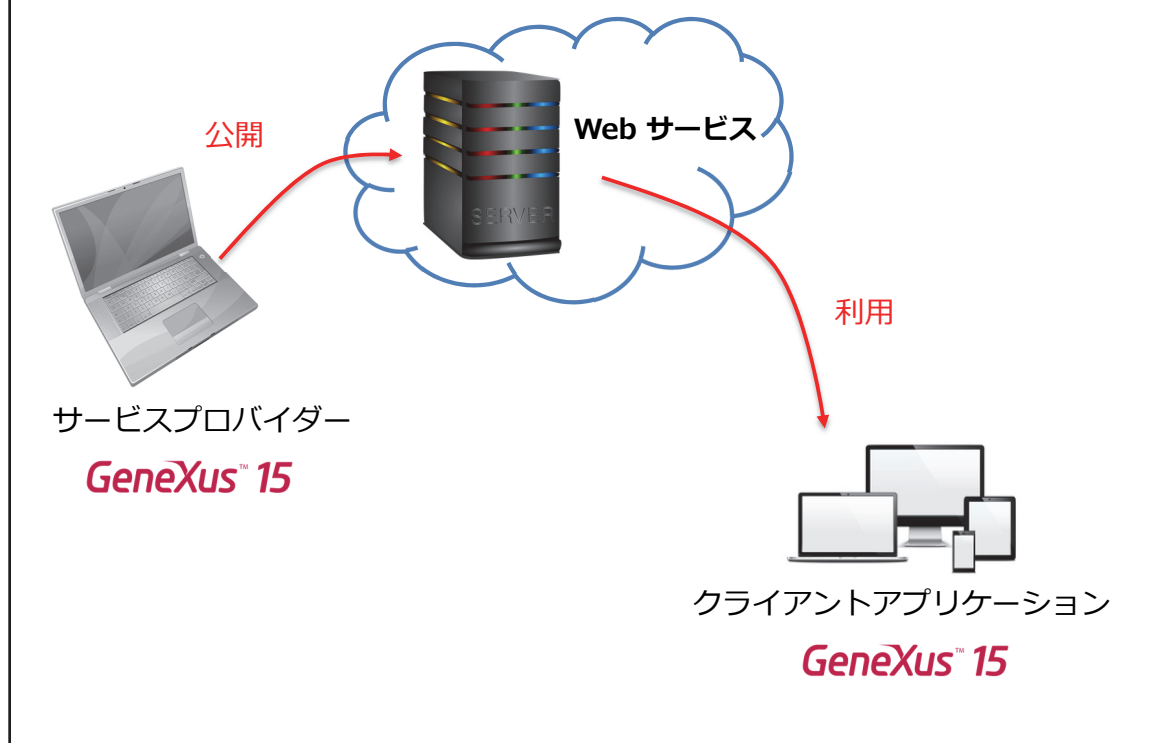










# Web サービス

*GeneXus™ 15*



Web サービスは、便利な機能を提供するプログラムです。  
これらは、通常インターネットなどのネットワークを通じて簡単に見つけたり、呼び出したりできるように Web サーバ上に配置されています。

## 国情報

| 名前     | 首都           | 通貨            | 国旗  |
|--------|--------------|---------------|---|
| ブラジル   | Brasilia     | Brazil Real   |  |
| 中国     | Beijing      | Yuan Renminbi |  |
| フランス   | Paris        | Euro          |  |
| 日本     | Tokyo        | Yen           |  |
| アメリカ   | Washington   | Dollars       |  |
| アルゼンチン | Buenos Aires | Pesos         |  |
| チリ     | Santiago     | Pesos         |  |
| ウルグアイ  | Montevideo   | Pesos         |  |

旅行代理店が新しい機能を要望してきた場合について考えてみましょう。新機能は、顧客にすべての国のリストを提供し、各国についてその首都、通貨、および国旗を表示できる機能です。

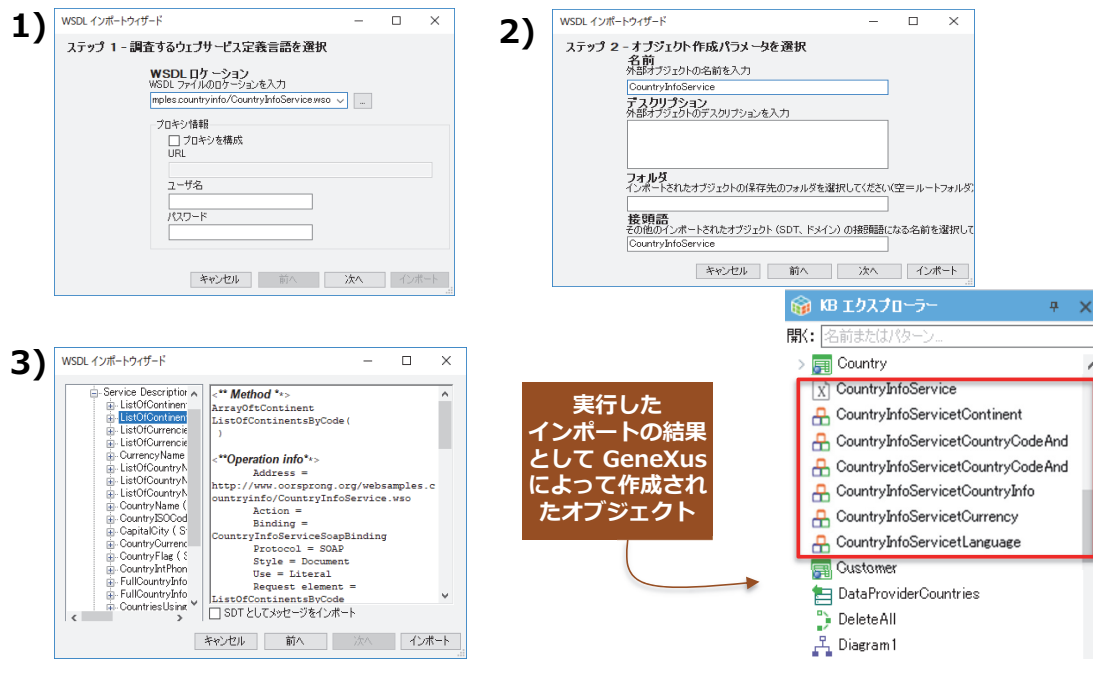
# Web サービス

## CountryInfoService

これを実現するには、検索の結果、公開されていることが判明した特定の Web サービスを呼び出します。これは CountryInfoService という名前のサービスであり、上記以外の情報も返すことができます。

## GeneXus からの Web サービスの使用

[ツール] → [アプリケーションの統合] → [WSDL インポート]



では、GeneXus に戻って、[ツール] メニューから、[アプリケーションの統合] → [WSDL インポート] オプションの順に選択します。

ウィザードが表示され、はじめに、使用する Web サービスの場所を指定する必要があります。

[次へ] をクリックし、既定の情報をそのまま保持して、もう一度 [次へ] をクリックします。

提供されているサービスのリストが表示されます。最後に **[インポート]** をクリックします。

[フォルダ表示] ウィンドウに、実行したインポートの結果として GeneXus によって作成されたオブジェクトがいくつか表示されます。

構造化データのグループと、外部オブジェクト（強調表示）が、インポートした Web サービスの名前とともに表示されます。

## GeneXus からの Web サービスの使用

外部オブジェクト…

Web サービスが提供するメソッド (機能)

**ISO コード!**  
 ウルグアイ - UY  
 ブラジル - BR  
 米国 - US

| メソッド名                                | タイプ                 | Is Collection                       | デスクリプション |
|--------------------------------------|---------------------|-------------------------------------|----------|
| ListOfContinentsByCode               | CountryInfoServi... | <input checked="" type="checkbox"/> |          |
| ListOfContinentsByName               | CountryInfoServi... | <input checked="" type="checkbox"/> |          |
| ListOfCurrenciesByCode               | CountryInfoServi... | <input checked="" type="checkbox"/> |          |
| CurrencyName                         | Character(9999)     | <input type="checkbox"/>            |          |
| sCurrencyISOCode                     | Character(9999)     | <input type="checkbox"/>            |          |
| ListOfCountryNamesByCode             | CountryInfoServi... | <input checked="" type="checkbox"/> |          |
| ListOfCountryNamesByName             | CountryInfoServi... | <input checked="" type="checkbox"/> |          |
| ListOfCountryNamesGroupedByContinent | CountryInfoServi... | <input checked="" type="checkbox"/> |          |
| CountryName                          | Character(9999)     | <input type="checkbox"/>            |          |
| sCountryISOCode                      | Character(9999)     | <input type="checkbox"/>            |          |
| CountryISOCode                       | Character(9999)     | <input type="checkbox"/>            |          |
| sCountryName                         | Character(9999)     | <input type="checkbox"/>            |          |
| CapitalCity                          | Character(9999)     | <input type="checkbox"/>            |          |
| sCountryISOCode                      | Character(9999)     | <input type="checkbox"/>            |          |
| CountryCurrency                      | CountryInfoServi... | <input type="checkbox"/>            |          |
| sCountryISOCode                      | Character(9999)     | <input type="checkbox"/>            |          |
| CountryFlag                          | Character(9999)     | <input type="checkbox"/>            |          |
| sCountryISOCode                      | Character(9999)     | <input type="checkbox"/>            |          |
| CountryIntPhoneCode                  | Character(9999)     | <input type="checkbox"/>            |          |
| sCountryISOCode                      | Character(9999)     | <input type="checkbox"/>            |          |

外部オブジェクトとは、GeneXus のオブジェクトタイプで、アプリケーションと外部リソース間のやりとりを可能にします。この例では、外部リソースは使用する Web サービスです。

外部オブジェクトをダブルクリックすると、Web サービスが提供するすべてのメソッド (機能) が表示されます。

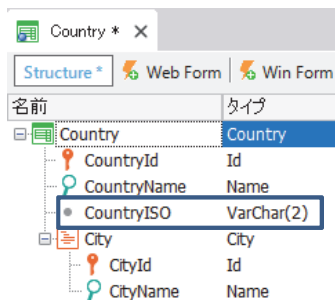
要望では、各国の首都、通貨、および国旗を取得する必要があるので、それぞれ CapitalCity、CountryCurrency、および CountryFlag メソッドを使用します。

ここで呼び出す 3 つのメソッドは、パラメーターとして「国の ISO コード」を受け取ります。これは、国にコードを割り当てる国際標準で、世界中のさまざまな組織、企業、政府によって使用されています。

たとえば、ウルグアイのコードは「UY」、ブラジルは「BR」、米国は「US」です。

## GeneXus からの Web サービスの使用

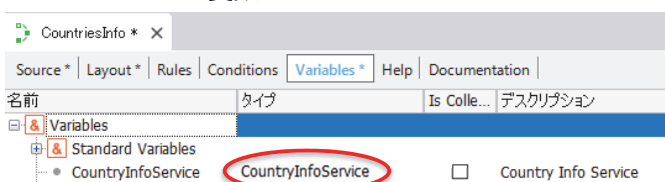
各国に対応する ISO コード  
の値を格納する新しい  
CountryISO 項目属性の追加



「CountriesInfo」：要望された PDF リスト



CountriesInfo の変数:



Web サービスが提供するすべてのメソッドにアクセス可能

Web サービス

ここで、Country トランザクションに移動すると、各国に対応する ISO コードを格納する新しい CountryISO 項目属性を定義したことが分かります。

また、旅行代理店に要望された PDF リストを実装するプロシーチャーは、既に作成を始めています。

このプロシーチャーの [Layout] エレメントには、タイトルを表示する [PBTitle] プリントブロックと、必要な項目属性や変数を追加する [PBCountry] プリントブロックが表示されています。国名を表示する必要があり、これはデータベースにある情報なので、CountryName 項目属性を追加します。

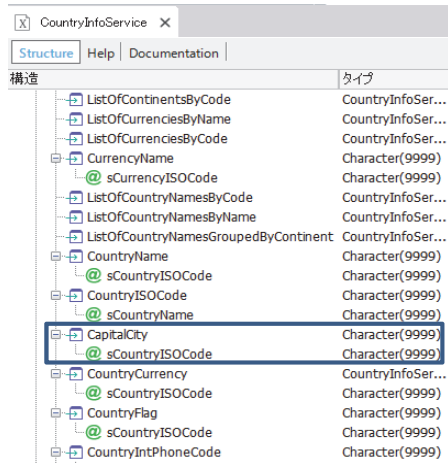
各外部オブジェクトについて、GeneXus は関連するデータタイプを作成します。つまり、Web サービスを使用するために、CountryInfoService タイプの変数を定義し、そのすべてのメソッドにアクセスできるようにします。

## GeneXus からの Web サービスの使用

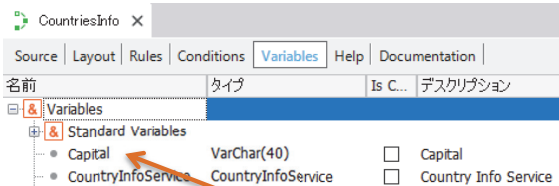
国の首都の取得…

## CapitalCity メソッド

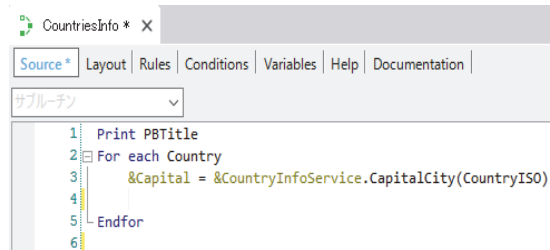
CapitalCity メソッドは、国の ISO コードを受け取り Character データタイプを返します。



ソースの文…



国の首都を受け取る新しい  
Capital 変数



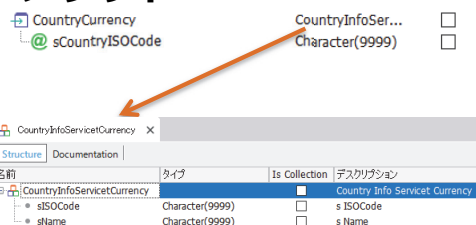
国の首都を取得する方法を理解するところから始めましょう。再度メソッドリストを確認すると、国の ISO コードを受け取って Character データタイプを返す CapitalCity メソッドがあることが分かります。

検索した各国について処理を実行するので、ソースの For each の中に、&Capital 変数、続けて割り当てるために等号を記述します。そして、国の首都を Capital 変数で受け取るために、記号の右側に次の構文を記述します: 変数 &CountryInfoService、ピリオド、実行する Web サービスのメソッド。この例では次のメソッドを記述します: CapitalCity。最後に、パラメーターを使用して、呼び出すメソッドに CountryISO 項目属性を渡します。



## GeneXus からの Web サービスの使用

各国の通貨の取得…

CountryCurrency  
メソッド

新しい変数…

| CountriesInfo X   |                    |                          |                      |
|---|--------------------|--------------------------|----------------------|
| Source   Layout   Rules   Conditions   Variables   Help   Documentation |                    |                          |                      |
| 名前  | タイプ                | Is C...                  | デスクリプション             |
| Variables   |                    |                          |                      |
| Standard Variables  |                    |                          |                      |
| Capital   | VarChar(40)        | <input type="checkbox"/> | Capital              |
| CountryInfoService  | CountryInfoService | <input type="checkbox"/> | Country Info Service |
| Currency  | VarChar(40)        | <input type="checkbox"/> | Currency             |

## SDT

## 「CountryInfoServicetCurrency」

| CountriesInfo * X   |   |  |  |
|---|---|--|--|
| Source *   Layout   Rules   Conditions   Variables   Help   Documentation |   |  |  |
| サブルーティン   |   |  |  |
| 1   | Print PBTitle   |  |  |
| 2   | For each Country  |  |  |
| 3   | &Capital = &CountryInfoService.CapitalCity(CountryISO)            |  |  |
| 4   | &Currency = &CountryInfoService.CountryCurrency(CountryISO).sName |  |  |
| 5   |   |  |  |
| 6   | Endfor  |  |  |

ソースの文…

Web サービス

続いて、各国の通貨を取得します。CountryCurrency メソッドもパラメーターとして国の ISO コードを受け取りますが、CountryInfoServicetCurrency データタイプを返します。

CountryInfoServicetCurrency データタイプの定義は、ナレッジベースで作成されたこの SDT です。

構造を見てみましょう。通貨コードと通貨名という 2 つの単純な項目で構成されているのが分かります。

通貨名、つまり、sName フィールドにロードされた値を取得する必要があります。プロシージャーに Currency という名前で、VarChar データタイプの新しい変数を定義します。

[Source] エlementに移動し、次の文を定義します：

&Currency = &CountryInfoService.CountryCurrency(CountryISO).sName

## GeneXus からの Web サービスの使用

国旗の取得…

CountryFlag  
メソッドCountryFlag  
CountryISOCodeCharacter(9999)  
Character(9999)新しい変数:  
&FlagChar と &FlagImage

| CountriesInfo ×   |                    |                          |                      |
|---|--------------------|--------------------------|----------------------|
| Source   Layout   Rules   Conditions   Variables   Help   Documentation |                    |                          |                      |
| 名前  | タイプ                | Is C...                  | デスクリプション             |
| Variables   |                    |                          |                      |
| Standard Variables  |                    |                          |                      |
| Capital   | VarChar(40)        | <input type="checkbox"/> | Capital              |
| CountryInfoService  | CountryInfoService | <input type="checkbox"/> | Country Info Service |
| Currency  | VarChar(40)        | <input type="checkbox"/> | Currency             |
| FlagChar  | VarChar(40)        | <input type="checkbox"/> | Flag Char            |
| FlagImage   | Image              | <input type="checkbox"/> | Flag Image           |

| CountriesInfo ×  |   |
|--|---|
| Source*   Layout   Rules   Conditions   Variables   Help   Documentation |   |
| サブルーチン   |   |
| 1  | Print PBTitle   |
| 2  | For each Country  |
| 3  | &Capital = &CountryInfoService.CapitalCity(CountryISO)            |
| 4  | &Currency = &CountryInfoService.CountryCurrency(CountryISO).sName |
| 5  | &FlagChar = &CountryInfoService.CountryFlag(CountryISO)           |
| 6  | &FlagImage.FromURL(&FlagChar)                                     |
| 7  |   |
| 8  | EndFor  |

ソースの文…

Web サービス

次に、国旗を取得します。

CountryFlag メソッドも、国の ISO コードを受け取って Character データタイプを返します。

プロシージャーにもう 1 つ、FlagChar という名前で VarChar タイプの変数を定義します。

[Source] エlementに移動して、&CountryInfoService 変数、ドットと入力します。次に CountryFlag メソッドを選択し、国の ISO コードを渡します。変数 &FlagChar で結果を受け取ります。

ここで、次のようにコードを分析します。

&FlagChar 変数に、CountryInfoService Web サービスの CountryFlag メソッドが返す文字列の値を格納しましたが、国旗の画像を表示するため、この文字データを画像データに変換します。

もう一度、プロシージャーの [Variables] Elementに移動し、今度は、Image データタイプの FlagImage という変数を定義します。

[Source] Elementで、次の文を定義します：

&amp;FlagImage.FromURL(&amp;FlagChar)

この文で、&amp;FlagChar 変数の文字列を画像に変換します。

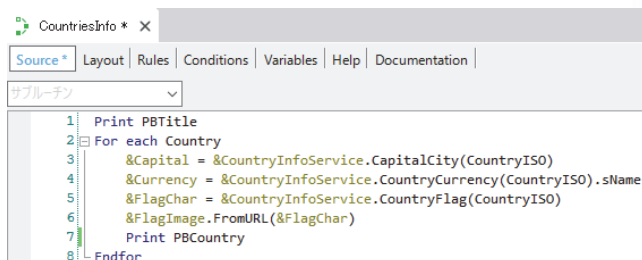
FromURL メソッドは、パラメーターでパスの文字列を渡します。Image タイプのすべての変数と項目属性に適用することができます。

## GeneXus からの Web サービスの使用







## PBCountry プリントブロックへの変数の追加…



## ソースの完成 …



## 実行時のリスト表示…

| 国情報    |              |               |   |
|--------|--------------|---------------|---|
| 名前     | 首都           | 通貨            | 国旗  |
| ブラジル   | Brasilia     | Brazil Real   |  |
| 中国     | Beijing      | Yuan Renminbi |  |
| フランス   | Paris        | Euro          |  |
| 日本     | Tokyo        | Yen           |  |
| アメリカ   | Washington   | Dollars       |  |
| アルゼンチン | Buenos Aires | Pesos         |  |

ここまでで、表示すべきすべての変数に対応する値をロードしました。[Layout] エlementに移動して、これらの変数を [PBCountry] プリントブロックに追加します。

これで、実行時にリストを表示する準備ができました。

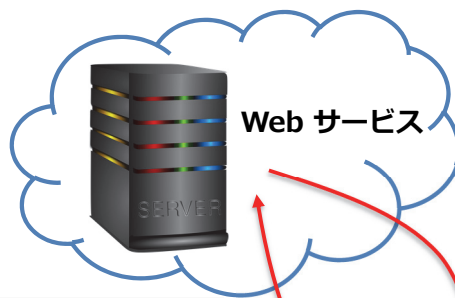
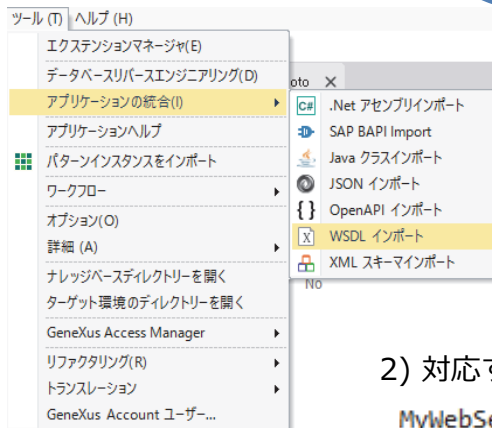
ところで、出力形式が PDF になるようにプロパティを宣言する必要がありました。

そこで、[Main Program] プロパティを [True] の値、[Call Protocol] プロパティを [HTTP] の値に設定して、PDF リストを生成する Output\_file ルールを宣言します。

ここで、プロシージャーのタブを右クリックして、[実行] を選択します。すると、すべての情報のリストが表示されます。

## 呼び出し

- 1) ウィザードが Web サービスに対応する GeneXus オブジェクトを作成



クライアントアプリケーション

- 2) 対応する GeneXus オブジェクトを呼び出し:

`MyWebService.ServiceName(parameters)`

サービスプロバイダーがサーバー上で Web サービスを「公開」し、クライアントアプリケーションがその Web サービスを呼び出して「利用」します。GeneXus では、Web サービスを公開するアプリケーションに加え、任意のプログラミングツールやプラットフォームで開発された Web サービスを利用するアプリケーションの開発も可能です。

公開されている Web サービスにアクセスするには、その場所が分かっている、WSDL (Web サービス記述言語) で記述された定義をインポートする必要があります。その後、Web サービスの機能にアクセスできます。

Web サービスを GeneXus アプリケーションに統合するには、[ツール] → [アプリケーションの統合] を選択し、[WSDL インポート] を選択し、ウィザードの手順に従います。最終的に、GeneXus によって外部オブジェクトと呼ばれる特殊なオブジェクトタイプが自動的に作成され、Web サービスに関連付けられます。また、データの管理に必要な構造化データタイプも作成されます。

その後、Web パネルやプロシージャといった別のオブジェクトから、必要なパラメーターを渡すことで、外部オブジェクトと必要なサービスを呼び出します。

公開



- Procedure オブジェクト
- ビジネスコンポーネント
- Data Provider オブジェクト

| プロパティ                     |                |
|---------------------------|----------------|
| フィルタ                      |                |
| ▼ Procedure: MyWebService |                |
| Name                      | MyWebService   |
| Description               | My Web Service |
| Module/Folder             | Root Module    |
| Main program              | True           |
| Call protocol             | SOAP           |
| Execute in new LUW        | False          |
| Qualified Name            | MyWebService   |
| Object Visibility         | Public         |

WSDL 表示方法: オブジェクト URL の最後に「?WSDL」を加えてください  
(オブジェクト名の最初に [a] がついていることに注意します)。

<http://localhost/travelagency/amywebservice.aspx?WSDL>

Web サービスを公開するには、Procedure オブジェクトを作成し、提供する機能を設定し、プロパティで次の設定を行います。[Main program] プロパティの値を [True]、[Call protocol] プロパティを [SOAP] にします。コンパイルすると、このプロシージャを (GeneXus で構築されたかどうかを問わず) 任意のアプリケーションで利用できるようになります。

ビジネスコンポーネントや Data Provider オブジェクトを Web サービスとして公開することもできます。GeneXus で作成した Web サービスのプロトコルは、SOAP または REST です。

SOAP で公開した Web サービスの場所を確認するには、サービスを実行し、ブラウザーの URL の末尾に「?WSDL」を付加します。

URL 例:

<http://localhost/travelagency/amywebservice.aspx?WSDL>

Web サービスの使用に関する詳細情報:

<http://wiki.genexus.com/commwiki/servlet/wiki?4210> (英語のみ)

GeneXus での Web サービスの使用方法の詳細については、次の wiki リンクにアクセスしてください:  
<http://wiki.genexus.com/commwiki/servlet/wiki?4210> (英語のみ)