

モジュールと外部オブジェクト

*GeneXus*TM

ここでは、機能をカプセル化し、ナレッジベース内のオブジェクトを整理する際に役立つ、いくつかの GeneXus オブジェクトについて考えてみます。

モジュール

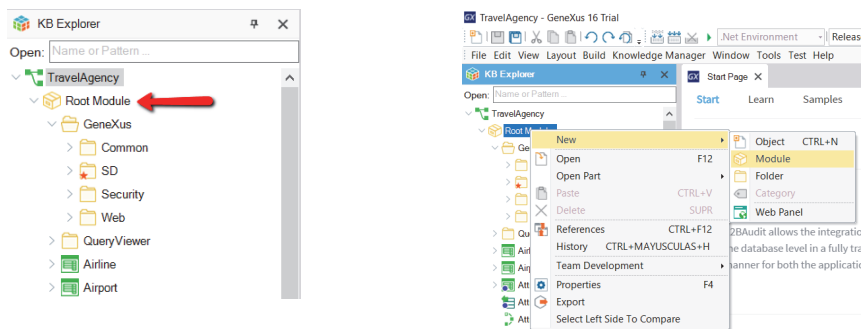
まずモジュールについて見ていきましょう。

KB1**KB2**

モジュールはコンテナとして機能する GeneXus オブジェクトで、ナレッジベース内のオブジェクトをグループ化して分かりやすくしたり、メンテナンスしやすくしたりします。また、オブジェクトをほかのナレッジベースと統合するのにも役立ちます。

モジュールとは

- ナレッジベース内のオブジェクトをグループ化し、その機能をカプセル化するための、GeneXus オブジェクトである。
- オブジェクトに対する理解、オブジェクトのメンテナンス、複数のナレッジベースのオブジェクトの統合が容易になるように設計されている。
- モジュールやフォルダを使用して階層を構築できる。



ナレッジベースを作成すると、KB エクスプローラーに Root Module が作成され、既定では、作成するすべてのオブジェクトがそのモジュール内に保持されます。

モジュールとフォルダは、どちらもオブジェクトを整理するのに役立ちます。ただし、モジュールとフォルダには概念的な違いがあります。モジュールは、ナレッジベースの各部をカプセル化したりモジュール化したりするのに役立ちます。また、どのオブジェクトをほかのオブジェクトから見えるようにするかを決めるのにも役立ちます。これについては後で説明します。

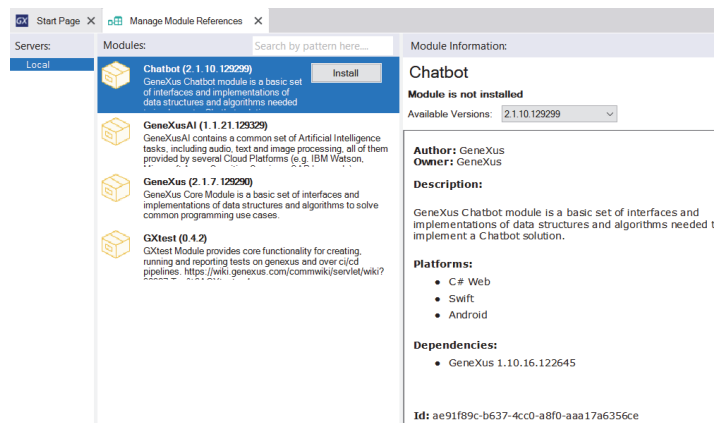
一方、フォルダは、特定の基準に基づいてオブジェクトを分離して整理するためのコンテナとして機能します。フォルダはモジュールとともに階層ツリーを構成します。そのルートは必ず、Root Module など、1 つのモジュールになります。これは KB エクスプローラーで確認できます。

モジュールには子モジュールが存在する場合がありますが、フォルダがモジュールを子として持つことはありません。通常、モジュールはカプセル化に、フォルダはモジュール内のオブジェクトの整理に使用できます。

オブジェクトをモジュールに追加するには、KB エクスプローラーで該当するオブジェクトを目的のモジュールにドラッグするか、目的のモジュールを右クリックして [新規オブジェクト] を選択します。また、該当するオブジェクトの [Module/Folder] プロパティの値を変更して追加することもできます。

作成済みモジュールのナレッジベースへの追加

- [ナレッジマネージャ] > [参照モジュールの管理]



共有されているパッケージ済みモジュールは、[ナレッジマネージャ] > [参照モジュールの管理] メニューで表示できます。

利用可能なモジュールそれぞれの情報を見て、ナレッジベースにインストールするかどうかを判断できます。インストールすると、KB エクスプローラーの [参照モジュール] ノードに保存されます。これに対して、作成したオブジェクトは、既定で Root Module に保存されます。

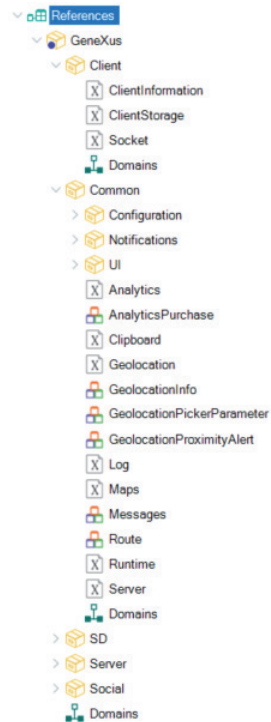
コンパイル済みのモジュールのオブジェクトは変更できません (読み取り専用になります)。そのため、F5 キーを押すときに、指定したり生成したりする必要はありません。

ただし、アプリケーション内のオブジェクトから自由に使用でき、利用可能なすべての機能を使用できます。

コミュニティのメンバーは、自分でモジュールを作成、共有できるほか、Marketplace を介して販売することもできます。

このようなモジュールの 1 つに、GeneXus Core と呼ばれる GeneXus モジュールがあります。

GeneXus モジュール



GeneXus モジュールは自動的に配布され、すべてのナレッジベースにインストールされます。アプリケーションの外部モードでは、[参照モジュール] ノードに保存されます。

API のセットと、それらに対応するドメインおよび SDT が含まれる一連のサブモジュールで構成されており、多様なテクノロジー、デバイス、センサー、アプリケーションなどの呼び出しが可能になります。

API は外部オブジェクトとして実装されます。次は、これについて説明します。

モジュールに関する詳細情報

<http://wiki.genexus.jp/hwikibypageid.aspx?22411>

モジュールオブジェクトの詳細については、Wiki を参照してください:
<http://wiki.genexus.jp/hwikibypageid.aspx?22411>

外部オブジェクト

次に、外部オブジェクトについて見てみましょう。

外部オブジェクトは、ナレッジベースの外部リソースに、ナレッジベース内の別のオブジェクトであるかのようにアクセスできる GeneXus オブジェクトです。

こうした特性を持つため、Web アプリケーションやモバイル デバイス アプリケーションで利用されるケースが増え、その重要性が増しています。次に、その使用方法を見てみましょう。

ナレッジベースの外部オブジェクトとしてのリソース:

- プログラミング言語で生成およびコンパイルされるネイティブオブジェクト:
 .NET のアセンブリ (.dll)
 Java クラス (.class)
- 別の外部ソースに格納されているリソース:
 JEE サーバーの Enterprise Java Beans (EJB)
 DBMS 上に作成されているストアードプロシージャ
 Web サーバーで公開されている Web サービス (WSDL、OpenAPI)
 SAP BAPI モジュール
 JSON ファイル
 XML スキーマ
- GeneXus で利用可能な外部オブジェクト:
 スマートデバイス向け API: ハードウェア (カメラ、GPS、マイクなど) およびソフトウェア (カレンダー、連絡先、通知など) へのアクセス
 イベントへのアクセス、サーバー/メモ帳/マップ/ソーシャルネットワークなどとの通信に使用する API
- GeneXus Marketplace で公開されている外部オブジェクト

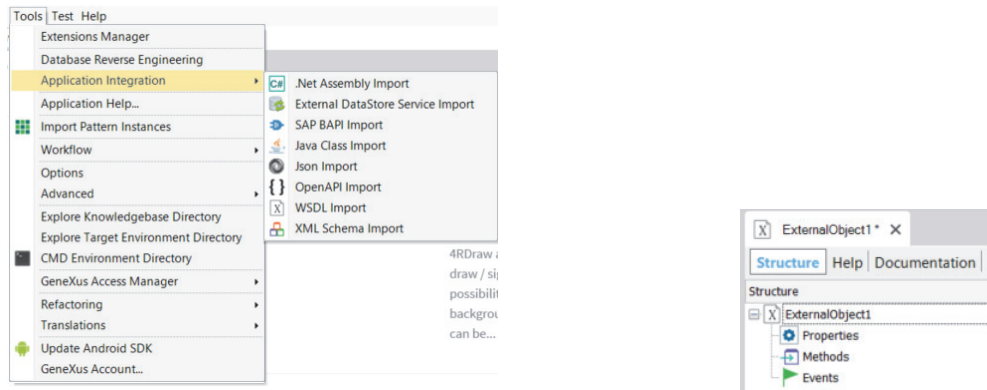
ナレッジベースには、さまざまなタイプのリソースをインポートできます。たとえば、.NET でプログラミングされたものがある場合に、DLL を生成し、それを外部オブジェクトとしてナレッジベースにインポートできます。その DLL に含まれている関数を、GeneXus でプログラミングされたプロシージャであるかのように、アプリケーションから呼び出すことができます。Java で作成されたクラスについても同じです。

ほかの外部ソースに格納されているリソース (Java Beans プログラムなど)、データベースに格納されているプロシージャ、Web サービス (SOAP および REST)、SAP モジュール、任意のアプリケーションで生成された JSON ファイル、XML スキーマをインポートすることも可能です。

GeneXus では、Root Module または GeneXus モジュールに格納されている一連の外部オブジェクトも提供され、それらを使用してさまざまなリソースにアクセスできます。たとえば、ハードウェアやモバイルデバイスのネイティブアプリケーションと相互作用するための API、サーバー、イベント、Windows アプリケーション (メモ帳など)、マップやソーシャルネットワークを使用するための外部サイトにアクセスするための API などのリソースです。

さらに、GeneXus Marketplace で公開されている外部オブジェクトをアプリケーションに含めることもあります。

ウィザードを使用した外部オブジェクトの作成

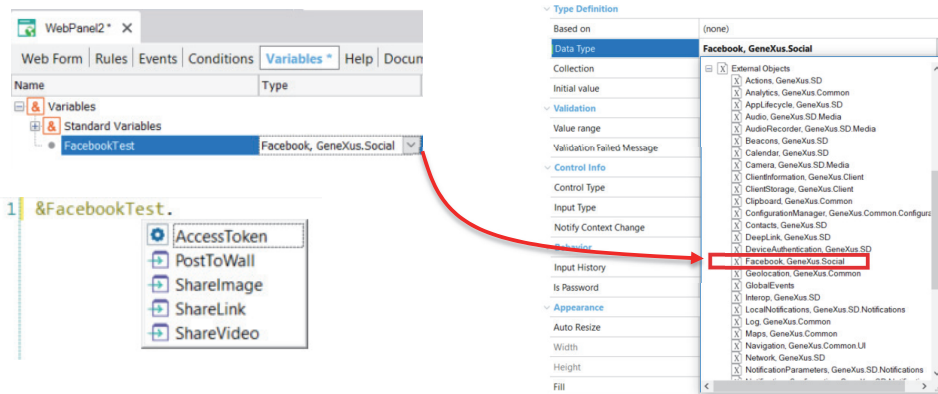


外部オブジェクトを作成する最も効果的な方法は、ウィザードを使用することです。[ツール] メニューで [アプリケーションの統合] を選択すると、インポート可能なさまざまなリソースが表示されます。また、リソースの実行に適したウィザードが実行されます。ウィザードを完了すると、作成した外部オブジェクトにリソースが自動的に関連付けられます。インポートされたタイプまたはリソースに基づいて、外部オブジェクトのすべてのプロパティが調整されます。

ほかの GeneXus オブジェクトの場合と同様に、[新規オブジェクト] を使用して外部オブジェクトを作成することもできます。その場合は、プロパティ、メソッド、およびイベントを手動で設定する必要があります。

外部オブジェクトの使用方法

- 1) 外部オブジェクトタイプの変数を作成する
- 2) メソッドを呼び出すか、利用可能なプロパティを割り当てる



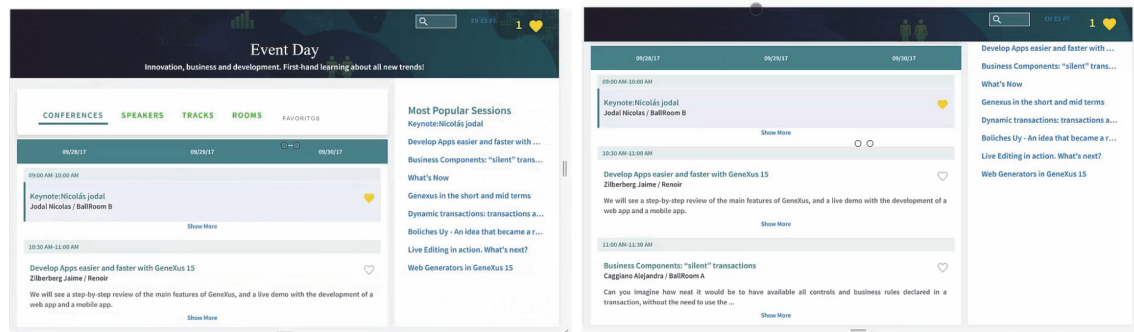
使用したい外部オブジェクトに対応するプロパティに基づいて外部オブジェクトを作成すると、ナレッジベース内のほかのデータタイプと同様に使用できるようになります。

ほかのタイプの拡張データと同じように、タイプの変数を定義し、メソッドを呼び出すか、必要なプロパティを設定するか、その両方を行うかして使用します。

外部オブジェクトの詳細については、画面に示すリンク

<http://wiki.genexus.jp/hwikipageid.aspx?5669> を参照してください。

JavaScript で外部オブジェクトを使用する例



外部オブジェクトを使用して JavaScript とやり取りすることもできます。たとえば、外部 JavaScript に実装されているイベントを GeneXus イベントとリンクさせることができます。

サンプルアプリケーションでこの処理を見てみましょう。

アプリケーションをスクロールすると、上部のバーが小さくなることに注目してください。

この機能は、外部でプログラミングされた `JavaScriptChangeOnScroll` イベントで実装されます。この JavaScript と GeneXus をリンクさせる方法を見てみましょう。

JavaScript で外部オブジェクトを使用する例 (続き)

The screenshot displays the GeneXus IDE interface. On the left, a code editor shows a JavaScript file named 'changeonscroll.js' with the following content:

```

1 var changeonscroll = {
2   shrinkOnHeight: 30
3 };
4
5 $(function() {
6   $(window).on('scroll', function() {
7     var distanceY = window.pageYOffset || document.documentElement.scrollTop;
8     var shrinkOn = changeonscroll.shrinkOnHeight;
9     if (distanceY > shrinkOn) {
10      gx.fx.obs.notify("changeonscroll.scrollToShrink");
11    }
12    else {
13      gx.fx.obs.notify("changeonscroll.scrollToExpand");
14    }
15  });
16 });

```

On the right, the 'Structure' pane shows the 'ChangeOnScroll' object. It has a 'Type' of 'Native Object' and a 'JavaScript External' reference to 'changeonscroll'. The 'Events' section lists 'ScrollToShrink' and 'ScrollToExpand'. The 'ScrollToShrink' event is linked to the 'ExternalObjectEvent: ScrollToShrink()' method.

Below the code editor, the 'Event Start' configuration for the 'ScrollToShrink' event is shown. It includes the following code:

```

Form.HeaderRawHTML = !"<link href='https://fonts.googleapis.com/css?family=SourceS
Form.HeaderRawHTML += GetChangeOnScrollScript()
changeonscroll.shrinkOnHeight = 20

```

この JavaScript は非常にシンプルなものです。スクロールが特定の高さに達すると Shrink イベントをトリガーし、そうでない場合は Expand イベントをトリガーします。

この JavaScript を GeneXus にリンクさせるには、どうしたらよいでしょうか。

これを行うには ChangeOnScroll という外部オブジェクトを使用します。この外部オブジェクトは、基本的に、Changeonscroll という外部 JavaScript と関連付けられています。ここに示すのが、この JavaScript がトリガーするイベントです。ここでは JavaScript と同じ名前になっていますが、名前を変えることもできます。

これらのイベントが実装された Web マスターパネル (マスターページ) に、外部オブジェクト ChangeOnScroll が、ScrolltoExpand イベントと ScrolltoShrink イベントとともに含まれています。

一方ではコンポーネントを非表示に、もう一方では表示にしています。これは、スクロールバーの位置に応じて表示したり非表示にしたりするためです。

JavaScript をアプリケーションに含めるには、これまでと同じように、スクリプトを HTML コードに追加します。

外部オブジェクトの詳細:

<http://wiki.genexus.jp/hwikibypageid.aspx?5669>

外部オブジェクトの詳細については、画面に示すリンクを参照してください。