

GeneXus™

Grow thru knowledge



GeneXus X

クイックスタート

2011年7月

ジェネクス・ジャパン株式会社

〒141-0022 東京都品川区東五反田 5 丁目 10-18、TK 五反田ビル 3F

電話番号 : 03-5793-5481 FAX : 03-5793-5484

<http://www.genexus.jp>

Copyright · Artech Consultores S. R. L. 1988-2008.

All rights reserved. This document may not be reproduced by any means without the express consent of Artech Consultores S.R.L. The information contained herein is intended for personal use only.

Registered Trademarks

Artech and GeneXus are trademarks or registered trademarks of Artech Consultores S.R.L. All other trademarks mentioned herein are the property of their respective owners.

目次

序論	3
GeneXus トライアルバージョン	3
インストールとオーソライズ	4
はじめに:ステップバイステップ チュートリアル	8
パート 1 : GeneXus の紹介	9
パート 2 : 最初のアプリケーション	10
パート 3 : 保守	35
パート 4 : 機能の追加	38
パート 5 : インタラクティブなダイアログ	40
パート 6 : アプリケーションの改良	45
パート 7 : プロシージャ	48
パート 8 : マルチプラットフォーム	55
要約	56
FAQ	57
コンタクトとリソース	60

序論

この文書は、GeneXus が提唱するナレッジベース開発方法論の潜在能力を、その主要な機能を経験しながらあなたが発見するのを援助することが目的です。

- データモデルの自動設計
- コードの自動生成
- データベースとコードの自動的な保守
- マルチプラットフォーム開発と実装

GeneXus トライアルバージョン

機能制限

GeneXus トライアルバージョンは全機能を網羅しています。利用できるジェネレータ(.Net+SQL Server)はシングルサイトキーでオーソライズすることができます。キーは、起動してから 60 日後に期限切れになりますが延長は可能です。(再オーソライズ)

しかしながら、ナレッジベースに作成できるGeneXusオブジェクトと項目属性の最大数には制限が掛かっています。

- 項目属性の最大数は 90 個
- オブジェクトの最大数は 140 個

ライセンス制限

GeneXus トライアルバージョンは、評価とトレーニングの目的のみに使用することが許可されています。ライセンスは、シングルユーザのローカルインストールのみとなります。

ライセンス 同意

GeneXus X トライアル版ライセンス (知的財産) の利用規約

1. この合意は、**GeneXus X トライアル版**と知識ベースに基づき自動的に設計、生成、保守されるデータベースやアプリケーションを対象とします。
2. **Artech Consultores S.R.L. (これ以降は Artech)** は以下のことを宣言しユーザは同意します。**GeneXus**の知的な生成物、**GeneXus, GXflow, GXplorer, GXportal, GXquery** そして **Artech**の名前やロゴ、その他**GeneXus**に關係して発表するものは、それらが登録されているいかに係わらず **Artech**の資産です。この合意は、直接的/間接的を問わず、いかなる所有権の移動を意味しませんし、ユーザに本件のライセンスを譲渡する権利も与えません。
3. ユーザは、GeneXus X トライアル版を使うことで得るGeneXusがベースとなったアイデアや技術は公開せず、自分の利益のためだけに使うことに同意します。特に、ユーザはそのコードを解釈するためにリバースエンジニアリング技術を使用しない、また他の人にもそうさせないことに同意します。
4. ユーザはライセンスを複製したり、GeneXusの認可されない使用に対するプロテクションの仕組みを無効にしたりすることを決して行ってはなりません。Artechは、これらのメカニズムを修正したりいつでも新しいものを加える権利を保有します。
5. Artechは以下の状況の下で全ての条件に従うことを前提に、ツール (GeneXus X トライアル版) の使用の許可をユーザに与えます：

- a) 各々のライセンスは、1 台のパソコンで使用します。ユーザはどのライセンスも複数のパソコンで同時に利用しないことに同意します。

- b) このライセンスは排他的ではありませんが、移転することはできません。GeneXus X トライアル版で生成したプログラムを、営利目的のために使うことはできません。
 - c) ユーザは個人的な利用のため、または評価目的のためだけに、GeneXusプログラムとドキュメンテーションを使用します。プログラムとドキュメンテーションの何れもコピーを第三者に提供しないことに同意します。
 - d) Artechは暗黙的にも明示的にも決して、GeneXus X トライアル版の使用に関して直接的、あるいは間接的な付帯的な損害賠償に対しても責任を負いません。
 - e) ユーザは、教材で示される指示と仕様に従ってライセンスを使用することに同意します。
 - f) GeneXus X トライアル版（全体または一部の）のどんな商用利用も禁じられています。この種のアプリケーションを開発するためには、ユーザは製品版を購入しなければなりません。
 - g) GeneXus X トライアル版は、60 日間の期間限定で.NETジェネレータとSQLデータベースのサポートだけが提供されます。また最大で、90 の項目属性と 140 のオブジェクトまで作成することができます。
- 6.このライセンスが必要なソフト製品をインストールすることは、あなたがすべての契約条件を受け入れたことを意味します。

テクニカルサポート

GeneXusに関するテクニカルサポートについては、以下のメニューが用意されています。

必要に応じてサービスを選択し、ご利用ください。

1. GeneXusに関する総合的な技術情報については、以下のWikiをご利用ください。

[http://wiki.genexus.jp/hwiki.aspx?Wiki+Home,](http://wiki.genexus.jp/hwiki.aspx?Wiki+Home)

2. Wikiを利用し、それでも問題が解決しない場合は、以下のサポートフォーラムをご利用ください。

<http://www.genexus.jp/gxpsites/hgxpp001?6,7,94,O,J,0,MNU;E;19;5;MNU>

3. トレーニングサービス及び資料に関する情報は、以下のサイトをご確認ください。

<http://www.genexus.com/community-and-support-jp/training-313?ja>

<http://www.genexus.com/40/-317?ja>

さらなる情報が必要な場合は、お近くの代理店にお問い合わせください。

<http://www.genexus.com/company-jp/sub-distributors-in-japan?ja>

インストールとオーソライズ

インストール要件

GeneXusトライアルバージョンは以下の製品を含みます：

- **GeneXus モデラー**

Integrated Development Environment (IDE) は、「統合開発環境」です。それは、アプリケーションが使用する稼働プラットフォームに関係なくビジネスアプリケーションを開発及び保守することができます。

- **GeneXus ジェネレータ**

GeneXusは、市場の主要なプラットフォーム向けにネイティブコードを生成します。

GeneXus が対応しているジェネレーター一覧に関しては、

GeneXusトライアルバージョンに提供されるジェネレータは.NETのみになります。

GeneXus及びGeneXusで作成されたアプリケーションを実行するために必要なハードウェアとソフトウェア要件は以下のとおりです。

ハードウェア要件	プロセッサ：最低 1 GHz（マルチコア対応プロセッサをお勧めします）
	メモリー：最低 1 GB（2 GB 以上の容量をお勧めします）
	ハードディスク：インストールするために最小限 300 MBの空き容量が必要です。 GeneXusアプリケーションを作成するためには、知識ベース及び生成するコードのためにさらなる空き容量あるいは共有ディスクが必要です。
ソフトウェア要件	Microsoft Windows XP Professional SP2 以降の SP 適用済み Microsoft Windows Vista 各エディション(※1) Microsoft Windows Windows7 各エディション(※1)
	Microsoft .NET Framework 3.5 SP1 もしくは以降のバージョン
	Microsoft Internet Information Service 5.1 もしくは以降のバージョン(※2)
	Microsoft SQL Server 2005 あるいは 2008(Express, Standard あるいは他のエディション) (※3)
	Adobe Acrobat Reader10 もしくは以降のバージョン

(※1)OS 要件につき、IIS 機能がインストールできるエディションに限ります。

(※2)IIS7.0 以降のバージョンについては、「IIS6 と互換性のある管理」機能を有効化して下さい。

(※3)Microsoft SQL Server Express Edition は無料であり、再配布可能なバージョンです。

アプリケーションデータベースを作成して、生成されたアプリケーションを実行するために、下記の DBMS を必要とします：

- Microsoft SQL Server 2005 あるいは 2008(Express, Standard あるいは他のエディション)。

GeneXus トライアルバージョンをインストールする端末に Microsoft SQL Server がインストールされていない場合は、GeneXus トライアルバージョンのインストーラが自動的に Microsoft SQL Server 2005 をインストールし、Microsoft SQL Server のシステムアドミニストレータユーザー「sa」のパスワードは「genexustrial」で設定されます。

オーソライズ

1. GeneXus トライアルバージョンのセットアップファイル(genexusxtrial.exe)を実行してください。Windows スタートメニューから、あるいはトライアルバージョンの CD で対応するオプションを選択してください。

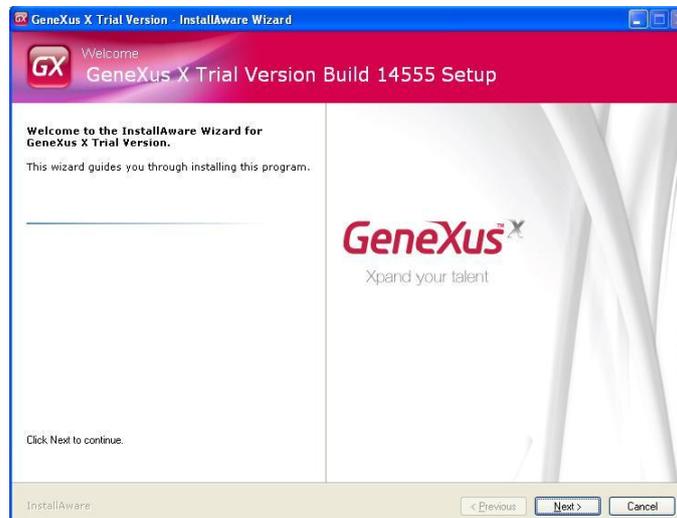


図 : GeneXus X トライアルバージョンのインストールウィザード

1. インストールウィザードに表示される手順に従ってインストールを行ってください。
2. GeneXus トライアルバージョンを初めて使用する際には、実行するためにこの製品をオーソライズしなければなりません。オーソライズは、オンラインもしくは電子メールで行うことが可能です。インターネットアクセスがあるならば、オンラインでのオーソライズをお勧めします。



図 : GeneXus トライアルバージョンを登録するためのダイアログボックス

3. 必須項目を入力してください：会社名と電話番号は任意ですが、名前、電子メールと国名は必須です。
4. 一度**オンラインでのオーソライズ**を選び全ての必須のデータを完備したならば、GeneXus トライアルバージョンは即座に 60 日の期間中に利用することができます。
5. もし**電子メール**オプションを選ぶならば、名前と国名を入力しなければなりません。
会社名と電話番号の項目に関しては、任意です。
入力したデータを確認し登録したデータが正しければ、“CONTINUE”を押してください。入力したデータを変更するためには、“BACK”を押してください。
このオーソライズ方法は Outlook がデフォルトの電子メールクライアントであることを想定しています。メールを送付していただくと、ライセンスマネージャーが生成するサイトコードを記載した電子メールを生成し、あなたの電子メールアドレスに**サイトキー**が送られます。



Grow thru knowledge

Outlook がデフォルト電子メールクライアントでない場合は、**手動**オーソライズで、どんな電子メールアドレスからでもライセンスマネージャーが生成したデータを送ることができます。

サイトキーは、送付していただいた際と同じ電子メールアドレスに送られます。Authorization Required 画面のサイトキー項目に**サイトキー**をコピー&ペーストしてから**続行**をクリックします。

これで GeneXus トライアルバージョンを使い始めることができます！

GeneXus トライアルバージョンを起動させるために支援が必要な場合、サポートフォーラムをご利用ください。

<http://www.genexus.jp/gxpsites/hgxpp001?6,7,94,O,J,0,MNU;E;19;5;MNU>

はじめに:ステップバイステップ チュートリアル

このチュートリアルの目的は、GeneXus の初体験をしていただきながら GeneXus 方法論の基礎を学んでいただくことです。やがてあなたが考えているよりも早く、複雑なビジネスアプリケーションを作成することができるようになります。

このステップバイステップの演習において、単純なビジネスアプリケーションを作成し、GeneXus トライアルバージョンがサポートするプラットフォーム (.Net+SQL Server) 上でそれを実装します。

GeneXus がサポートしているその他のプラットフォーム一覧に関しては、<http://www.genexus.com/technologies/> をご覧ください。

このチュートリアルに取り組みながら、GeneXus の主な機能を経験していきます：

- ナレッジベースアプリケーション設計
- 知的なデータベース生成
- 自動的なコード生成
- 十分に機能的なアプリケーション
- インクリメンタル開発及びアプリケーションの自動的な保守
- ノンインタラクティブなプロセス設計(プロシージャ)
- マルチプラットフォームに向けた開発

使用されるアイコン

	このアイコンは重要な概念であることを意味します。
	このアイコンはヒントを意味します。このフレームの内にあるテキストを読むことをお勧めします。
	このアイコンは実習に必要な目安となる時間を意味します。
	このアイコンは、注釈を意味します。

パート 1 : GeneXus の紹介



推定された時間 : 5 分

GeneXus とは何か？

簡潔にまとめると、GeneXus はプログラムを作成するプログラムです。

これはユーザ要件から始まるツールであり、我々が**ナレッジベース（知識ベース）**と呼ぶものに知識をカプセル化します。

GeneXus はその知識を体系化し、その後自動的にデータベース及びプログラムを設計、作成そして保守を行います。



要約すると、GeneXus は複数のプラットフォームでクリティカルなアプリケーションの、急速な開発を成し遂げるため、自動的にプログラムとデータベースを**設計、作成**そして**保守**を行う**知識に基づく**ツールです。

パート2：最初のアプリケーション



推定された時間：1 時間 40 分

知識に基づくアプリケーションの設計

このセクションでは、ユーザの要件に基づいて GeneXus ナレッジベースを作成及びアプリケーションを設計することを学びます。今回のケースにおいては、請求書を最初に設計して、その後セールス管理システムで要求されている他の構成要素を設計します。



ビジネスアナリスト 対 開発者

エンドユーザの要求を理解することは、自動化することができないソフトウェア開発における作業の一つです。そのため、我々は GeneXus 開発者をプログラマー、コーダーあるいは単なる開発者と呼ぶのではなく、**ビジネスアナリスト**と呼んでいます。



アプリケーションのプログラミング 対 アプリケーションの宣言

GeneXus 方法論は、アプリケーションが取り扱うエンドユーザの実体を**記述**することに基づきます。(有形と無形両方の現実オブジェクト)

エンティティにエンドユーザ要件を高い抽象概念レベルで**記述**することを行います。

この理由により、我々は**宣言プログラミング**を使用します。

このように、**ビジネスアナリスト**は現実を記述し、GeneXus は要求された機能を満たすためにデータモデルを特定のデータベースに生成し、アプリケーションプログラムを生成します。

この現実が変わる際に、ビジネスアナリストはその変更を記述します。そして、GeneXus はその変更を自動的にデータモデルとプログラムに反映してくれます。

ステップ0：ケーススタディ

ある会社が請求書システムを管理するアプリケーションを必要とするしましょう。

現在彼らはそのアプリケーションを保有しておらず、製品コードが記載された大きなカタログを参照しながら、請求書は手作業で作成しています。

このチュートリアルでは GeneXus 方法論の基礎を学習しながら、請求書(Invoice)、顧客(Customers)と製品(Products)を処理していきます。

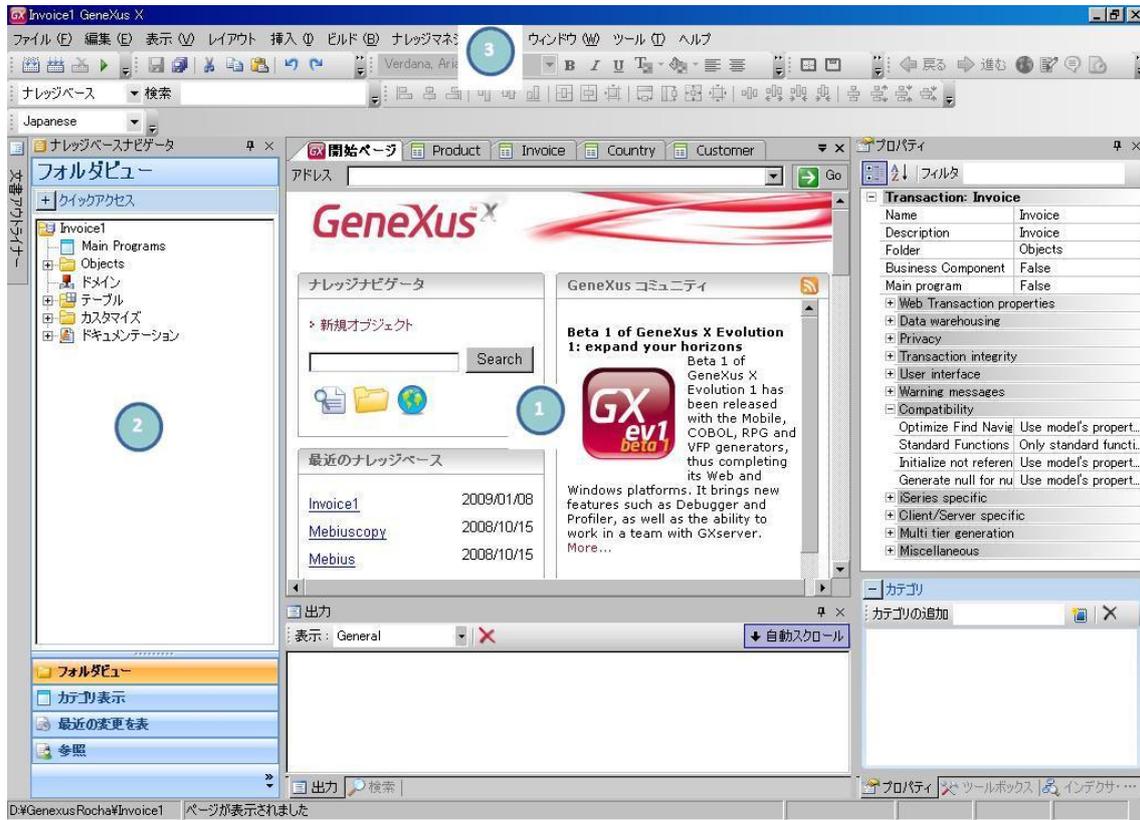
このチュートリアルで GeneXus に現状の実務を記述することで、NET ジェネレータと SQLServer データベースを使用するウェブ請求書システムが生成されることを学びます。

ステップ1：開発環境

GeneXus を起動すると、下記の図のようなインターフェースが表示されます。

このインターフェースは IDE(統合開発環境)と呼ばれます。それは各開発者にとって直感的で使いやすく、そして独自設定ができるインターフェースです。

このインターフェースは、複数のウィンドウに分割されています：



図：開発環境(IDE)

1	<p>メインウィンドウ：RSS を使ってダイナミックに、ツールに関する技術情報やユーザコミュニティを表示する開始ページです。</p> <p>この情報を得るためにはインターネット接続を必要とします。</p> <p>もしパソコンがインターネットに接続されていない場合は GeneXus コミュニティに発表されるニュースを表示することができません。</p>
2	<p>ツールウィンドウ</p>
3	<p>ツールバー</p>

ステップ2：ナレッジベースの作成および環境の定義

GeneXus でアプリケーションを作成するための第一歩は、ナレッジベースを作成し、作業環境を定義することです。

!	<p>ナレッジベース(KB)</p> <p>それは、複数のプラットフォームでアプリケーションを生成するために必要な情報を保管する場所です。</p> <p>言い換えると、現実の記述を保管する場所といえます。</p>
---	--

!	<p>環境</p> <p>アプリケーションを特定のソフトウェアプラットフォームで生成及び保守を行うためには実行プラットフォームに関連する情報を全て環境に定義します (例：プラットフォームのジェネレータ、データベース接続情報、DBMS、ユーザインタフェース、その他のプラットフォームのプロパティ)。</p> <p>このためには、環境ごとに DBMS、対象言語及びパラメータを設定します。</p> <p>GeneXus は、選択されたプラットフォーム上でデータベーススキーマと全てのプログラムを生成及び保守します。</p> <p>従って、GeneXus アナリストは対象プラットフォームの深い知識を必要としません。</p>
---	---

もし初めて GeneXus トライアル版を開いた場合、GeneXus がナレッジベースを自動的に作成するメッセージが表示されます。

初めてトライアル版を起動すると、ナレッジベースは SQL+.NET 環境において自動的に作成されます。

新しいナレッジベースと新しい環境を作成する場合は、下記のステップのとおりに行ってください：

1. **メインウィンドウに表示されている**開始ページ内にある、「**新規ナレッジベース作成**」をクリックしてください。
2. ナレッジベースの名前に“Demo”と入力してください。
3. ナレッジベースを作成するディレクトリを選択してください。
4. 生成する言語を選択してください。
5. 作成を完了するには「**作成**」をクリックしてください。



図：ナレッジベースを作成するためのダイアログボックス

ステップ3：トランザクションオブジェクトの作成

ユーザー要件を GeneXus オブジェクトに定義することが目的です。

!	<p>トランザクションオブジェクト</p> <p>これは、アプリケーションが取り扱う現実のオブジェクトを表現します。</p> <p>定義済みのトランザクションは、アプリケーションのデータモデル（第三正規形）を推論するのに用いられます。</p> <p>GeneXus では、エンドユーザが物理データベースに対してインタラクティブにデータの新規登録、更新、削除を行うためのアプリケーションプログラムについて、トランザクションオブジェクトを用いることで実現します。</p>
---	--

一旦ナレッジベースが作成されると、次のステップは請求書を表す最初のトランザクションを作成することになります。そのためには、下記のステップの通りに行ってください。

1. **Objects** を右クリックして「新規」→「オブジェクト」を選択するか、もしくはツールバーの「ファイル」→「新規」→「オブジェクト」を選択してください。また **CTRL+N** ショートカットキーを使用して新しいオブジェクトを作成することもできます。

2. タイプの選択から、**Transaction** オブジェクトタイプを選択してください：

3. 名前に“Invoice”と入力してください。

4. デスクリプションに請求書と入力します。

5. 「作成」をクリックしてください。



図：新しいオブジェクトの作成



図：新しいオブジェクトダイアログボックス

ステップ4：トランザクション構造の定義

次のステップでは、請求書オブジェクトに属している項目属性の定義とそれらがどのように関連しているのかを定義します。

!	<p>トランザクションオブジェクト構造 (Structure)</p> <p>トランザクションオブジェクト構造は、現実のオブジェクトが必要とするデータ（項目属性）を表すものです。トランザクション構造では、（エンドユーザが操作するデータ）項目属性とそれらの関連性を定義します。この構造に基づいて、GeneXus は第三正規形に対応するデータベース（テーブル、キー、インデックス、整合性制約、その他）を自動的に設計及び保守します。</p> <p>各項目属性は、どこで使用されても常に同じ名前であればなりません。 更に、2 つの異なる項目属性は同じ名前を持つことはできません。</p> <p>トランザクション構造を定義する主な要素は、以下の通りです：</p> <ul style="list-style-type: none"> ● 名前、タイプ、デスクリプション：各項目属性は名前、データタイプと説明によって定義されます。 ● レベル：項目属性は一つ以上のレベルでグループ化されます。これらのレベルは、入れ子もしくは並列に定義できます（複数の入れ子レベルを定義することも可能です）。 例：請求書の明細行は、ルートレベルから 1 つ入れ子になったレベルを表します。 請求書の明細行レベルは、請求書が多くの行を持っていることを意味します。つまり、それは請求書と請求書の明細行間の 1 対多数の関係を定義しています。 ● 主キー項目属性(PK)： 各レベルにおいて、一つ以上の項目属性が主キーとして定義されなければなりません。 <ul style="list-style-type: none"> ○ 主キーは、各レベルのインスタンスの識別子です。 ○ 主キーの値は唯一で、一旦主キーが入力されると更新することができません。 ○ もし自然にオブジェクトの主キー（識別子になる項目）が決まらない場合は、明示的に例えば CustomerId の様な主キーを作られなければなりません；
---	--

生産性ヒント:

	<p>新しい項目属性の名前を入力し始める時にピリオドキーを押すことにより、トランザクションの名前が項目属性名に書きこまれます。</p> <p>項目属性のデスクリプションが自動的に推論されて意味をなすために、項目属性名にキャメルケースを使用してください（日本語の場合は項目属性名から意味を推論することはできません）。</p> <p>例：「InvoiceTotal」項目属性の場合は、デスクリプションが「Invoice Total」と推論されます。</p> <p>データタイプ（型）をコンボボックスから選択するには、指定するデータタイプの最初の文字を入力してください。</p> <p>また、Character 型や LongVarChar 型、Numeric 型あるいは VarChar 型の場合は、桁数を設定してください。</p>
---	--

下のテーブルで示される通り、Invoice トランザクションの Structure エlementに項目属性名、データタイプとデスクリプション（説明）を入力してください。

項目属性名、データタイプとデスクリプションの間でカーソルを移動するためには、TAB キーを使用してください。新しい項目属性を追加するには、ENTER キーを使用してください。

名前	タイプ	デスクリプション
InvoiceID	Numeric(4.0)	請求書番号
InvoiceDate	Date	請求日
CustomerID	Numeric(4.0)	顧客番号
CustomerName	Character(20)	顧客名
<p>ここまで、請求書のヘッダ項目を入力しました。これからは明細行を入力します。 明細行を入力するために、データ構造に新しいレベルを加えます。CTRL + 右矢印を押してください。</p>		
ProductID	Numeric(4.0)	製品番号
ProductName	Character(20)	製品名
ProductPrice	Numeric(8.2)	単価
LineQuantity	Numeric(4.0)	数量
LineTotal	Numeric(8.2)	行金額
<p>ENTER そして CTRL + 左矢印を押して、ヘッダレベルに戻ります。フッタを入力してください。</p>		
InvoiceSubtotal	Numeric(8.2)	小計
InvoiceTax	Numeric(8.2)	税
InvoiceTotal	Numeric(8.2)	合計

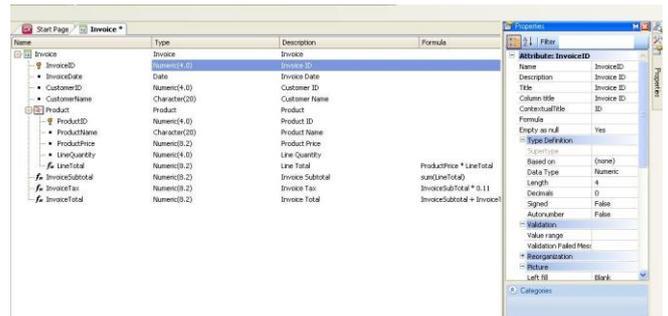
各レベルの最初の項目属性はそのレベルの主キーとして定義されます。項目属性を右クリックして「設定・解除キー」又は (CTRL + K) オプションを選ぶことによって変更することができます。

主キー項目属性は、名前の左にある小さい鍵(🔑)で識別されます。

この例では、InvoiceID はレベル1の識別子です、そして、ProductID 項目属性はレベル2の識別子です。

ある請求書番号 (InvoiceID) に対して ProductID 項目属性の値が異なる行で繰り返されないことを意味します。

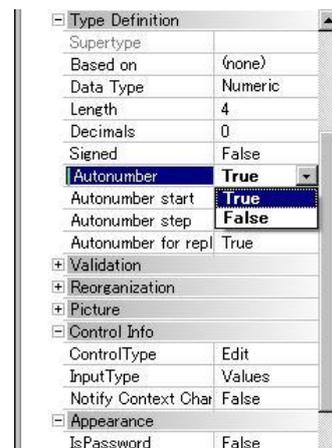
1. 項目属性 InvoiceID を選択後、画面の右側にマウスを動かすことにより、選択した項目属性のプロパティメニューが表示されます。



図：項目属性の定義画面

2. 「Type Definition」オプションを展開して、InvoiceID の「Autonumber」プロパティを True に設定してください。

生成されるプログラムで InvoiceID（請求書番号）は、自動採番されます。



図：「Autonumber」プロパティ

3. 編集ツールバーにある保存ボタン()を押して、トランザクション構造を保存してください。

このツールバーが見当たらない場合は、メニューバーを右クリックして「編集」オプションにチェックを入れてください。保存をするもう一つの方法はメニューの「ファイル>>保存」オプションから保存ができます。

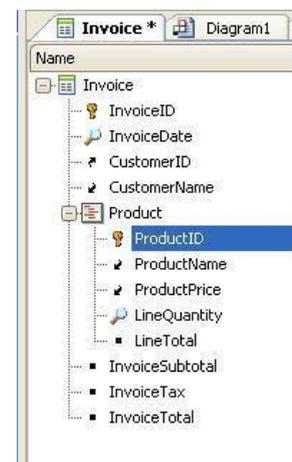
これでレベルが二つあるトランザクションが定義されました。

- レベル 1 (Invoice)に請求書ヘッダに必要なとされる情報の定義。
- 入れ子レベルに請求書の明細行で繰り返される情報の定義。

請求書の明細行では行番号を主キーとして使っていないことに注目してください。

これは、意図的な設計です。

請求書の明細行に Product ID を主キーとして使用することで、同じ製品が同じ請求書に 2 回登録されることがありません。



図：Invoice トランザクションの Structure エlement

!	<p>普遍的な関係仮定(URA)</p> <p>GeneXus 方法論の主な要素は、同じ名前を持っている項目属性が同じ項目属性であるという仮定です。これは普遍的な関係仮定 (URA) と呼ばれています。</p> <ul style="list-style-type: none"> 概念的に等しいものは、全て同じ名前であればなりません。 異なる概念を持つものは、同じ名前を持つてはいけません。 <p>これにより、同じ項目属性を他の GeneXus オブジェクト (トランザクション、プロシージャ、その他) で参照するだけで使用できます。</p> <p>GeneXus は、項目属性の名前に基づいてデータモデルのテーブル間の関係 (外部キー) を確立します。</p>
!	<p>GIK(Genexus インクレメンタル知識) 命名規則</p> <p>GIK は、Artech 社が作った命名規則です。</p> <p>項目属性の概念を出来るだけはっきり伝えられるように名前が付けられています。</p> <p>この命名規則では、項目属性の名前は四つの構成要素からできています。そして、そのうちの幾つかは任意です：</p> <ul style="list-style-type: none"> オブジェクト+カテゴリ[+ 特性] [+ 補足] <p>オブジェクト：実際のエンティティ名。項目属性の名前に表すことで項目属性の管理がしやすくなります。</p> <p>通常は、項目属性が定義されたトランザクションの名前になります。項目属性は定義されたトランザクションと関連したテーブルに保管されますので、項目属性の名前には常にトランザクション名をつけないでも構いません。</p> <p>(これは、特定のトランザクションのみで推論されるものではありません)。</p> <p>例：Invoice、Customer、Product、Line。</p> カテゴリ：項目属性の意味を表すカテゴリです。項目属性がオブジェクト間及びトランザクション環境で果たす役割を意味します。 <p>例：ID、Date、Invoice、Description など。</p> 特性及び補足は任意な部分です。例：Start、End、Minimum、Maximum、など。

ステップ 5 : 計算された項目の定義⇒式

!	<p>式</p> <p>式は、他の項目属性の値から推論することができる項目属性です。</p> <p>式項目属性は、通常の項目属性と同様に定義します。例えば、名前、データ型と説明を定義し、さらに計算式を定義します。</p> <ul style="list-style-type: none"> 特に明記していない限り、式が定義される項目属性はデータベースに保管されません (仮想項目属性になります)。
---	--

	<ul style="list-style-type: none"> 式は、ナレッジベースに対してグローバルです それらは定義されたトランザクションの範囲内だけで有効ではなく、ナレッジベース全体の中で有効です。 項目属性がトランザクションから、あるいは他の GeneXus オブジェクト (レポート、ウェブパネル、その他) から参照される度に式が実行されることを意味しています。 ユーザが定義する変数 これらの変数は式に記述することができません。なぜなら、変数はローカルであるため定義されたオブジェクト内でしか使用できません。
--	---

以下の式項目属性を定義しましょう：

LineTotal=ProductPrice * LineQuantity

InvoiceSubTotal = SUM(LineTotal)

InvoiceTaxes= InvoiceSubTotal * 0.05

InvoiceTotal = InvoiceSubtotal + InvoiceTax

- (項目属性のデスクリプションの右側にある) LineTotal 項目属性の**式**の列をダブルクリックしてください。
- 次の式を入力してください：「ProductPrice * LineQuantity」。式のエディターを開くには、項目属性のプロパティ一覧にある「Formula」プロパティから右クリックで開くこともできます。
- このセクションの始めにリストされた残りの式は、ステップ 1 と 2 を繰り返して定義してください。

新しい式を保存するためには「**保存**」をクリックしてください。

式の定義を完了されたら、トランザクションは以下の図のようになります。

名前	タイプ	デスクリプション	式	再計算
Invoice	Invoice	請求書		
InvoiceID	Numeric(4.0)	請求書番号		No
InvoiceDate	Date	請求書日		No
CustomerID	Numeric(4.0)	顧客番号		No
CustomerName	Character(20)	顧客名		No
Product	Product	Product		
ProductID	Numeric(4.0)	製品番号		No
ProductName	Character(20)	製品名		No
ProductPrice	Numeric(8.2)	単価		No
LineQuantity	Numeric(4.0)	数量		No
fx Line Total	Numeric(8.2)	行金額	ProductPrice * LineQuantity	
fx InvoiceSubTotal	Numeric(8.2)	小計	sum(LineTotal)	
fx InvoiceTax	Numeric(8.2)	税	InvoiceSubtotal * 0.05	
fx InvoiceTotal	Numeric(8.2)	合計	InvoiceSubtotal + InvoiceTax	

図 1 : 式が定義された Invoice (請求書) トランザクション

ステップ 6 : GeneXus が推論したデータモデルを見ます

GeneXus が推論したデータモデルを見て、それを修正することもできます。

!

知的なデータモデル生成

トランザクションを保存する度に GeneXus はエンドユーザのエンティティに対応する最適なデータモデル(第3正規形において冗長なし)をトランザクションオブジェクトから推論してくれます。

このデータモデルに基づいて、DBMS が設定されますと物理テーブルが生成されます。

トランザクションオブジェクトの構造は、作成されるテーブルとインデックスを確定します：

- ・ テーブルとインデックスの名前は、GeneXus により自動的にトランザクション名が割り当てられます。しかし、必要であればアナリストが指定した名前に変更することが可能です。
- ・ GeneXus は冗長無しで第三正規形においたデータモデルを推論します。しかしながら、必要であれば GeneXus が自動的に管理する冗長項目属性を定義することができます。
- ・ N レベルトランザクションに対応しているテーブルの主キーは、前の入れ子にされた N-1 レベルの識別子と N レベルの識別子を連結して得られます。

1. 左側のメニューにある**テーブル**フォルダを選択してください。
2. Invoice と InvoiceProduct テーブルをダブルクリックして開いてください。GeneXus は、トランザクションオブジェクトから推論したデータ構造を表示します。

名前	タイプ	デスクリプション	式
Invoice構造		請求書	
InvoiceId	Numeric(4.0)	請求書番号	
InvoiceDate	Date	請求日	
CustomerId	Numeric(4.0)	顧客番号	
CustomerName	Numeric(4.0)	顧客名	
ロジカルな項目属性			
InvoiceSub Total	Numeric(4.0)	小計	sum(LineTotal)
InvoiceTax	Numeric(4.0)	税	InvoiceSub Total*0.05
InvoiceTotal	Numeric(4.0)	合計	InvoiceSub Total*InvoiceTax

図 : Invoice (請求書) テーブル

名前	タイプ	デスクリプション	式
InvoiceProduct構造		Product	
InvoiceId	Numeric(4.0)	請求書番号	
ProductId	Numeric(4.0)	製品番号	
ProductName	Numeric(4.0)	製品名	
ProductPrice	Numeric(4.0)	単価	
LineQuantity	Numeric(4.0)	数量	
ロジカルな項目属性			
Line Total	Numeric(4.0)	行金額	ProductPrice*LineQuantity

図 : InvoiceProduct (Product) テーブル

上のデータベース一覧において、GeneXus が正規化したデータモデルを自動的に推論したことが確認できます。推論した結果、Invoice（請求書）トランザクションオブジェクトに対応する以下の 2 つのテーブルが作成されました：Invoice（請求書のヘッダ）と InvoiceProduct（請求書の明細行）：

<u>Invoice</u>	<u>InvoiceProduct</u>
InvoiceID	InvoiceID
InvoiceDate	ProductID
CustomerID	ProductName
CustomerName	ProductPrice
	LineQuantity

注釈：

- InvoiceProduct テーブルの主キーは 2 つの項目属性により構成されました：InvoiceID と ProductID（レベル1の識別子（InvoiceID）とレベル2の識別子（ProductID））。
- GeneXus は式として定義された項目属性を自動的にテーブルから削除して、グローバル式に変換しました。グローバル式に変換されたことにより、その項目属性はナレッジベース内のどこからでもアクセスできます。
- Invoice テーブルにおいて
 - 同じ InvoiceID を持つ請求書は存在しません。
 - InvoiceID ごとに InvoiceDate、CustomerID と CustomerName には **1 つ**の値しか存在しません。
- InvoiceProduct テーブルにおいて
 - 同じ InvoiceID と ProductID を持つ請求書の明細行は存在しません。
 - InvoiceID と ProductID の各ペアにとって、ProductName、ProductPrice と LineQuantity には **1 つ**の値しか存在しません。

ステップ7：トランザクションオブジェクトのフォームを見ます

作成したトランザクションオブジェクトに GeneXus は自動的に既定のウェブフォームを生成しますので、それを確認しましょう。まず Invoice（請求書）トランザクションオブジェクトを開いてください。

!	<p>ウェブフォーム</p> <p>トランザクションオブジェクトの構造を定義する際に GeneXus はエンドユーザがアプリケーションでデータにアクセスする既定のウェブ画面を作成してくれます。</p> <p>これらのウェブフォームは、GeneXus 設計者によるカスタマイズが可能です。</p>
----------	--

ウェブフォームを見るためには、下記のステップを行ってください：

1. Invoice（請求書）トランザクションオブジェクトの **Web Form** エlementを選択してください。

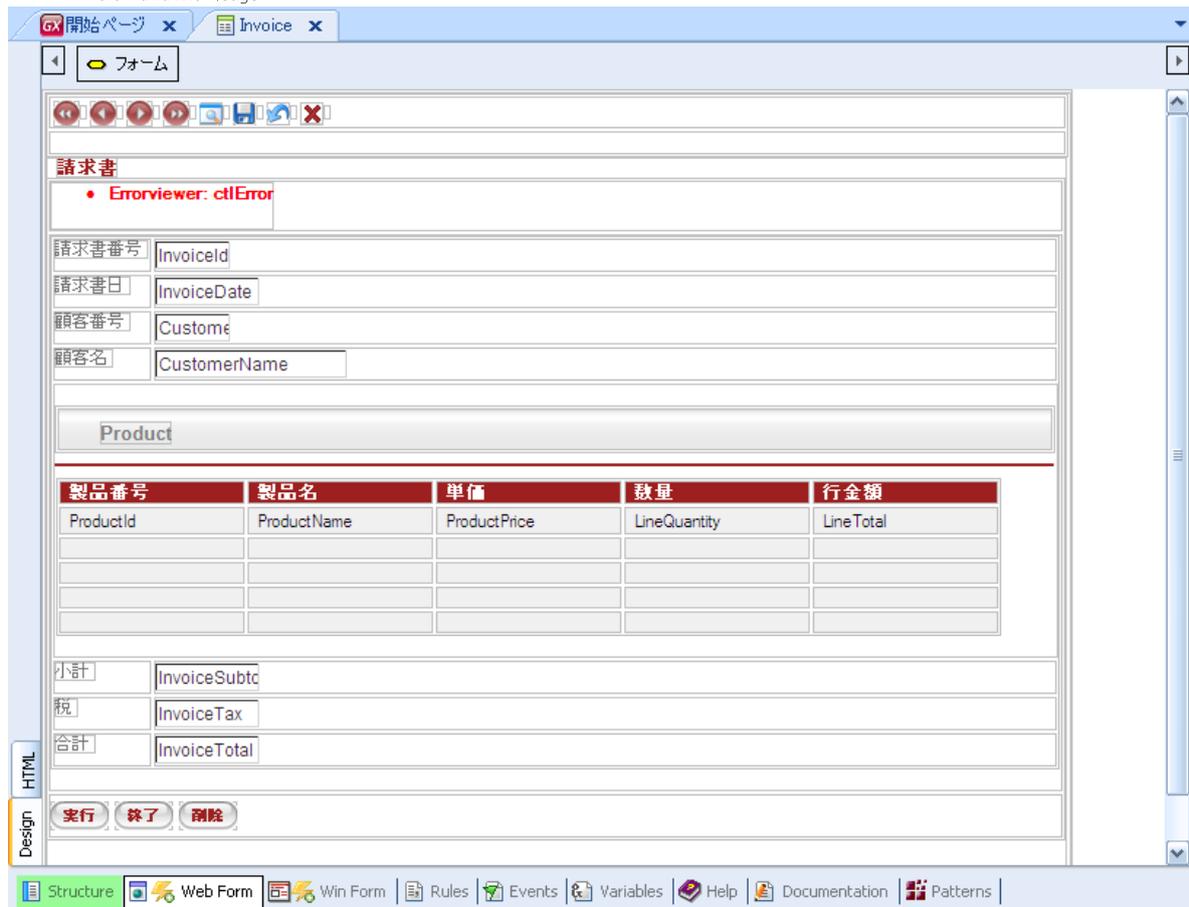


図 : Invoice トランザクションにおける既定のウェブ画面

“Error Viewer: ctlError” は、エラーメッセージが表示される既定のコントロールです。

フォームのどこにでも配置することができ、また、プロパティの設定を変更することができます。

実行時に表示されるメッセージは、Msg ルール、Error ルールと GeneXus の自動検証が出力するものです（例えば、参照整合性、データ型エラー、その他）。

このフォームでエンドユーザは対応するテーブルに新しい請求書を新しいレコードとして登録することができます。

さらに、ユーザは既存の請求書を更新及び削除することができます。

新規登録・更新・削除の機能はトランザクションのロジックに内在されているため GeneXus 設計者は何も記述する必要はありません。

GeneXus は選択された言語のネイティブコードを自動的に生成してくれます。

GeneXus のトランザクションオブジェクトを定義する際には、以下のことを意識してください：

- 明示的に：データの表示及び入力用のユーザインタフェース定義。
- 暗黙的に：アプリケーションのデータモデル設計(テーブル、インデックス、その他)

ステップ 8 : アプリケーションの実行

!	<p>第3 正規形のデータベースの作成</p> <ul style="list-style-type: none">推論したデータモデルに基づいて、GeneXus は選択された DBMS でデータベースを作成するための実行プログラムを生成します。データ構造を更新した際には、GeneXus はデータベースを再編成するための実行プログラムを生成します。再編成を行う際には、データベース構造の変更を行いデータも新しい構造に移行します。これらの場合には、GeneXus が何をするかを示すデータベース作成レポートあるいは影響分析レポートが表示されます。
----------	--

このステップでは、DBMS として SQL Server を使用し.NET でアプリケーションを実行します。

正規バージョンでは、GeneXus がサポートしている DBMS のどれでも使用することができます。

GeneXus がサポートしている DBMS の一覧を見るには以下のリンクをクリックしてください：

<http://www.genexus.com/technologies>

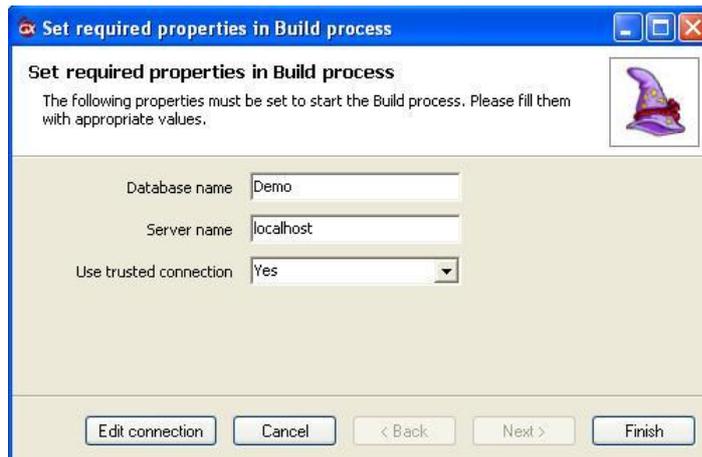
!	<p>データベース作成レポート：</p> <p>これは、選択された DBMS において GeneXus が生成するデータベース構造を説明するレポートです。</p> <p>これは、推論したデータモデルを基に生成する推奨するデータベース構造の全ての情報を示します。</p> <p>各テーブルに関する情報は、5 つのセクションに分離されます：</p> <ul style="list-style-type: none">ヘッダ：テーブルの名前、テーブルに対してのアクション、警告及びエラーを含みます。データモデルにエラーがある場合は、再編成ボタンは使用不能になります。テーブル構造：テーブルの項目属性、それらの関連及び項目属性に対してのアクションを示します。インデックス：GeneXus がデータベースの参照整合性の管理を行うためのインデックスと能率的にテーブルをアクセスするためのインデックスを示します。外部キー制約：テーブルの整合性制約を示します。ステートメント：実行される順番を示します。
----------	---

F5 を押すか、もしくはツールバーの「ビルド>>開発者メニューを実行」オプションを選択してください。

1. DBの設定について、ウィザードに従い、以下の通りに設定してください：

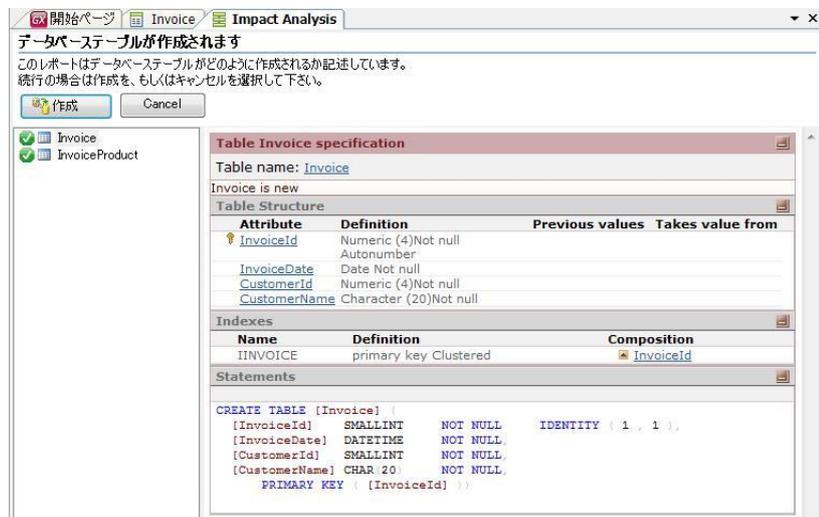
- データベース名：
Demo
- サーバー名：
localhost
- 信頼認証接続を使用
します：Yes

続行するために、Finish ボタンをクリックしてください。



図：モデル設定の詳細

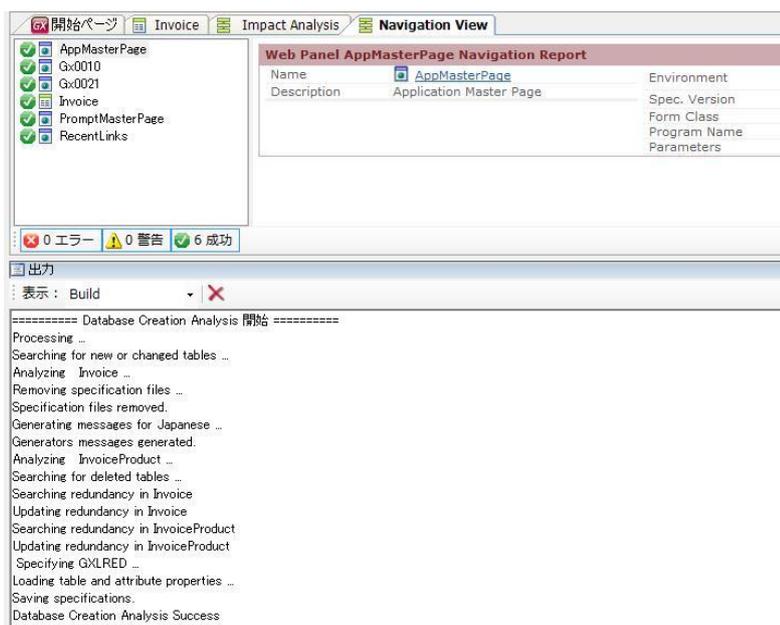
2. GeneXus はデータベース作成レポート (Impact Analysis) を出力します。



図：データベース作成レポート画面

3. GeneXus はテーブルを作成するためのコードを選択された言語で生成し、データベースをアクセスします。

また、各オブジェクトの生成結果は、ナビゲーションビューに表示されます。



図：出カウィンドウのメッセージ

1. 開発者メニュー

(DeveloperMenu) は、実行可能なオブジェクト全てを表示するXML ファイルです。これは、アプリケーションをプロトタイピングするための補助メニューです。請求書のリンクをクリックしてください。

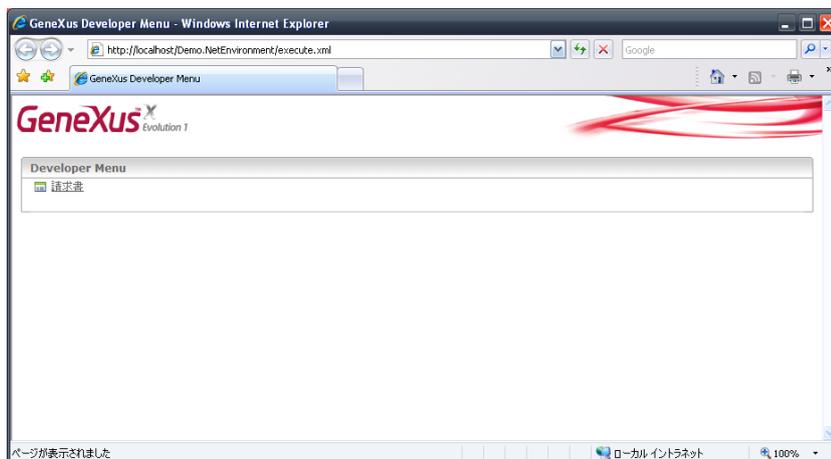


図 : 開発者メニュー

2. いくつかの請求書を登録してください。AJAX 技術を利用しており、ページを再ロードせずとも、式の値を自動的に算出します。

3. 完了したらブラウザーを閉じてください。

Application Header

Recents: 請求書

請求書

請求書番号

請求書日 📅

顧客番号

顧客名

Product				
製品番号	製品名	単価	数量	行金額
0		0.00	0	0.00
0		0.00	0	0.00
0		0.00	0	0.00
0		0.00	0	0.00
0		0.00	0	0.00
[行追加]				

小計 0.00
 税 0.00
 合計 0.00

図 : 請求書トランザクションのインスタンス

ステップ 10 : ビジネスルールの追加 (ルール)

いくつかの基本的なビジネスロジックをアプリケーションに追加しましょう。

!	<p>GeneXus ルール</p> <p>GeneXus ルールは、各オブジェクトのビジネスロジックを定義するための場所です。</p> <p>ルールは宣言的に記述し、GeneXus はどのルールをいつ適用すべきか (トリガーオーダー) を知的に決定します。</p> <p>これらのルールは、トランザクションオブジェクトで非常に重要な役割を持っています。それは、トランザクションの動作を定義することです (例えば: デフォルト値の設定、データ規制の定義、その他) 。</p> <ul style="list-style-type: none">• ルールには、トランザクション構造で定義されている項目属性、変数及び関数を含める (記述する) ことができます。• ルールは、宣言的な手法で定義します。ルールが記述される順序が、必ずしも実行される順序どおりである必要はありません。• ルールの実行順序は、GeneXus が自動的に決定します。 <p>ルールの有効範囲は、定義されたトランザクション内のみです。ルールはローカルであるからです。</p>
----------	---

請求日の初期値を当日の日付に設定するルールを記述しましょう :

1. Invoice (請求書) トランザクションの **Rules** エレメントを選択してください。
2. 項目属性及び変数の既定値を設定する **Default** ルールを使用しましょう。
3. 次の通りにルールを記述してください :
Default(InvoiceDate, &today);
これは、請求日の初期値に当日の日付を設定することを示しています。
4. **保存** ボタンをクリックしてください。

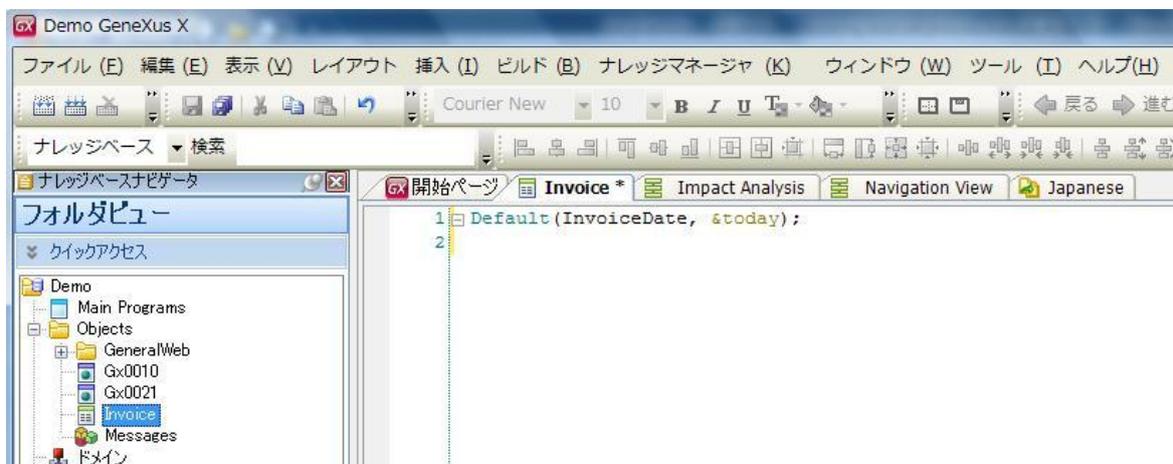


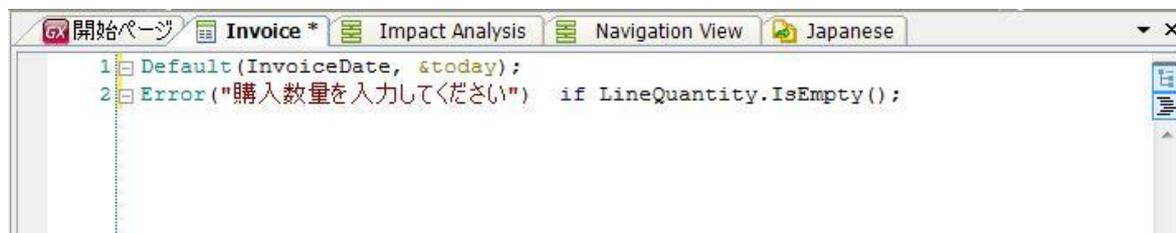
図 : トランザクション ルール

それでは、入力された製品の購入数量が未入力の時に表示されるエラーメッセージのルールを見てみましょう：

5. 次のルールを記述してください：

```
Error("購入数量を入力してください") if LineQuantity.IsEmpty();
```

6. **保存**ボタンをクリックしてください。



図：ルールの記述

7. 入力したルールの動作を確認するために、アプリケーションを実行しましょう。(F5 を押下、もしくはツールバーより管理者メニューのビルドを実行する)

Application Header

Recents: 請求書

請求書

請求書番号

請求書日

顧客番号

顧客名

Product					
	製品番号	製品名	単価	購入数量を入力してください	額
x	1	製品01	10.00	0	0.00
	0		0.00	0	0.00
	0		0.00	0	0.00
	0		0.00	0	0.00
	0		0.00	0	0.00

[行追加]

小計 0.00
 税 0.00
 合計 0.00

図：アプリケーションの実行

ステップ 1 1 : Customer (顧客) トランザクションオブジェクトの作成

顧客は請求書とはまた別のエンティティなので、顧客用のトランザクションを定義しなければなりません。

1. 「名前 : Customer ; デスクリプション=顧客」 トランザクションオブジェクトを作成してください
2. Customer (顧客) トランザクションの構造に以下の項目属性を定義してください :

名前	タイプ	デスクリプション
CustomerID	-----	-----
CustomerName	-----	-----
CustomerAddress	Character(50)	顧客住所
CustomerEmail	Character(50)	顧客電子メール

「CustomerID」と「CustomerName」項目属性の入力を始めると、GeneXus は全体の名前、タイプとデスクリプションを自動的に表示されます。

これらの項目属性は既にナレッジベースに定義されているため、この動作 (サジェスト) が起こります。

3. CustomerID (顧客番号) 項目属性を右クリックして**プロパティ**を選択してください。
4. CustomerID の Type Definition グループ内にある **Autonumber** プロパティを **True** に設定してください。
この定義によって、CustomerID が自動採番されます。
5. CustomerID のプロパティを開いて、サジェスト機能を有効化しましょう。まずは InputType プロパティを Description に設定します。そして、ItemDescription に CustomerName 項目属性 (顧客名) を設定します。

この手法は顧客 ID を特定するために、ID を入力する代わりに顧客名を入力することにより顧客番号は自動的に推論されます。

サジェストプロパティは、入力されたものと一致する顧客名をサジェストしてくれます。

これらのプロパティは、GeneXus が自動的に実装する **AJAX** の一部です。

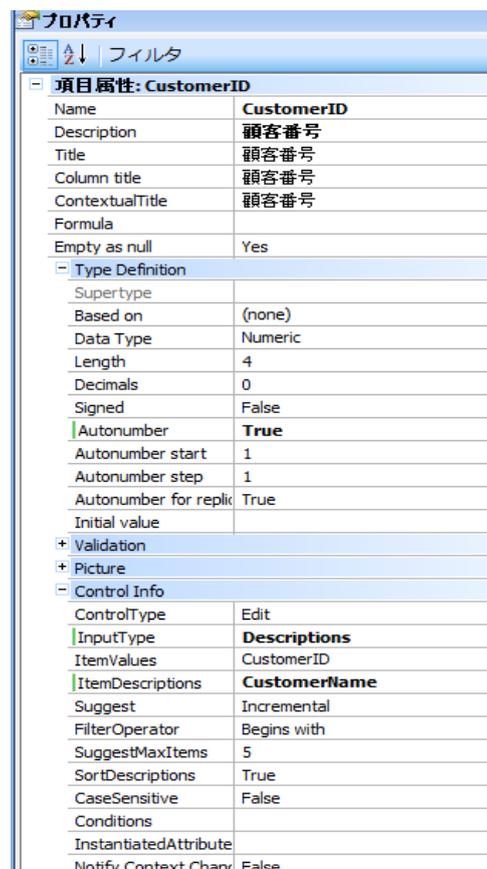


図 : プロパティ画面

下記の図は、顧客トランザクションのウェブフォームを示しています。

名前	タイプ	DESCRIPTION	式	ヌル許容
Customer	Customer	顧客		
CustomerId	Numeric(4,0)	顧客番号		No
CustomerName	Character(20)	顧客名		No
CustomerAddress	Character(50)	顧客住所		No
CustomerEmail	Character(50)	顧客電子メール		No

図：Customer（顧客）トランザクションの **Structure** エlement

顧客

• Errorviewer: ctlError

顧客番号	Customer
顧客名	CustomerName
顧客住所	CustomerAddress
顧客電子メール	CustomerEmail

実行 終了 削除

図：Customer（顧客）トランザクションの **Web Form** Element

Invoice（請求書）トランザクションが既に開かれている場合は、閉じて開きなおします。CustomerID（顧客番号）項目属性のプロパティ変更が請求書のウェブフォームにまで反映されたことにお気づきでしょうか？顧客名のところには CustomerName（顧客名）項目属性から、CustomerID（顧客番号）項目属性に変更されていることが分かります。

ご確認いただけたとおり、これは大いにアプリケーションのユーザビリティを強化します！

図：請求書トランザクションの Web Form エlement

ステップ 1 2 : データモデルに対する影響・変更内容の確認

F5 キーを押して、影響分析レポートを生成してください。

!	<p>影響分析</p> <p>GeneXus が推論したデータモデルの変化を物理データベースに反映するための更新・差分処理を示します。</p>
----------	--

新しいトランザクションの定義後に GeneXus がデータモデルを自動的に正規化したことが分かります。

Invoice テーブルが影響を受け、CustomerName（顧客名）項目属性が CustomerID（顧客番号）項目属性を通して参照することができるため、Invoice テーブルから CustomerName（顧客名）項目属性を削除します。

開始ページ Invoice Navigation View Japanese Customer Impact Analysis

データベースは再編成を必要とします。
 データベース上の変更と、それが再編成プログラムでどのように処理されるか、このレポートに記載されます。
 続行の場合は再編成を、もしくはキャンセルを選択して下さい。

再編成 Close

Customer Invoice

Table Invoice specification

Table name: Invoice

Invoice needs conversion

Table Structure

Attribute	Definition	Previous values	Ta
InvoiceId	Numeric (4)Not null Autonumber		Inv
InvoiceDate	Date Not null		Inv
CustomerId	Numeric (4)Not null		Inv
Del	CustomerName	Character (20)Not null	

Indexes

Indexes	Definition	Compositio
IINVOICE	primary key Clustered	InvoiceI
New IINVOICE1	duplicate	Custom

Foreign key constraints

Referenced table	Attributes
New Customer	CustomerId

0 エラー 1 警告 1 成功

Customer の新しいテーブルが作成されます :

注 GeneXus の分析表で請求書テーブルに複製の顧客がある可能性から正規化する際に影響を受けることを警告で示します。もう一つの警告は、テーブルの新しい項目属性はヌルで作成されることを示します。

データベースは再編成を必要とします。
 データベース上の変更と、それが再編成プログラムでどのように処理されるか、このレポートに記載されます。
 続行の場合は再編成を、もしくはキャンセルを選択して下さい。

再編成 Close

Customer Invoice

Table Customer specification

Table name: Customer

Customer is new

Warnings

- rqz0007 Attribute CustomerAddress does not allow nulls and has not a Initial Value. A value will be used.
- rqz0007 Attribute CustomerEmail does not allow nulls and has not a Initial Value. An value will be used.
- rqz0005 For each value of CustomerId there may be several values of CustomerName

Table Structure

Attribute	Definition	Previous values	Takes
CustomerId	Numeric (4)Not null Autonumber		Invoice.
CustomerName	Character (20)Not null		Invoice.
CustomerAddress	Character (50)Not null		''
CustomerEmail	Character (50)Not null		''

Indexes

図 : データベース(データモデル)表

注釈：

- GeneXus は新しいテーブルを追加しました：Customer テーブル（Customer（顧客）トランザクションに関連します）。
- データベースが正規化されることで CustomerName 項目属性は Invoice テーブルから Customer テーブルに移動されます。
- Customer テーブルに関して：
 - 複数の顧客が同じ CustomerID を持つことはできません。
 - 各顧客に対して**一つの**名前、住所と電子メールしか持てません。
- 請求書テーブルに関して：
 - 複数の請求書が同じ InvoiceID を持つことはできません。
 - Invoice テーブルの CustomerID 項目属性は Customer テーブルからの外部キーです。このように、Customers と Invoices 間は 1 対 N で関連しています：
 - 各 InvoiceID には CustomerID が**1つしか**ありません。
 - 各 CustomerID は**複数の**請求書（InvoiceID）を持つことができます。

ステップ 13 : 分析表を見ます

データベースの再編成を行う準備ができました。

再編成オプションを選択してください。

再編成プログラムは、影響分析で捉えた変更を物理データベースに反映してデータも移行してくれます。

再編成を実行する際に各プログラムの解析表を含む**分析表**を表示してくれます

!	<p>分析表</p> <p>プログラムがどのように実行され、どのようにテーブルにアクセスして、どのように動作をするのかを示します。</p>
----------	--

Transaction Customer Navigation Report

Name	Customer	Environment	.net C#
Description	顧客	Spec. Version	10_0_3-18367
		Form Class	HTML
		Program Name	Customer
		Parameters	

Levels

Level Customer

`=Customer(CustomerId)`

Insert into Customer (CustomerName, CustomerAddress, CustomerEmail)

Update Customer (CustomerName, CustomerAddress, CustomerEmail)

Delete from Customer

Referential integrity controls on delete: **1**

- Invoice(CustomerId)

Table	Program	In Parameters	Out Parameters
Customer	Gx0030		CustomerId

図 : 顧客トランザクションの解析表

1

「削除する際の参照整合性の制御」とは、顧客トランザクションから顧客を削除する時にプログラムがその顧客に請求書がないことを確認することを意味します。

この検索を効率よく実行するために Invoice テーブルの外部キーである CustomerID 項目属性のインデックスが使用されます。

図： 請求書トランザクション解析表

請求書トランザクションの**参照整合性**： INVOICE レベルの CustomerID 外部キーは、INVOICE トランザクションを使って請求書を追加あるいは更新する時に、GeneXus が CustomerID 外部キーに入力される値が Customer テーブルに主キーとして存在することを自動的に確認することを意味します。

この検索を効率よく実行するために、Customer テーブルの主キーである CustomerID 項目属性のインデックスが使用されます。

参照整合性のチェックは入力される項目属性が（例えば Invoice の CustomerID）有効であることを確認しますが、有効な値が何であるかという情報は提供しません。有効な値の検索を容易にするために、GeneXus は**検索一覧**オブジェクト（プロンプト）を作成して、全ての有効な値をその中から選ぶことができます。

ステップ 14： 見栄えの良いフォームの作成（テーマ）

このセクションでは、あらかじめ定義されたテンプレートに基づいて新しいテーマを作成し、アプリケーションのテーマとして指定します。

!	<p>テーマオブジェクト</p> <p>GeneXus テーマオブジェクトでビジネスアナリストの仕事とウェブデザイナーの仕事に分けてウェブアプリケーションの開発と保守が行えます。</p> <ul style="list-style-type: none"> • ウェブフォームは、既定のテーマオブジェクトに基づいて作成されます。 • 新しいテーマを作成してナレッジベース全体に適用、あるいは特定のオブジェクトに適用することができます。
---	--

1. 左のフォルダビューの中にあるカスタマイズノードを展開してテーマノードをさらに展開します。そうすると、いくつかのテーマオブジェクトが確認できます。
2. **GeneXusX** テーマオブジェクトをダブルクリックして開いてください。
3. **Form** を選択して右にあるプロパティ編集画面から「Defaults for Form Control Properties」の配下にある「BackGroundColor」プロパティを変更し背景色を変更してください。
4. 保存ボタン()を押して、変更を保存してください。

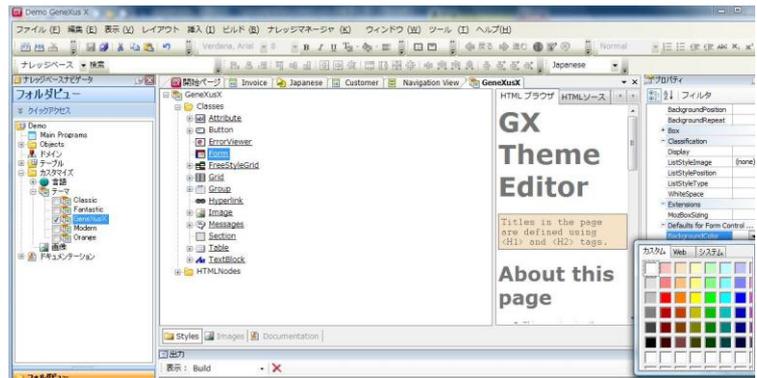


図 : GeneXus テーマエディター

5. 再び送り状トランザクションのウェブフォームを開くと、背景色が反映されたことが見られます。
注釈 : 既にトランザクションが開いて行っていた変更が見られない場合は、一度トランザクションを閉じて再度開きなおしてください。

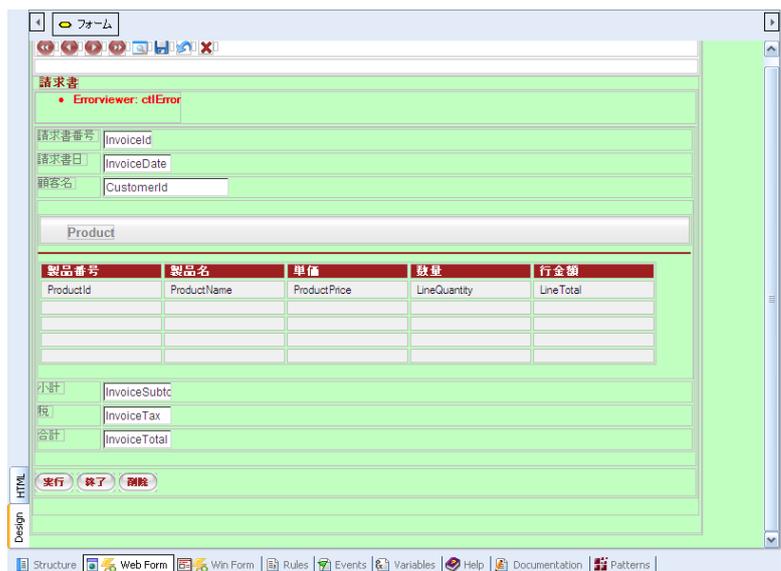


図 : テーマ変更により、背景色が変化した

Invoice (請求書) トランザクションウェブフォーム

パート3 : 保守



推定された時間 : 15 分

インクリメンタル開発とアプリケーション保守

これまではいくつかのトランザクションオブジェクトとビジネスルールに基づいてアプリケーションを作成してきました。今度は GeneXus アプリケーションの保守について、単にオブジェクトを編集もしくは新しいオブジェクトを追加し、その後でデータベースを更新して自動的にアプリケーションプログラムが再生成される事を確認してみましょう。

!	<p>GeneXus によるインクリメンタル開発</p> <p>実務が変わることによって、システム要件も変化します。そしてその変化は GeneXus オブジェクトの更新あるいは新しいオブジェクトの追加により対応します。</p> <p>新しいオブジェクトに基づいて GeneXus は自動的にデータベース（システム変更をデータベース構造に反映してデータの移行もしてくれます）を更新して変更の影響を受けたアプリケーションプログラムを再生成します。</p> <p>GeneXus 方法論がインクリメンタル方法論であると言えます。なぜなら、その方法論は繰り返し作業で連続した近似に基づいたシステム開発であるからです。</p> <p>GeneXus がデータベース及びアプリケーションのプログラムの構造を自動的に保守することで、インクリメンタルな開発が可能になります。</p>
----------	--

ステップ 15 : 新しいオブジェクトの追加 – 製品トランザクション

「名前 : Product ; デスクリプション = 製品」トランザクションオブジェクトを作成してください。

アプリケーションをテストした際に、各請求書の明細行に製品番号、製品説明及び製品価格を入力しなければならないことに気がついたかと思います。これは、販売管理システムに求められることではありません。

このサンプルアプリケーションのような単純なシステムでもそのようなアプリケーション構造は好まれません！

別のシステム画面から製品を追加、更新そして削除する機能が必要とされます。

そのために、製品トランザクションをナレッジベースに追加しましょう：

1. パート2のステップ 3：“トランザクションオブジェクトの作成”及びパート2のステップ4：“トランザクション構造の定義”を基に製品トランザクションを作成してください。

以下の項目属性を製品トランザクション構造に挿入してください：

名前	タイプ	デスクリプション
ProductID	-----	-----
ProductName	-----	-----
ProductPrice	-----	-----

これらの項目属性の名前を入力し始めるとすぐに GeneXus がフルネームでサジェストを表示してくれることに注目してください。

項目属性が既にナレッジベースに定義されているためそのサジェストされた値が表示されるのです。

製品トランザクションの構造とそのウェブフォームは以下のような形になります。

名前	タイプ	デスクリプション	式	ヌル許容
Product	Product	製品		
ProductId	Numeric(4.0)	製品番号		No
ProductName	Character(20)	製品名		
ProductPrice	Numeric(8.2)	単価		

図 : Product (製品) トランザクションの構造

The screenshot shows the 'Design' view of a web form in GeneXus. The form is titled '製品' (Product) and contains a red error message: 'Errorviewer: ctlError'. Below the error, there are three input fields: '製品番号' (Product ID) with the value 'Product', '製品名' (Product Name) with the value 'ProductName', and '単価' (Unit Price) with the value 'ProductPrice'. At the bottom of the form, there are three buttons: '実行' (Execute), '終了' (End), and '削除' (Delete). The interface includes a toolbar with navigation and editing icons, and a bottom menu with options like 'Structure', 'Web Form', 'Win Form', 'Rules', 'Events', 'Variables', 'Help', 'Documentati...', and 'Patterns'.

図 : Product (製品) トランザクションの Web Form

ステップ 16 : データベースの分析と再編成

製品トランザクションオブジェクトを保存する時に、GeneXus は再びデータモデルを正規化します。

データモデルチェックで GeneXus がデータモデルを自動的に正規化したことがわかります。これまで InvoiceProduct テーブルにて構成されていた項目属性(ProductId、ProductName、ProductPrice)が、新しい Product というテーブルに移動されています。

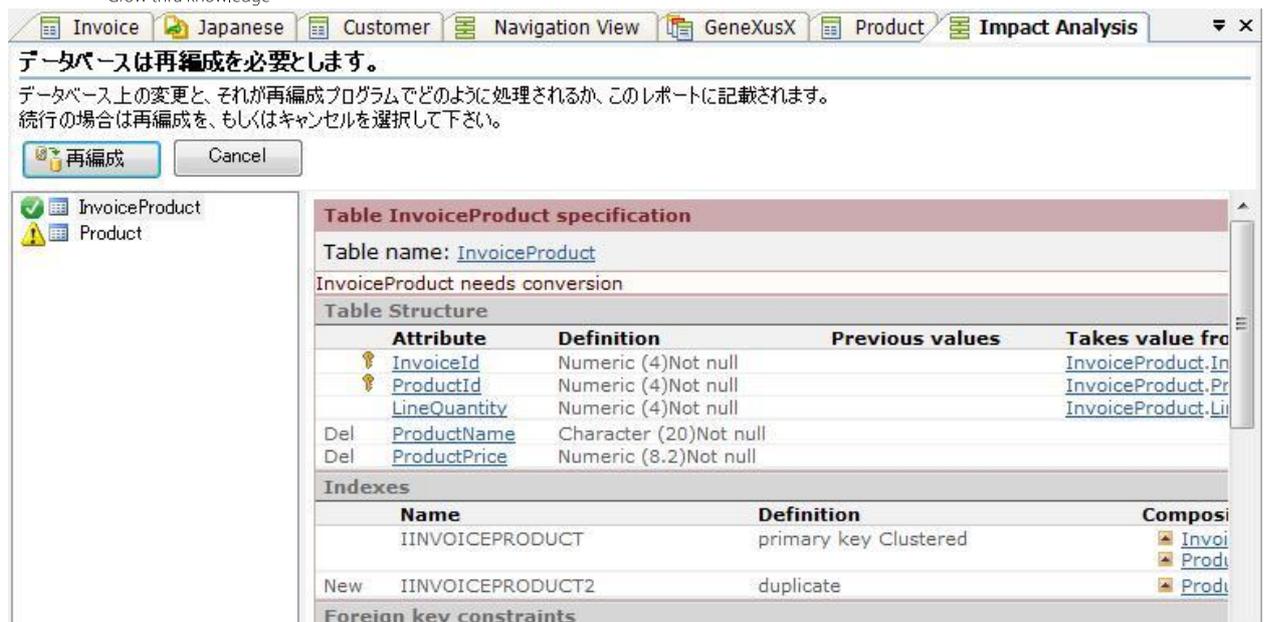


図 : InvoiceProduct と Product テーブルの変化

1. 再編成ボタンを押してください。
2. GeneXus は、データベース再編成を実行し、またアプリケーションプログラムに必要な変更を行います。

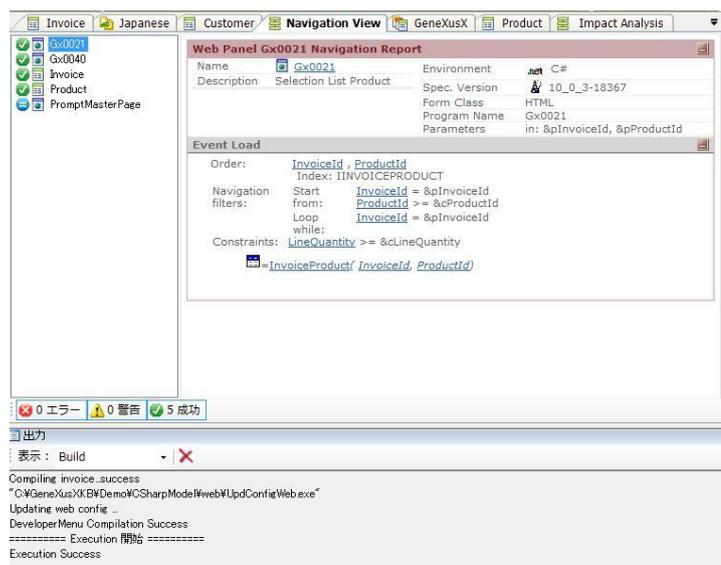


図 : 再編成と分析の実行結果を表示するダイアログボックス

3. これでアプリケーションを実行することができます！

Application Header

Recents: 請求書

←
→
↺
↻
🔍
📄
🔗
✖

請求書

請求書番号

請求書日 📅

顧客名

Product

製品番号	製品名	単価	数量	行金額
0	↑	0.00	0	0.00
0	↑	0.00	0	0.00
0	↑	0.00	0	0.00
0	↑	0.00	0	0.00
0	↑	0.00	0	0.00

[行追加]

小計 0.00

税 0.00

合計 0.00

実行
終了
削除

図：アプリケーションの実行画面

パート4：機能の追加



推定された時間：10分

開発の一部にパターンを使用します。

アプリケーションのテストをすると、トランザクションを通してデータの入力を行っていることに気づきます。

しかし、顧客や請求書あるいは製品を入力、更新、削除するだけではなく、特定の基準に従ってそれらのデータをフィルターして並び替えて一覧表示ができるとさらに便利になります。

この機能を迅速かつ容易な形で実装するために、パターンを使います。

!	<p>パターン</p> <p>パターンは、異なるエレメントに適用される類似したアクション（アナリストの操作）です。</p> <p>パターンにより特定の機能を成し遂げるために必要である全てのオブジェクトを自動作成します。これにより手動による実装を避けることができます。</p>
----------	--

ステップ17：パターンの適用

「ワークウィズ」パターンを作成したトランザクションに適用しましょう。

1. 「表示」メニューから「オブジェクト一覧」オプションを選択してください。
2. Customer（顧客）トランザクション、Invoice（請求書）トランザクション及び Product（製品）トランザクションを選択してください。

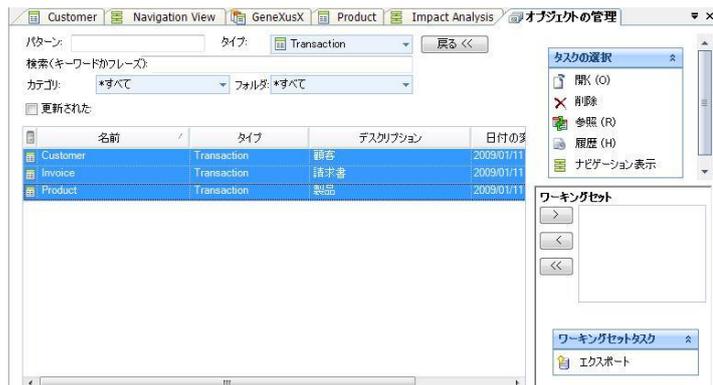


図 2：トランザクションを選択するダイアログ

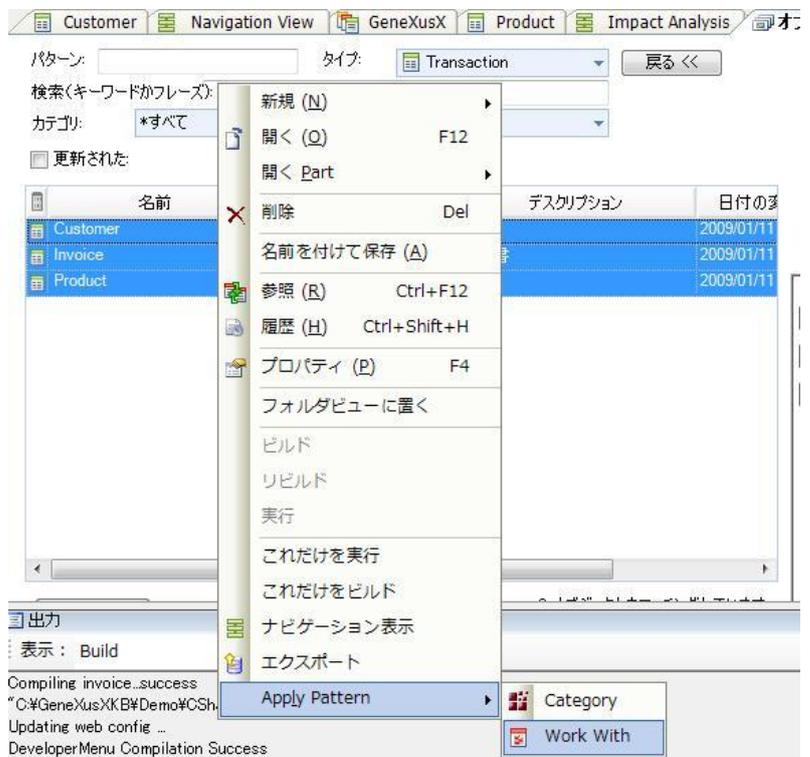
3. 選択したトランザクションを右クリックして、Apply Pattern(パターンを適用します)、Work With オプションを選択してください。

4. 変更を保存してください。

5. F5 を押して、アプリケーションを実行してください。

トランザクションを直接実行できないことが分かります。

顧客、請求書と製品それぞれのパターン “Work With”の実行ができます。

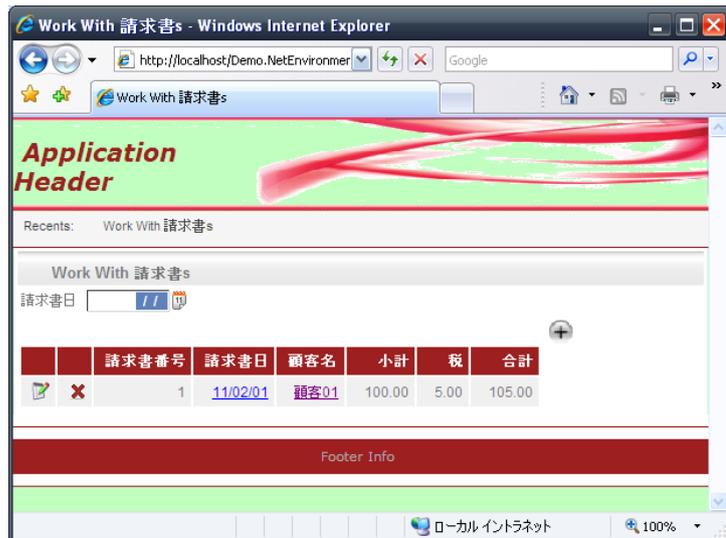


図：トランザクションの選択

6. 請求書の Work With を実行してください。

パターンは、以下の機能を追加しました：

- アプリケーションを再表示せずに日付でデータが抽出できます。
- 列の見出しをクリックしてデータを並べ換えることができます。
- 一覧表は 10 行単位でページングされています。
- リンクをクリックすることで、関連しているデータの照会画面が開くことができます。
- アクセスしたリンクの履歴が保存されます。



図：アプリケーションの実行

パート5：インタラクティブなダイアログ

	推定された時間：15分
---	-------------

インタラクティブなダイアログとクエリーの設計(ウェブパネル)

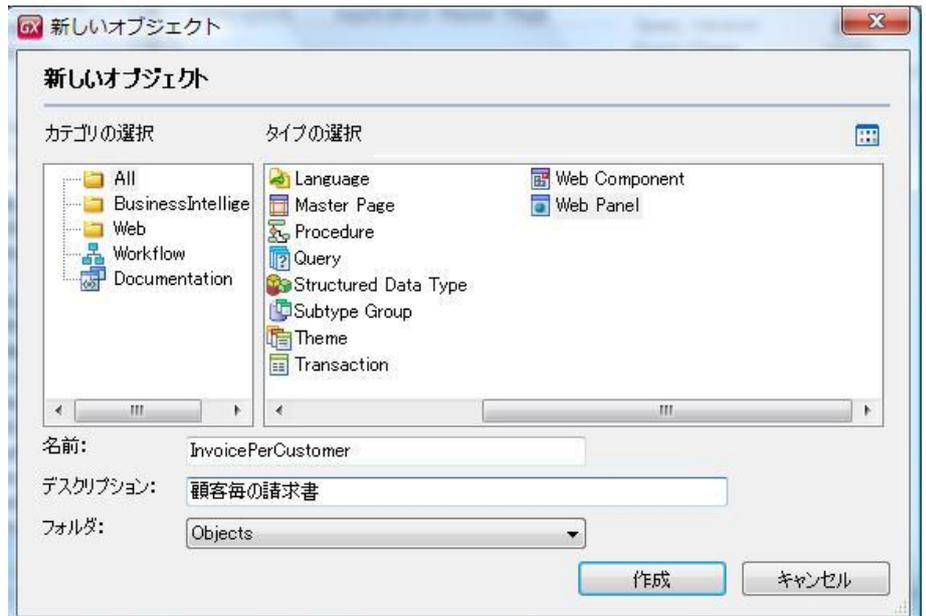
!	<p>ウェブパネル</p> <p>ウェブパネルは、ウェブ環境でデータベースへとインタラクティブなクエリー画面を作成するために使用されます。</p> <p>ウェブパネルは、様々な形のクエリーを実行してデータを多様な方法で見ることを可能にします。ウェブパネルは、単純なイベントドリブン言語で記述されます。</p>
----------	--

ステップ 18：ウェブパネルの作成：顧客ごとの請求書

各顧客の請求書を見るためにウェブパネルを作成しましょう。

ウェブパネルを作成するためには下記のステップを行ってください：

1. オブジェクトフォルダを右クリックして「新規」→「オブジェクト」を選択してください。
2. 作成する「Web Panel」型オブジェクトを選択してください。
3. 名前には“InvoicePerCustomer”を入力してください。そしてデスクリプションに、“顧客毎の請求書”と入力してください。
4. 作成をクリックしてください

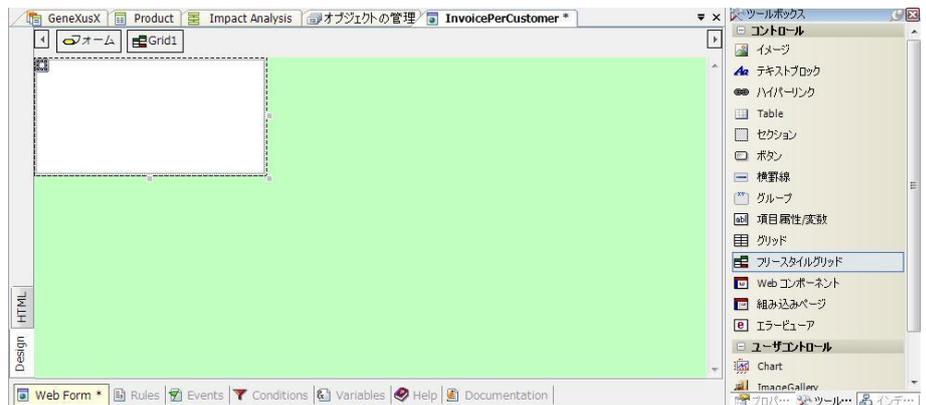


図：オブジェクトの新規作成画面

5. オブジェクト作成後、ウェブパネルは空のフォームを表示します。

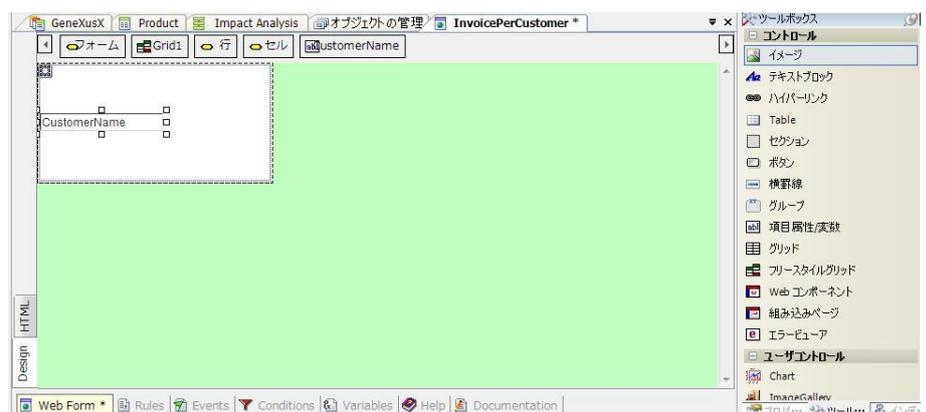
ツールバーにある「表示」メニューから「他のツールウィンドウ」の「ツールボックス」オプションを選択してください。

「フリースタイルグリッド」コントロールをドラッグしてください。



図：フリースタイルグリッドを配置した画面

6. 「フリースタイルグリッド」コントロールを拡大してください。
7. グリッド内に「項目属性/変数」コントロールを挿入してください。
8. **CustomerName (顧客名)** の項目属性を選択し、「OK」をクリックしてください。



図：ウェブパネルの Web Form

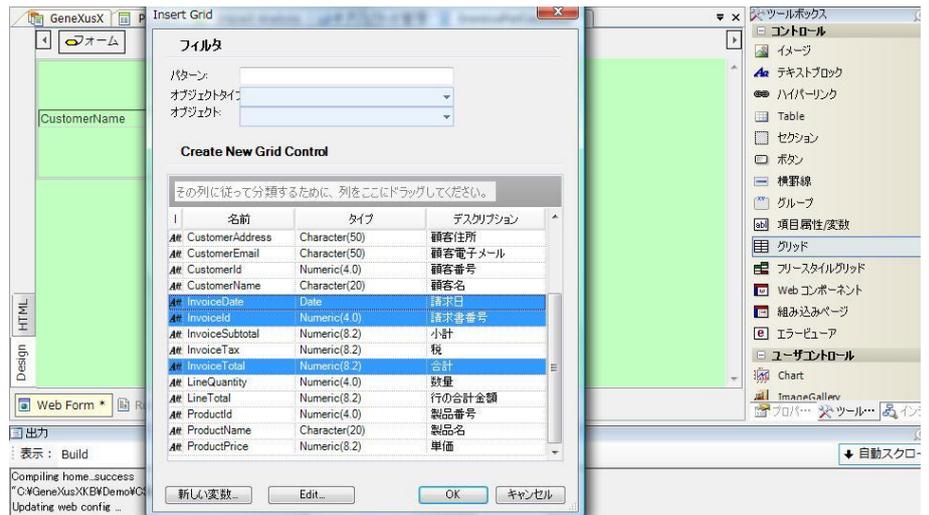
9. 各顧客に対応する請求書一覧を表示させるためには、グリッドを挿入して表示される次の項目属性を選択してください。

「InvoiceID（請求書番号）」

「InvoiceDate（請求日）」

「InvoiceTotal（合計）」

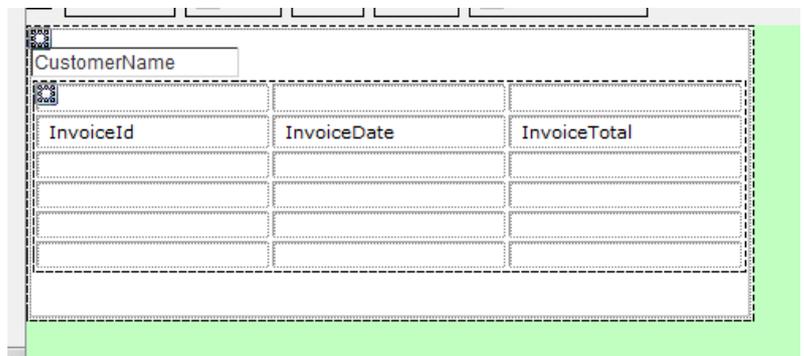
GeneXus は顧客ごとの請求書一覧を表示するためのテーブル間の関係を見つけます。



図：グリッド挿入画面

10. 「列の調整」画面を閉じてください。

11.  を押して変更を保存してください。



図：グリッドを持ったウェブパネル画面

12. F5 を押して、実行してください。



顧客01

請求書番号	請求日	合計
1	2009/02/02	12500.00
2	2009/02/02	50000.00

顧客02

請求書番号	請求日	合計
3	2009/02/02	12500.00

図：アプリケーションの実行

ステップ 19 : パターンインスタンスを変更してウェブパネルを再び実行

顧客トランザクションの「Work With 顧客」パターンインスタンスを変更して、「InvoicesPerCustomer」ウェブパネルを呼び出せるようにしましょう。

1. **Customer (顧客)** トランザクションを開いて、Patterns エレメントを選択し、パターンインスタンスを開いてください。

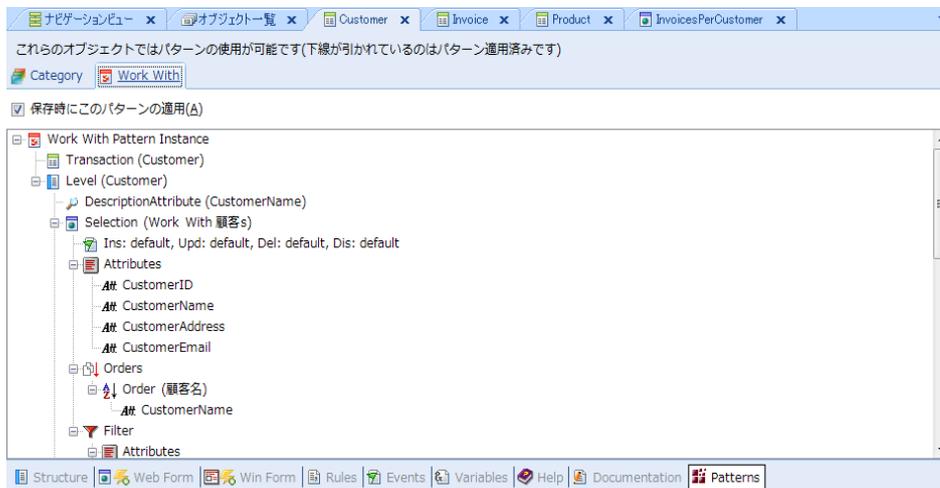
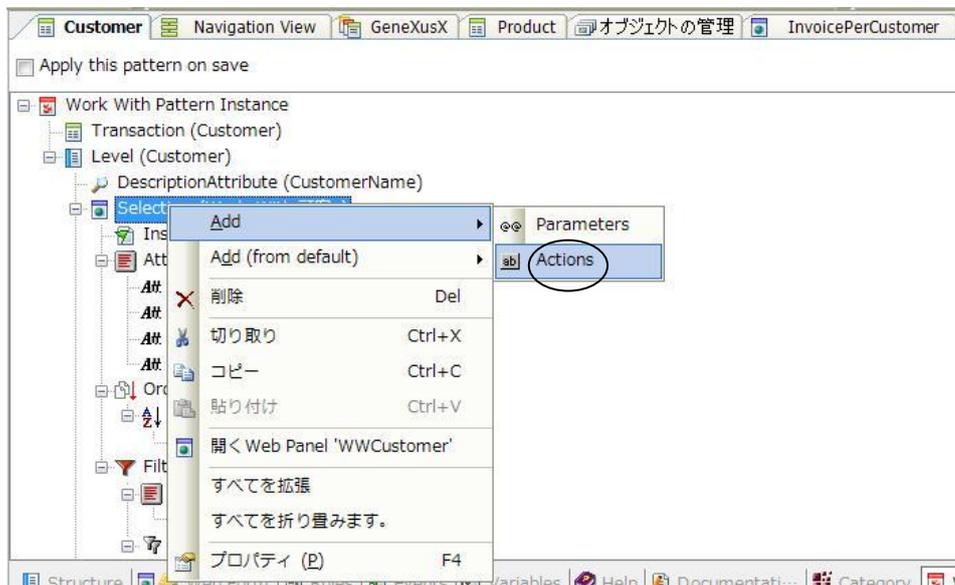


図 : Customer(顧客)トランザクション

2. このパターンのインスタンスにアクションを追加し、グリッドの外にボタンを配置してアクションを結びつけます。アクションは「Invoices per Customer (顧客ごとの請求書)」ウェブパネルを呼び出しましょう。



3. 「Selection(Work With 顧客s)」オプションを右クリックして、「Add」→「Actions」オプションを選択してください。

図: Work with 顧客s

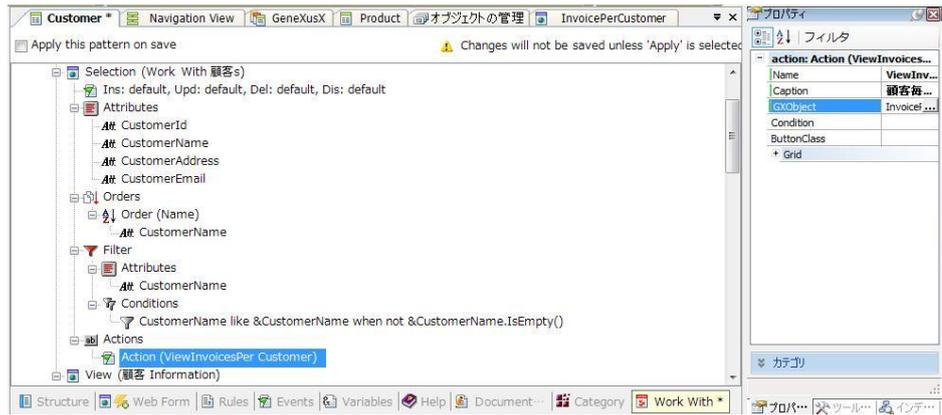
4. Action を右クリックし「Add」→「Action」オプションを選択し、新しいアクション追加してください。
5. F4 を押して、プロパティを編集してください。

Name に

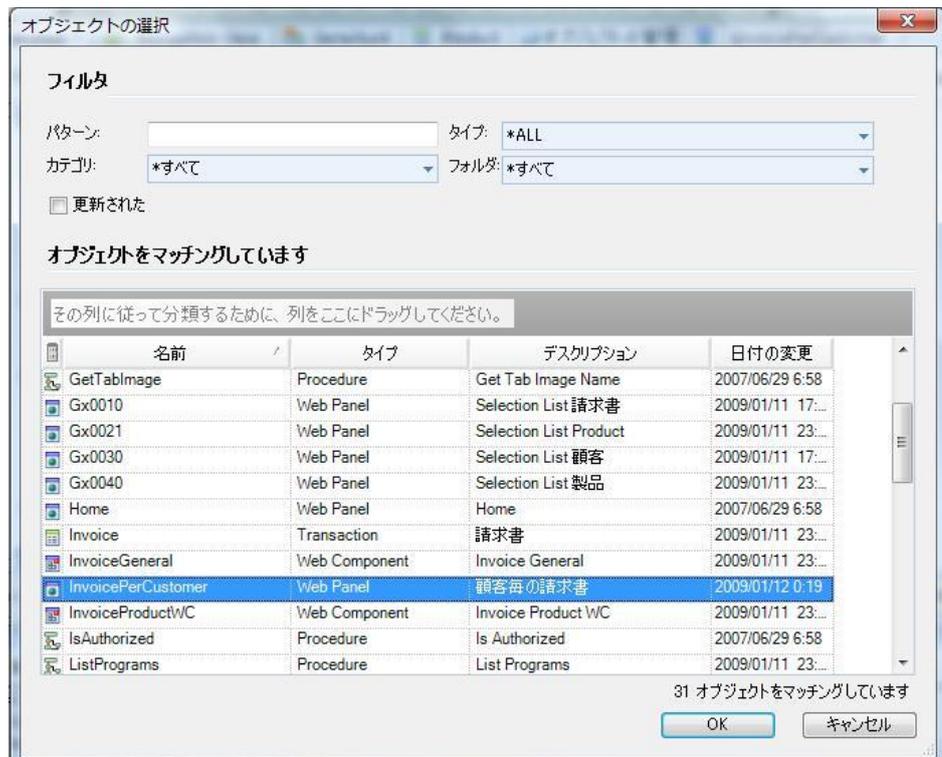
「ViewInvoicesPer Customer」を入力して、Caption には「顧客毎の請求書」を入力してください。

GXObject では InvoicesPerCustomer ウェブパネルを指定します。

6. 変更を保存してください。
7. F5 を押して、アプリケーションを実行してください。



図：パターンインスタンスの変更



図：オブジェクトの選択

8. 実行されると、作成されたウェブパネルを呼び出すボタンが追加された Work With 顧客を選択してください。



図：アプリケーションの実行

パート6：アプリケーションの改良

	推定された時間：15分
---	-------------

ユーザコントロール

!	<p>ユーザコントロール</p> <p>ユーザコントロールはアプリケーションの機能と外観を著しく高め、どの GeneXus ウェブオブジェクトのフォームにでも挿入することができます。提供される標準のユーザコントロール（Chart、Image、Gallery、Map、Treeview）とは別に、GeneXus はユーザコントロールエディターを提供しています。このエディターを使って、既存のユーザコントロールを変更し、独自のユーザコントロールを作成することもできます。</p>
----------	--

製品の売上チャートを表示させましょう。

この要件を満たすために、**GxChart** というユーザコントロールを使用しましょう。

ステップ 2 0 : ユーザーコントロールの定義

1. SalesxProd と呼ばれる Web Panel 型のオブジェクトを新規作成してください。デスクリプションには「製品売り上げ」と入力してください。

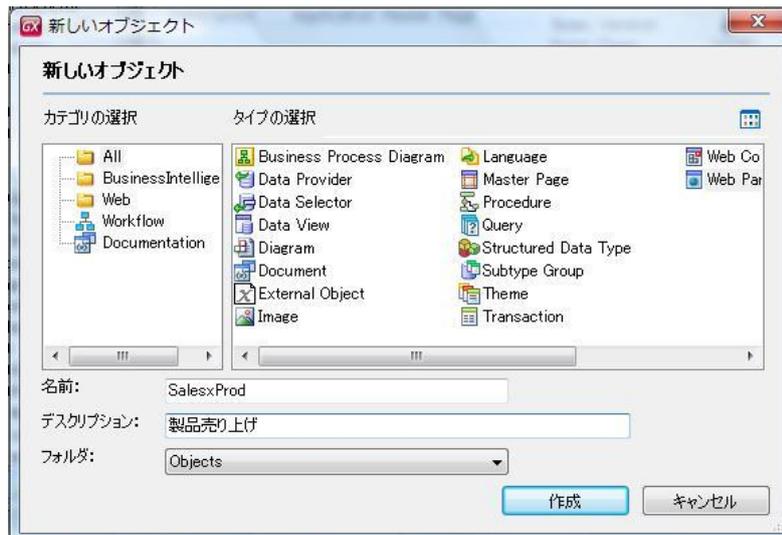


図 : SalesxProd ウェブパネル

2. 「表示」メニューから「他のツールウィンドウ」のツールボックスを表示させてください。

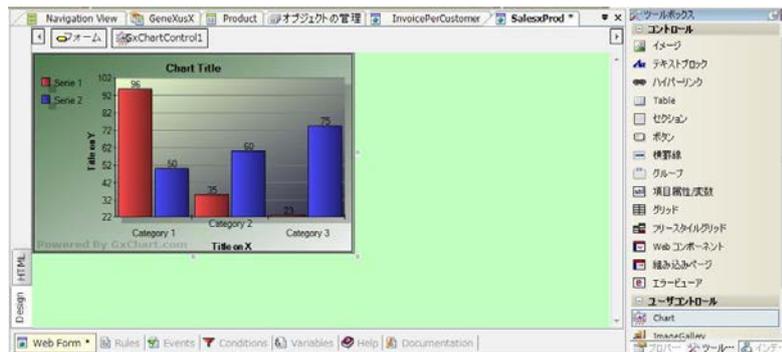


図 : GxChart コントロールを配置した WebForm エlement

3. ウェブパネルに「GxChart」ユーザーコントロールをドラッグしてください。

4. Events エlement内に自動生成されたコードは削除して、次のコードを記述してください。



図 : Events Element

Event Start

```
GxChartControl1.Title = 'ProductSales'
```

For Each

```
&GxChartData.Categories.Add(ProductName)
```

```
&GxChartSerie.Values.Add(InvoiceSubTotal)
```

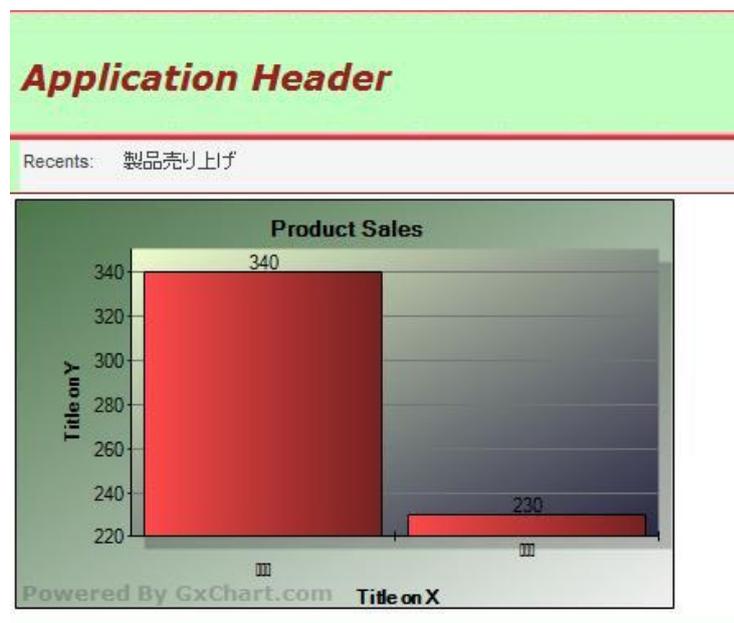
EndFor

```
&GxChartData.Series.Add(&GxChartSerie)
```

EndEvent

5. ウェブパネルを保存してください。

6. F5 を押して、アプリケーションを実行してください。



図：アプリケーションの実行

おめでとうございます！

GeneXus で最初のアプリケーションの作成に成功しました。

次の段階では、GeneXus のもう一層深い機能を紹介していきます。

パート7 : プロシージャー



推定された時間 : 15 分

インタラクティブではないプロセスの設計(プロシージャー)

!	<p>プロシージャー</p> <p>これまでに、データベースにデータの追加、更新及び削除するためにユーザーのアクションを必要とするトランザクションオブジェクトを作成してきました。</p> <p>しかしながら、ユーザーのアクションに頼らずに自動的に処理を実行するケースが必ずあります。その場合には、別の GeneXus オブジェクトを使用します :</p> <ul style="list-style-type: none"> • プロシージャー : プロシージャーは、インタラクティブではないデータベースの読み取り及び更新を行う処理を定義するオブジェクトです。 プロシージャーは、関数及びサブルーチンを定義するために使われます。 <p>GeneXus プロシージャーの主な機能定義は、トランザクションオブジェクトを定義した時と同様にナレッジベースに基づく方法論が使われます。</p> <p>プロシージャーオブジェクトの定義は、項目属性の値が保存される物理データベースのテーブルレイアウトに基づいているのではなく、ナレッジベースに登録された項目属性の名前に基づきます。</p> <p>したがって、プロシージャー内に記述される項目属性がナレッジベースにある限りプロシージャーの定義は有効であり続けます。</p>
---	--

!	<p>プロシージャーの特徴</p> <ul style="list-style-type: none"> • 高レベルのプロセス : プロシージャーは、非常に高度な手法で記述されます。実行順序は、設計者が記述するコントロール、印刷、データへのアクセス及び他のコマンドを含む単純なプロセス言語によって決定されます。 • ナレッジベースに基づく : プロシージャーのソースに書く項目属性は、物理データベースのテーブルを直接参照するのではなくナレッジベース内に定義されている項目属性名を参照します : <ul style="list-style-type: none"> ○ 項目属性の物理的な場所は GeneXus が推論します。 ○ GeneXus は、物理データベースのテーブル間の関連を保持しています。 ○ 式項目属性は、GeneXus により自動的に推論・計算されます。 <p>物理データベースのレイアウトが変化しても、プロシージャーの振る舞いには影響を与えません。</p>
---	--

合計と共に全ての顧客一覧を印刷するプロシージャーと製品を 10%値下げするプロシージャーを作成しましょう。

ステップ 2 1 : 印刷するプロシージャーの定義

1. オブジェクトフォルダを右クリックして「新規」→「オブジェクト」を選択してください。
2. 作成する「Procedure」型オブジェクトを選択してください。
3. 名前には“Customer_Report”を入力してください。そしてデスクリプションに、“顧客一覧”と入力してください。
4. 作成をクリックしてください

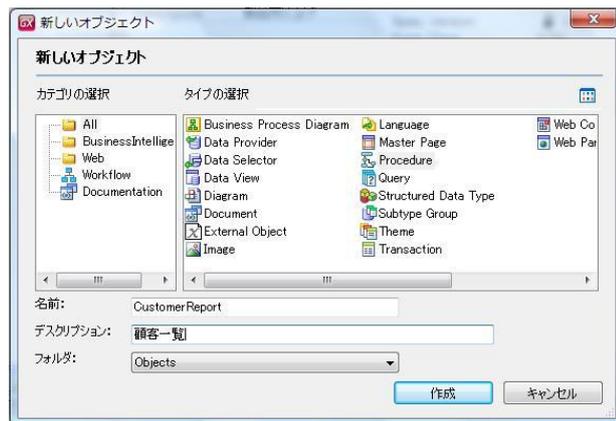


図 : Customer_Report プロシージャー

1. 以下のソースを Source エlement に記述してください :

```
For each
    &Total=SUM(InvoiceTotal)
    Print printBlock1
Endfor
```

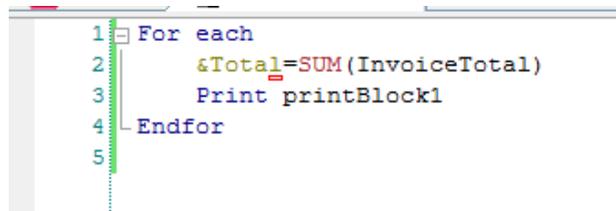


図 : Customer_Report プロシージャーの Source Element

2. Layout Element へ移動して、ツールバーのメニューから「挿入」→「項目属性」から CustomerName(顧客名)の項目属性を挿入してください。

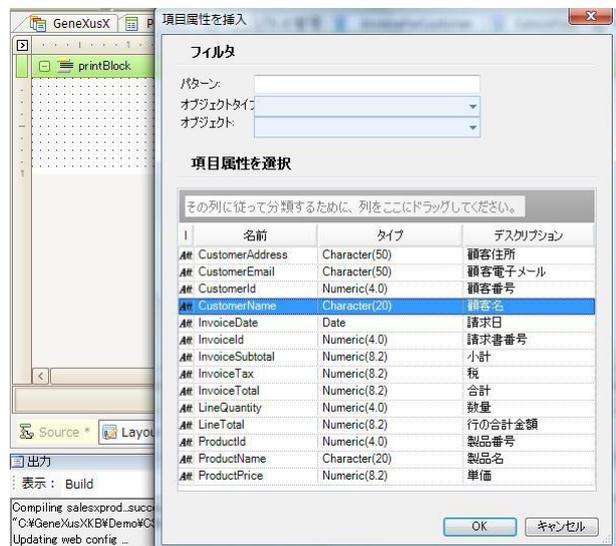


図 : プロシージャーのレイアウト

3. ツールバーのメニューから「挿入」→「変数」を選択し、変数の挿入画面を呼び出した後、画面左下にある「新規変数」ボタンを押下してください。

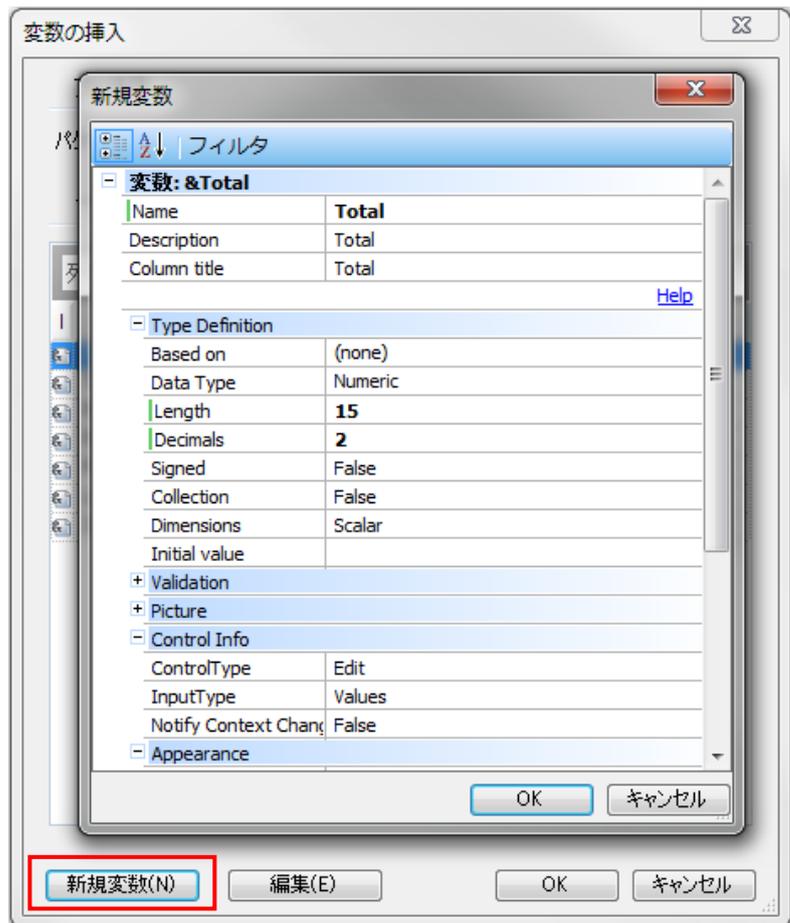
新規変数ウィンドウが起動しますので、以下のプロパティを設定してください。

Name : Total

Length : 15

Decimals : 2

プロパティ設定後、「OK」ボタンを押下し、変数を定義してください。



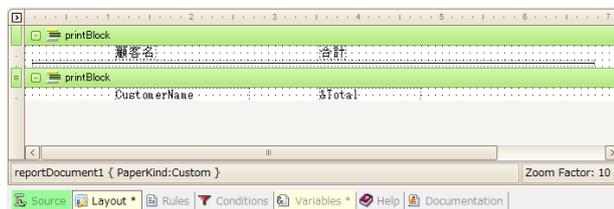
図：変数の挿入

4. ツールボックスを使用してレポートを設計することができます（ラベル、ライン等の挿入）。

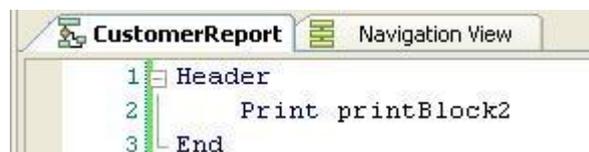
例えば、PrintBlock1 を右クリックし「プリントブロックを挿入」で、もう一つの PrintBlock を挿入して「顧客名」及び「合計」の見出しをツールボックスの TextBlock コントロールを使用して挿入することができます。

それから、マウスで右クリックして表示されるオプションを使用して PrintBlock2 を上方へ移動してください。

右記のコードを Source エlement に記述してください：



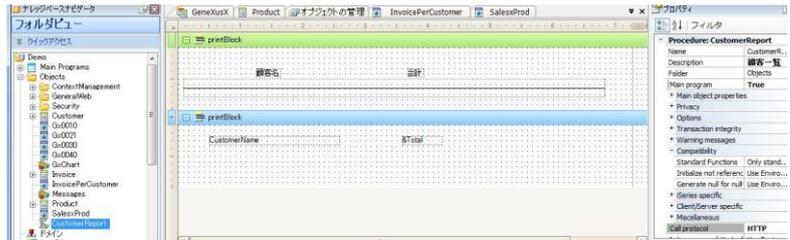
図：プロシージャーの Layout エlement



図：プロシージャーの Source Element への追記

5. プロシージャーを保存してください。

6. 右に表示されるプロシージャー自体のプロパティ内にある「Main Program」プロパティを TRUE に設定し、「Call Protocol」を HTTP に設定してください。



図：プロシージャーのプロパティ

7. ファイル名と拡張子を指定するために、Rules エlementに以下のコードを記述し、保存してください。

```
Output_file('Customers.PDF','PDF');
```

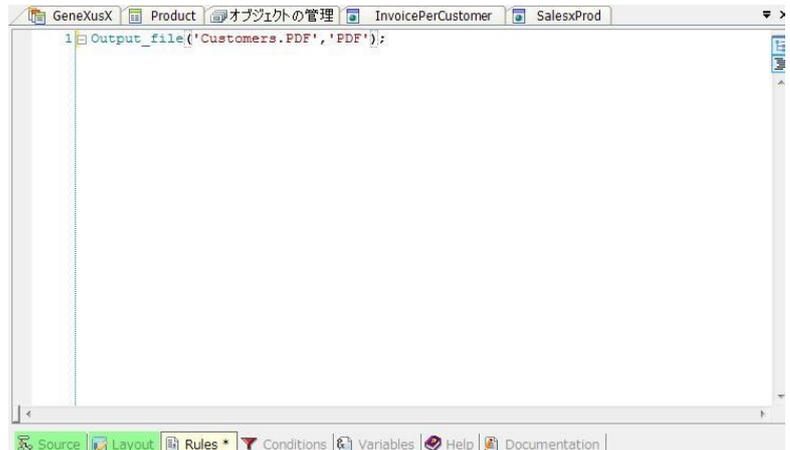
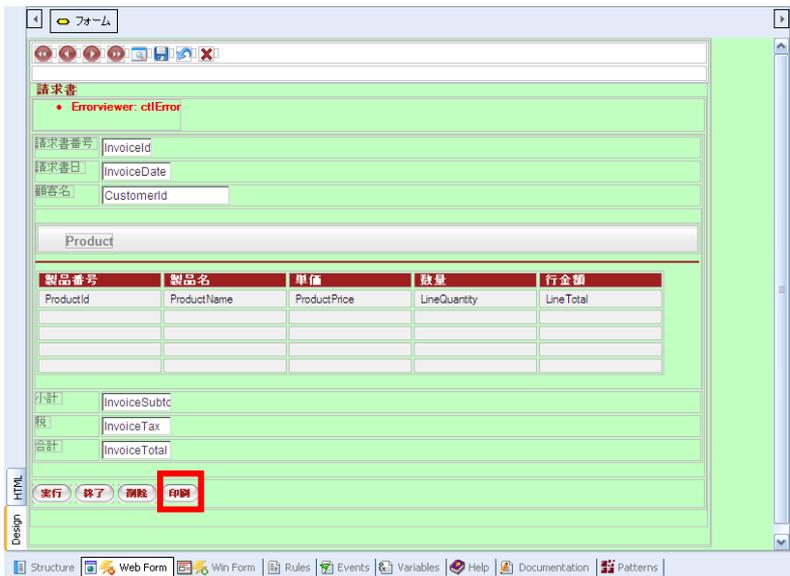


図:プロシージャーの Rules Element

8. 顧客ごとに合計を印刷するこのプロシージャーを呼び出すために、Invoice トランザクションにボタンを挿入してください。

まず Invoice（請求書）トランザクションオブジェクトを開きます。

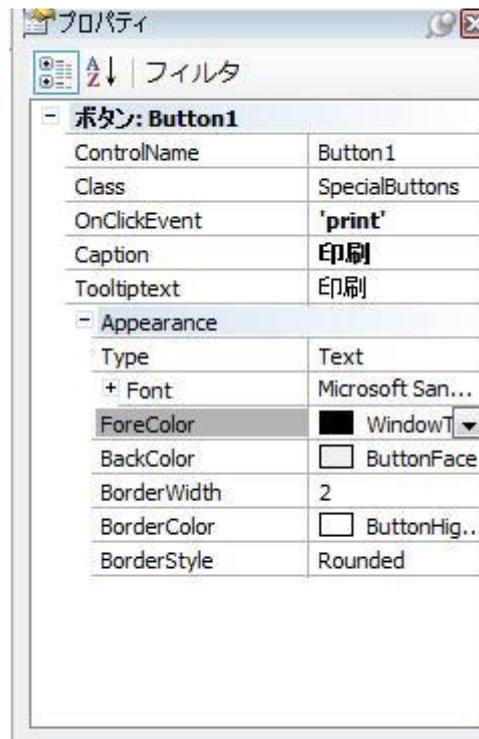
Web Form Elementを選択し、ツールボックスを開いてボタンコントロールをドラッグしてください。



図：Invoice（請求書）トランザクション

9. 挿入されたボタンを右クリックして、プロパティを編集してください。

「OnClickEvent」には「Print」と入力し、Captionに「印刷」を入力してください。ボタンを右クリックして「イベントへ行く」を選択してください。



図：ボタンのプロパティ

10. Events エlementには以下のコードを追加してください:

```
Event 'Print'
    Customer_Report.Call ()
EndEvent
```



図：トランザクションの実行

11. 保存し、F5 を押下して実行してください。

入力された新しい機能を試すために、新しい請求書を登録してください。

顧客名	合計
顧客01	62500.00
顧客02	12500.00

図：プロセスの実行

ステップ 2 2 : アップデートプロシージャーの実施

製品の価格を 10%値下げするプロシージャーを作成しましょう。

1. 「名前 : Discount_of_prices ; デスクリプション : 価格値下げ」 Procedure 型のオブジェクトを新規作成してください。デスクリプションには「価格値下げ」と入力してください。

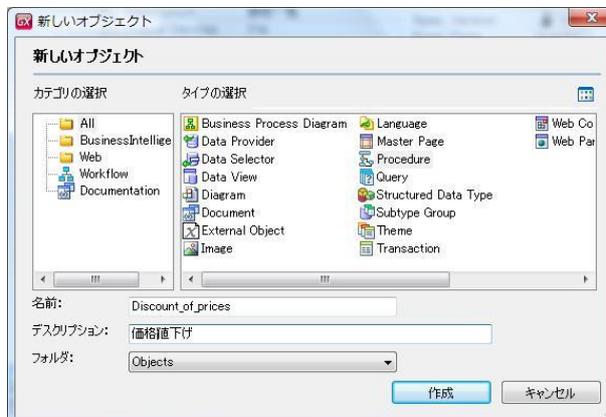


図 : 価格値下げプロシージャー

2. Source エlementに以下のコードを記述してください :

For each

 ProductPrice = ProductPrice-ProductPrice*0.10

Endfor

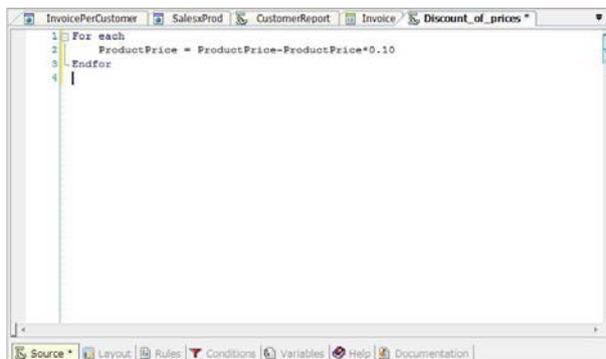


図 : プロシージャーの Source Element

3. プロシージャーを保存してください。

4. このプロシージャー を呼び出すために、ウェブパネルを新規作成してください。

ウェブパネルの名前に「Control_Panel」を入力してください。デスクリプションには「制御画面」と入力してください。



図 : Control_Panel ウェブパネル

5. Web Form エlementでツールボックスからボタンを挿入してください。

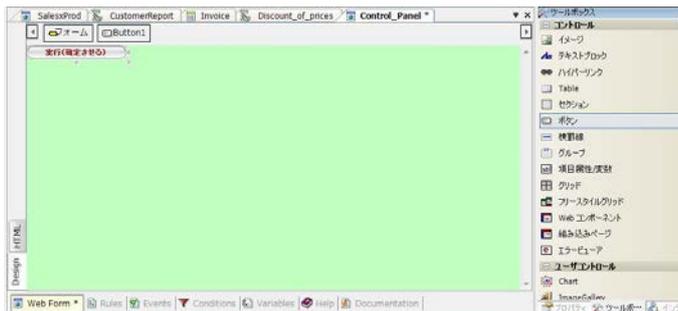
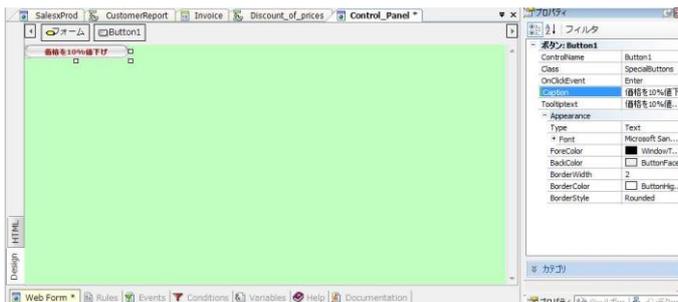


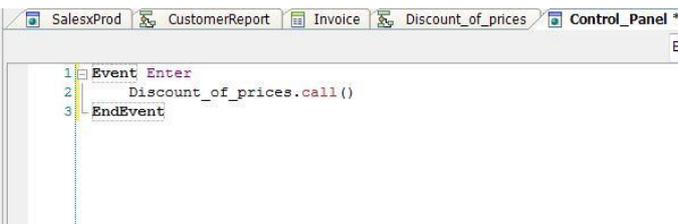
図 : Web Form Element

6. 挿入されたボタンを右クリックしてプロパティを編集してください。



キャプションに「価格を10%値下げ」と入力してください。

7. ボタンを右クリックして、「イベントへ行く」オプションでイベントへ移動してください。



プロシーチャーをイベントから呼びましょう :

```
Event Enter
Discount_of_prices.call()
EndEvent
```

図 : Events Element

8. ウェブパネルを保存し、F5 を押して実行してください。



図 : アプリケーションの実行

パート8 : マルチプラットフォーム

	推定された時間 : 5 分
---	---------------

マルチプラットフォーム開発

GeneXus はミッションクリティカルなアプリケーションを、複数のプラットフォームで生成することが可能です。

一旦アプリケーションが一つの環境で稼働した後、別の環境で同じアプリケーションを稼働させるには、新しい環境を定義することにより、簡単に対応することが可能です。

注：GeneXus トライアルバージョンは、GeneXus .NET ジェネレータのみを提供します。

GeneXus の正規版は市場の主要なプラットフォームをサポートしています。GeneXus がサポートする技術の一覧は次の URL でご覧ください <http://www.genexus.com/technologies>。

要約

このチュートリアルで GeneXus の主な機能と利点を経験することができたのではないかと思います。

知識を基にするアプリケーション設計

ナレッジベースを作成することから始まり、アプリケーションの要件をナレッジベースのトランザクションオブジェクトと呼ばれる一連のビジネス部分として記述しました。

GeneXus は、記述された情報を基に全てのトランザクションに対応する最適のデータモデル（第三正規形）を推論しました。追加したビジネスルールを基に、GeneXus はアプリケーションのビジネスロジックのコードを生成しました。

データベースの自動生成

特定の実行プラットフォームを設定して（ウェブ、プログラミング言語、DBMS、その他）、GeneXus は自動的に指定された DBMS サーバに物理のデータモデルを作成しました。

コードの自動生成と最適なプロトタイプ

アプリケーションプログラムのソースコードを生成し、プロトタイプ環境でアプリケーションをテストすることができました。

アプリケーションの保守

GeneXus オブジェクトの修正・追加によるデータモデルの変更が簡単に物理データベースに反映され、影響を受けたプログラムは再度自動生成されることでアプリケーションが完成することが実感できたと思われれます。

インタラクティブでないプロセスの設計

アプリケーションのデータベース構造を意識しないで良いプロシージャの全体的なイメージを提供できたと思います。

マルチプラットフォーム開発

最後には GeneXus アプリケーションの環境を簡単に移動できることを説明しました。

FAQ

質問	GeneXus は、ソースコードを 100%自動生成してくれるのでしょうか？
-----------	---

はい、GeneXus は、アプリケーションのデータベースとソースコードを設計、生成及び保守します。

質問	生成されるソースコードにはランタイムが必要でしょうか？
-----------	------------------------------------

いいえ、GeneXus は選択されたプラットフォームでネイティブコードを生成するため独自のランタイムは必要ありません。

質問	既存のデータベースを使用することは可能ですか？
-----------	--------------------------------

はい、可能です。GeneXus は、既存のデータベース構造を読み取る Data Base Reverse Engineering Tool(DBRET) というツールを持っています。

質問	生成されたソースコードを変更することは可能ですか？
-----------	----------------------------------

はい、可能ですが推奨できません。理由は大きく分けて二つあります：再度 GeneXus でソースを生成すると、ソースは上書きされてしまいます。また、独自のソースを書き込むことによりアプリケーションがその言語に依存してしまいます。

もし既存のコードを使用されるのならば、GeneXus の「アプリケーションの統合」オプションを使用して外部機能とのインターフェースをとります。生成されたコードに直接手を加える必要はありません。

質問	アプリケーションでコントロール (Flash /Silverlight/JavaScript) を統合することは可能ですか？
-----------	---

はい、可能です。GeneXus は様々なコントロールの統合を可能にしています。

詳細は以下のリンクをご覧ください。

<http://marketplace.genexus.com>

<http://wiki.genexus.jp/UC>

質問

拡張テーブルの概念は何ですか？

ベーステーブルがあれば、その**拡張テーブル**とは直接あるいは間接的にベーステーブルの（参照）キーを使って持つてくる事が出来る項目属性のセットです：

- ベーステーブルに属している項目属性。
- ベーステーブルと「N 対 1」の関係で直接あるいは間接的に関連がある全てのテーブルに属している項目属性。

使用例：

- データベース構造が変更されても拡張テーブルの概念により GeneXus のレポートとプロシージャが有効であり続けることが可能です。
- 拡張テーブルの概念は、トランザクションの Structure から参照されているベーステーブルの拡張テーブルに属している項目属性の登録、更新及び削除をトランザクションオブジェクトが行うのにも使用されます。

あるテーブルの拡張テーブルを見るためには、ナレッジベースのデータモデルのバツハマン図を使いましょう：

1. オブジェクトの新規作成で「Diagram」型のオブジェクトを作成してください。
2. 表示されたテーブルを選択して、ダイアグラムをドラッグしてください。

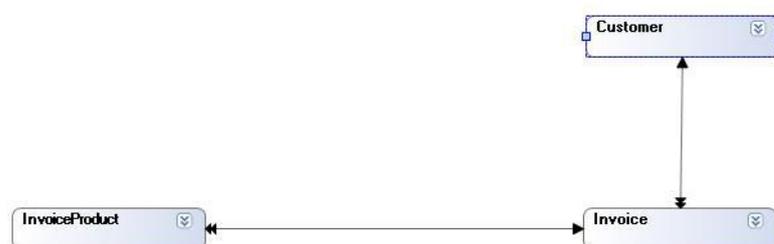


図 3：データモデルのバツハマン図

この例において、プロジェクトで各テーブルの拡張テーブルを特定することができます：

ベーステーブル	拡張テーブル
Customer	Customer
Invoice	Invoice, Customer
InvoiceProduct	InvoiceProduct, Invoice, Customer

質問

"For Each"コマンドとは何ですか？

"For Each"コマンドとは、GeneXus で帳票や処理を記述する時に使う手続き型言語の重要なコマンドです。

このコマンドを使用して、データを読み取ったり更新したりするオンライン処理ができます。

"For Each"コマンドでアクセスするデータを決めて、必要な項目属性を記述します。

GeneXus は、生成時に使用するテーブルを自動的に推論します。

データベース構造が変わっても「For Each」の定義は有効であり続けます。

"For Each"の基本的な構文は以下の通りです：

```
for each
    print <something>
endfor
```

各「For Each」コマンドは、ナビゲートするデータモデルのテーブルを必ず持っています。

ナビゲートするテーブルは、**"For Each"のベーステーブル**及びその**拡張テーブル**（コマンドに含まれる項目属性を取得するために必要とするテーブル）になります。

アクセスされるテーブルは、"For Each"コマンドの中に含まれる項目属性により決定されます。

- "For each"内の項目属性に対してGeneXusは最小限の拡張テーブルを推論します。

"For each"内の項目属性に対してのベーステーブルは"For Each"のベーステーブルであります。

コンタクトとリソース

GeneXus コミュニティ

GeneXus コミュニティは質問に対する回答をしたり、問題に対する解決策を出したり、経験の共有をする場所を提供したりしています。

以下のサイトで提供されているコミュニティリソースの一覧を見ることができます：

<http://www.genexus.com/community/>（英語）

サポート

Artech 社は、以下の2種類のサポートリソースとサービスを提供しています：

- **オンラインセルフサービスサポート**

これらのリソースは、誰でもオンラインで利用できます。

しかしながら、アクセスするデータは GXtechnical のユーザロールによって異なります（2種類のロールがあります：一般ユーザもしくは顧客）。

- **インタラクティブサポートサービス**

コミュニティメンバー又はサポートチームとのインタラクション（情報交換）ができます。

<http://www.genexus.com/support/>（英語）をご覧ください。

購入方法

GeneXus Technologies は、世界的な総代理店ネットワークを通して販売されています。

<http://www.genexus.com/distributors> で最も近い総代理店を検索してください。あるいは、sales@genexus.jp にお問い合わせください。