

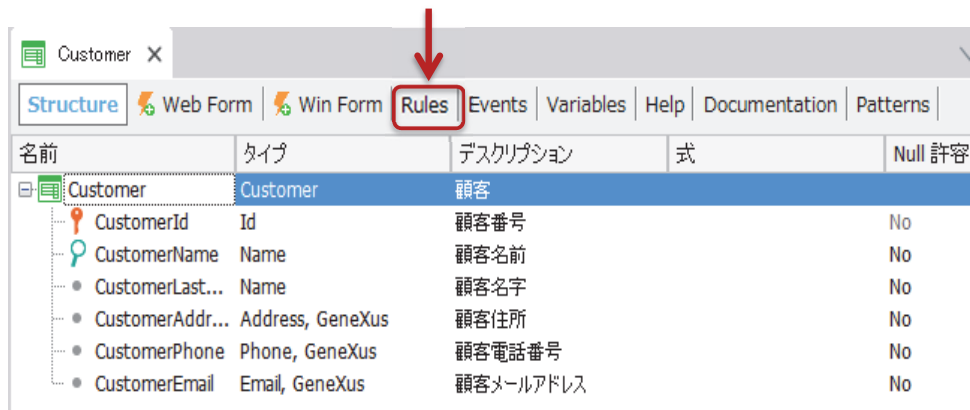
ユーザーから要望のあった 制御の定義

トランザクションのルール

GeneXusTM

GeneXus では、アプリケーションの生成時に自動的に追加される制御のほかに、ユーザーからの検証の要望に応じて特定の制御を追加できます。

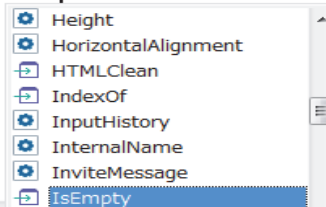
顧客の登録に関する制御



トランザクションでは、従うべきルールや、検証の要望のあった制御を、[Rules] セクションで定義します。

顧客の登録に関する制御

```
1 Error("名前の入力は必須です")
2   if CustomerName.IsEmpty();
3
4 Error("名字の入力は必須です")
5   if CustomerLastName.IsEmpty();
6
7 Msg("電話番号が未入力です") if CustomerPhone.i
```



ルールは定義した順序で実行されるわけではない

Error ルール:

フィールドから移動したときに条件式が成立した場合、メッセージが表示されます。引き続きほかのフィールドにデータを入力できますが、条件が満たされている間はデータを保存できません。これが既定の動作ですが、条件が満たされている間は次のフィールドに移動して編集できないように変更することもできます。動作を変更するにはプロパティを使用しますが、このコースでは取り扱いません。

Message ルール:

条件式が成立した場合、通知または警告のメッセージを表示します。ただし、引き続き操作することができます。Error ルールとは異なり、データを保存することもできます。

デモ

- ここまでに説明した内容を定義する
- アプリケーションを実行する (F5)
 - データを入力して、追加した制御をテスト



項目属性と変数

- **項目属性:** 値をデータベースに物理的に格納する
- **変数:** 値を、関連付けした名前およびデータ型とともにメモリに一時的に格納する。変数を参照する場合は「&」記号を使用する



例: 変数 &FirstName は、メモリに「John」という値を格納する

変数とは、関連付けられているシンボリック名および格納可能なデータ型 (テキスト、数値、日付など) を持つ、メモリ内のスペースです。変数には特定の値が格納されます。通常は、変数名を使用して、この格納されている値を参照します。

ほとんどの GeneXus オブジェクトで変数を定義できます。これらの変数は**ローカル**変数であり、各オブジェクト内でのみ使用できます。変数を参照するには、「&」記号を使用する必要があります。たとえば、&Total のようにします。

項目属性と変数の相違点:

項目属性は、値をデータベースに物理的に格納します。

変数は、オブジェクトの実行中に、値をメモリに一時的に格納します。

トランザクション内で **[Variables]** エレメントを開くと、一連の変数が既に定義されています。これらは &Today、&Mode などのシステム変数です。変数 &Today は、現在のシステム日付を取得できます。

開発者は、これらのシステム変数に加えて、独自の変数 (ユーザー変数) も作成できます。

変数の定義

変数は、それが使用されるオブジェクト内でのみ定義できる

1) [Variables] エlementを使用:

The screenshot shows the 'Variables' window in GeneXus. It has a menu bar with 'Structure', 'Web Form', 'Win Form', 'Rules', 'Events', 'Variables *', 'Help', 'Documentation', and 'Patterns'. Below the menu is a table with columns: '名前' (Name), 'タイプ' (Type), 'Is Collection', and 'DESCRIPTION'. The table is divided into two sections: 'Standard Variables' and 'New Variable'.

名前	タイプ	Is Collection	DESCRIPTION
Standard Variables			
GxRemove	Numeric(1.0)	<input type="checkbox"/>	Gx Remove
Mode	Character(3)	<input type="checkbox"/>	Mode
PgmDesc	Character(256)	<input type="checkbox"/>	PgmDesc
PgmName	Character(128)	<input type="checkbox"/>	PgmName
Time	Character(8)	<input type="checkbox"/>	Time
Today	Date	<input type="checkbox"/>	Today
New Variable			
NewVariable	Numeric(4.0)	<input type="checkbox"/>	New Variable

Annotations in the image:

- An arrow points from the 'Standard Variables' section to the text: システムで事前に定義されている変数 (Variables predefined in the system).
- An arrow points from the 'New Variable' section to the text: 開発者が定義した変数 (Variable defined by the developer).

2) 宣言時に定義:

Event Start
&FromDate
Endevent

右クリック

変数を追加

関数(F)...
イベント(E)...
コマンド(C).....

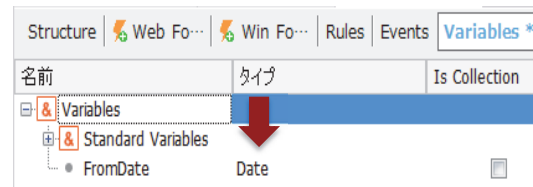
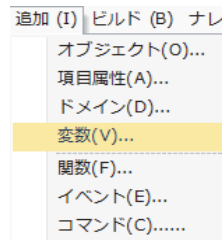
変数はローカルスコープです。つまり、その変数が使用されるオブジェクト内でのみ定義できます。

変数は、次のいずれかの方法で定義できます。

- 1) すべての GeneXus オブジェクトにある [Variables] Elementを使用して定義します。
- 2) オブジェクトの[Events]、[Source]などで宣言時に定義します: 「&」の後に変数の名前を入力し、右クリックします。[変数を追加] を選択します。
- 3) メニューバーの追加オプションを選択します: 次に、[変数] を選択します。

変数の定義

3) [追加] → [変数]:



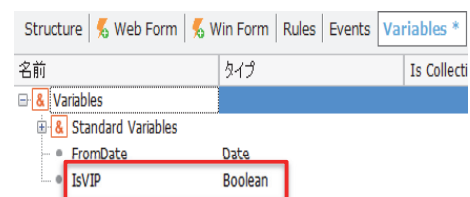
4) 名前のパターンに従って自動的に定義:

a) ドメイン: 名前

&FromDate - 自動的に Date として定義

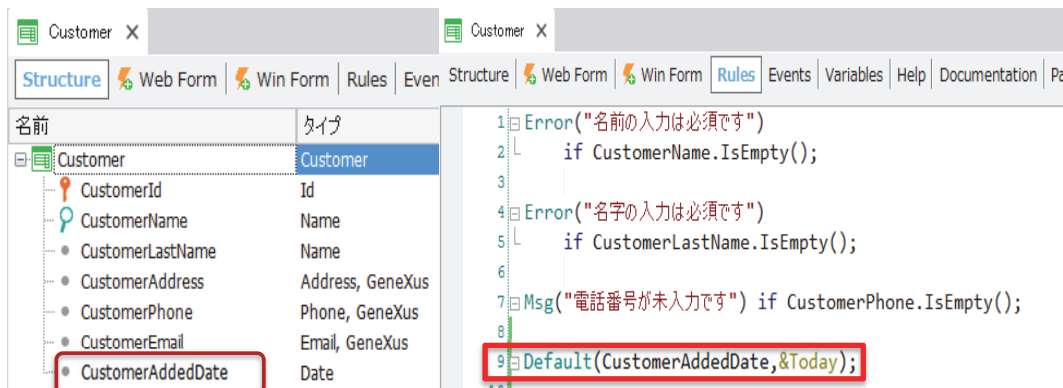
b) **&IsXxx** | **&HasXxx**

→ 自動的に Boolean として定義



さらに制御が必要な場合

- 「各顧客がシステムに登録された日付を記録」（既定値を設定）



- 日付フィールドは編集可能のまま、未来の日付を入力できないように制御

```

Error("追加日は今日以前でなければなりません")
if CustomerAddedDate > &Today;

```

この例では、Default ルールを使用して、現在の日付を [CustomerAddedDate] 項目属性に割り当てています。

Default ルールを宣言せずに、新しい顧客を追加した場合、CustomerAddedDate は空の値になります。ほかの項目属性でも同様です。

Default ルールは、レコードが**挿入**されたときに**のみ**実行されます。

つまり、保存済みの顧客を照会した場合は Default ルールは実行されません。

これは、その顧客には**既に登録日が記録されており**、Default ルールがそれを上書きすることはないためです。

フィールド/項目属性の編集不可

要望:

- 登録日フィールドの編集を許可**しない**。このフィールドは編集不可状態で表示され、顧客情報を登録時にシステムが指定する値を保存するようにする。

```
Error("追加日は今日以降でなければなりません")  
if CustomerAddedDate > &Today;
```



The screenshot shows the 'Rules' tab in the GeneXus IDE. The rules list contains the following code:

```
1 Error("名前の入力必須です")  
2   if CustomerName.IsEmpty();  
3  
4 Error("名字の入力は必須です")  
5   if CustomerLastName.IsEmpty();  
6  
7 Msg("電話番号が未入力です") if CustomerPhone.IsEmpty();  
8  
9 Default(CustomerAddedDate,&Today);  
10  
11 NoAccept(CustomerAddedDate);
```

The rule on line 11, `NoAccept(CustomerAddedDate);`, is highlighted with a red rectangular box.

デモ



顧客の入力日フィールド (CustomerAddedDate) は、Default ルールで**初期化**され、NoAccept ルールで**入力不可**の状態が表示されます。

Assignment ルール

CustomerAddedDate = &Today;

ユーザーがデータを挿入しているか
更新しているかに関係なく常に実行

CustomerAddedDate = &Today **if Insert;**

新しいレコードの挿入時にのみ実行。これは
Default ルールと同等の動作

このルールを指定すると、CustomerAddedDate 項目属性に現在の日付が**常に割り当てられます**。この **Assignment ルール**は、ユーザーがデータを挿入しているか更新しているかなどに関係なく、常に実行されます。

Assignment ルールに条件を追加して、ユーザーがデータベース内で特定の操作(挿入、更新、削除など)を実行しているときにのみ処理が実行されるようにすることができます。

これまでに説明した内容:

- ルールを使用してトランザクションの動作をプログラミングできる
- ルールは、宣言されたトランザクションでのみトリガーされる
- ルールがトリガーされる順序は GeneXus が決定し、その順序はルールが記述された順序と必ずしも一致しない
- ルールには、トランザクションの操作モード (**挿入**、**更新**、**削除**) などの条件式を指定することができる

ルールのトリガー順序

ルールと式が実行される順序

- GeneXus のアナリストは、トランザクションに作成したルールと式の実行順序を指定しない
- 生成中に、ルールと式の間が存在する依存関係を GeneXus が判断する

トランザクションの動作は、ルールを宣言することでプログラミングします。計算が必要な場合は、式の項目属性を作成できます。

GeneXus のアナリストは、トランザクションに作成したルールと式の実行順序を指示しません。生成中に、ルールと式の間が存在する依存関係を GeneXus が判断します。

トランザクションと関連付られたプログラムの生成中に、ルールと式の間が存在する依存関係を GeneXus が抽出します。GeneXus は、依存関係ツリー (**評価ツリー**) を論理的に作成し、それによって評価順序を決定します。