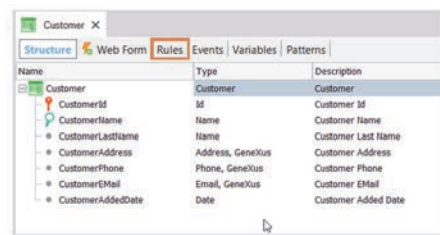


ルール: 評価および クライアント側検証

GeneXus[™]

ルール



Basic コースでは、Transaction オブジェクトに [Rules] というエレメントがあることを学習しました。このエレメントでは、制御や、従うべきルールを定義します。

ルール

The screenshot displays two components of the GeneXus development environment. On the left, a 'Customer' window shows the 'Rules' tab with the following code:

```
1 Error('Enter the customer name')
2 if CustomerName.IsEmpty();
```

On the right, a 'Customer' web form is shown. It contains fields for Id, Name, Last Name, Address, Email, Phone, and Added Date. The 'Name' field is highlighted with a red border, and a yellow tooltip with the text 'Enter the customer name' and an orange arrow points to it. At the bottom of the form, there are 'CONFIRM' and 'CANCEL' buttons.

特定の条件が満たされている間データベースにレコードが格納されないようにする **Error** ルールに焦点をあてました。たとえば、顧客の入力時に、名前を空のままにした場合、ユーザーが名前を入力するまでその顧客を挿入することはできません。

ルール

The screenshot displays the GeneXus IDE interface. On the left, the 'Rules' tab is active, showing a rule editor for a 'Customer' entity. The rule logic is as follows:

```
1 Error('Enter the customer name')  
2 if CustomerName.IsEmpty();  
3  
4 Msg('The phone is empty')  
5 if CustomerPhone.IsEmpty();
```

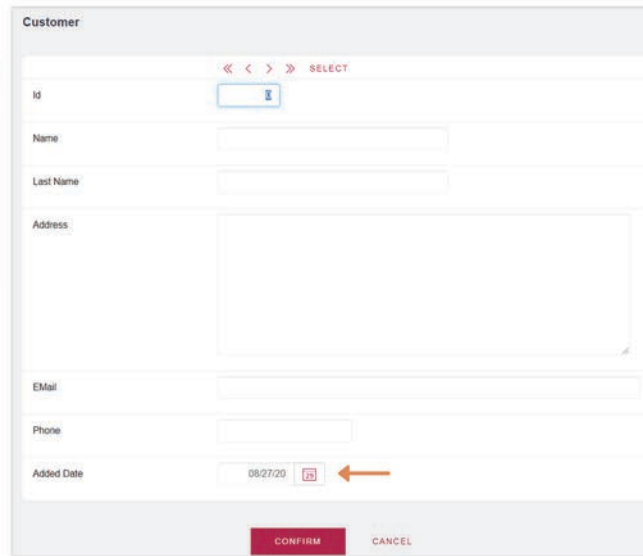
On the right, the 'Customer' web form is shown. It contains fields for Id, Name, Last Name, Address, Email, Phone, and Added Date. The 'Name' field is populated with 'Joseph'. The 'Phone' field is empty, and a yellow tooltip with the message 'The phone is empty' is displayed next to it, with an orange arrow pointing to the field. At the bottom of the form, there are 'CONFIRM' and 'CANCEL' buttons. A status bar at the top of the form indicates 'Data has been successfully added'.

また、メッセージをユーザーに伝えるのみで、保存は可能にする **Message** ルールも確認しました。

ルール



```
1 Error('Enter the customer name')
2   if CustomerName.IsEmpty();
3
4 Msg('The phone is empty')
5   if CustomerPhone.IsEmpty();
6
7 Default(CustomerAddedDate, &Today);
```



Customer

Id

Name

Last Name

Address

EMail

Phone

Added Date

CONFIRM CANCEL

Default ルールでは、挿入モードでトランザクションにアクセスする際に、項目属性または変数を初期化する値を指定できます。

ルール

The screenshot displays the GeneXus IDE interface. On the left, the 'Rules' tab is active for the 'Customer' entity, showing a list of rules:

- 1. Error('Enter the customer name')
- 2. if CustomerName.IsEmpty();
- 3.
- 4. Msg('The phone is empty')
- 5. if CustomerPhone.IsEmpty();
- 6.
- 7. Default(CustomerAddedDate, &Today);
- 8.
- 9. Noaccept(CustomerAddedDate);

On the right, the 'Customer' web form is shown. It includes fields for Id, Name, Last Name, Address, EMail, Phone, and Added Date. The 'Added Date' field is highlighted with an orange arrow, indicating it is the focus of the 'Noaccept' rule. The form has 'CONFIRM' and 'CANCEL' buttons at the bottom.

Noaccept ルールでは、フォーム内のフィールドを変更できないように、Disabled として表示されます。

ルール

The screenshot displays the GeneXus IDE interface. On the left, a 'Country' object is selected, and the 'Rules' tab is active. A rule is defined as follows:

```
1 Serial(CityId, CountryLastLine, 1);  
2  
3  
4  
5
```

Below the rule editor, the 'Structure' tab shows the object hierarchy:

- Country
 - CountryId
 - CountryName
 - CountryLastLine
 - City
 - CityId
 - CityName

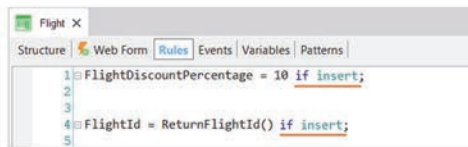
On the right, a web form for the 'Country' object is shown. It includes fields for 'Id' (0), 'Name' (Brazil), and 'Last Line' (2). Below these is a 'City' section with a table for 'Id Name':

	Id	Name
X	1	São Paulo
X	2	Rio de Janeiro
	0	
	0	

At the bottom of the form, there is a '[New row]' link and 'CONFIRM' and 'CANCEL' buttons.

最後の **Serial** ルールは、トランザクションの 2 番目、3 番目、またはその他のネストされたレベルに自動採番するために使用されます。

ルール



if insert

if update

if delete

また、ルールを使用して、値の割り当ての定義やオブジェクトの呼び出しを行ったり、追加、変更、または削除を行うときにのみ実行するように条件を付けたりすることも学習しました。

ルール

```

1 FlightDiscountPercentage = 10 if insert;
2
3
4 FlightId = ReturnFlightId() on BeforeInsert;
5

```

各行

VALIDATE the header
On BeforeInsert / BeforeUpdate / BeforeDelete
SAVE the header
On AfterInsert / AfterUpdate / AfterDelete
VALIDATE the line
On BeforeInsert / BeforeUpdate / BeforeDelete
SAVE the line
On AfterInsert / AfterUpdate / AfterDelete
On AfterLevel Level Line attribute
On BeforeComplete
COMMIT
On AfterComplete

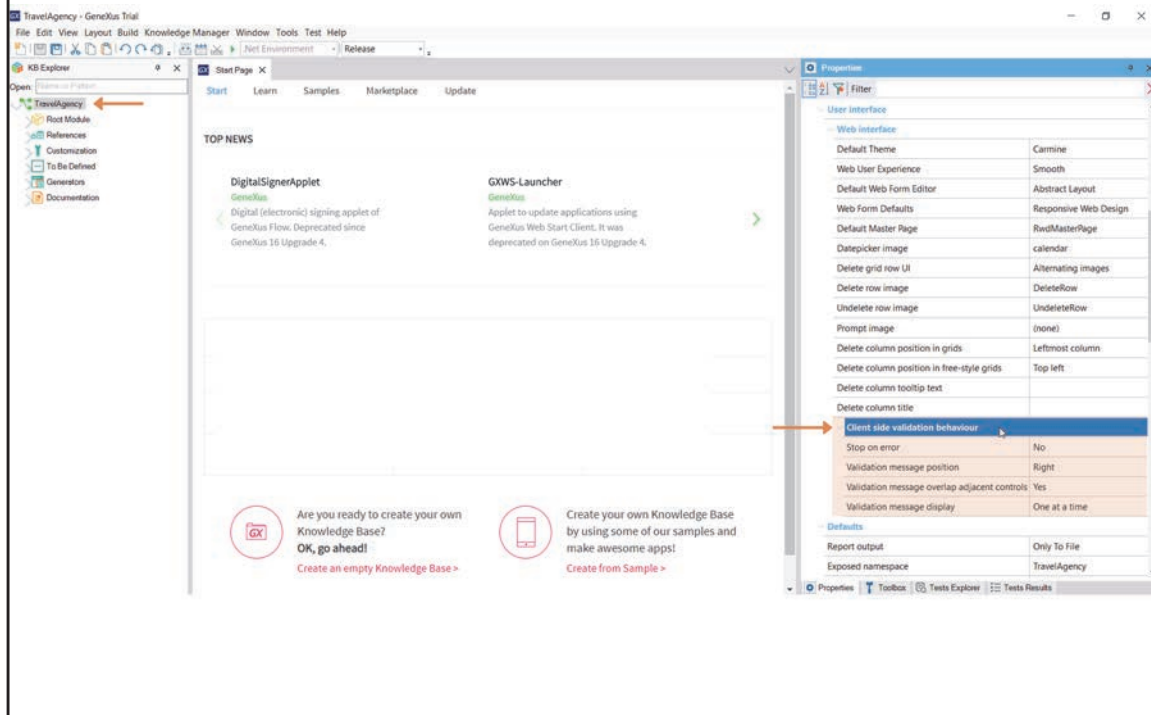
さらに、ルールを実行するように GeneXus で選択されたタイミングが適切でない場合は、トリガーイベントを使用して、実行の正確なタイミングを指定することができます。

ルールの実行



学習したこれらのルールはすべて、Web クライアントおよびサーバーの両方で検証されます。

クライアントで行われる検証は**クライアント側検証**と呼ばれ、その目的は、優れたユーザーエクスペリエンスを提供し、アプリケーションが常にインタラクティブな状態であるとユーザーに感じさせることです。しかし、送信されたすべての情報が一貫しており、システムのセキュリティを侵害せず、データベース上で操作できる唯一のものであることを実際に検証するのはサーバーです。



ここまで、ルールの既定の動作を確認してきました。ここで、バージョンレベルで使用可能な、**[Client side validation behaviour]** というプロパティのグループについて確認する必要があります。このプロパティを使用して、ルールの動作およびメッセージをさまざまな方法でカスタマイズし、アプリケーションをエンドユーザーにとってより魅力的なものにすることができます。

[Stop on error] プロパティ

Client side validation behaviour	
Stop on error	Yes
Validation message position	Right
Validation message overlap adjacent controls	Yes
Validation message display	One at a time

Error ルールを使用した際には、トランザクションでは保存が許可されなくても、エラーメッセージを表示したままで次のフィールドに移動できることを確認しました。エラーがトリガーされたときに、コントロールにフォーカスを置いたままでユーザーに値の修正を要求し、それから次のフィールドに移動するには、**[Stop on error]** プロパティの値を **[Yes]** に変更する必要があります。

[Validation message position] プロパティ

Client side validation behaviour	
Stop on error	No
Validation message position	Bottom
Validation message overlap adjacent controls	Yes
Validation message display	One at a time

また、[Validation message position] プロパティを使用して、フィールドに対するメッセージの位置を定義することもできます。このプロパティの値は、**Right** (既定値)、**Left**、**Top**、および **Bottom** です。

[Validation message overlap adjacent controls] プロパティ

The screenshot shows a 'Customer' form with fields for Id, Name, Last Name, Address, Phone, Email, and Added Date. A validation message 'Enter the customer name' is displayed over the 'Name' field. A settings dialog is open, showing the 'Validation message overlap adjacent controls' property set to 'Yes'.

Client side validation behaviour	
Stop on error	No
Validation message position	Bottom
Validation message overlap adjacent controls	Yes
Validation message display	Yes

メッセージが別のコントロールと重なる場合は、**[Validation message overlap adjacent controls]** プロパティも設定する必要があります。このプロパティは、メッセージを重ねるかどうかを指定します。

[Validation message display] プロパティ

The screenshot shows a 'Customer' form with fields for Id, Name, Last Name, Address, Phone, Email, and Added Date. A validation message 'The phone is empty' is displayed next to the Phone field. A configuration dialog titled 'Client side validation behaviour' is open, showing settings for validation messages. The 'Validation message display' property is set to 'One at a time'.

Client side validation behaviour	
Stop on error	No
Validation message position	Right
Validation message overlap adjacent controls	Yes
Validation message display	One at a time

グループの最後のプロパティは [Validation message display] です。既定値は **One at a time** で、アプリケーションによって実行された最後の検証メッセージを常に表示します。

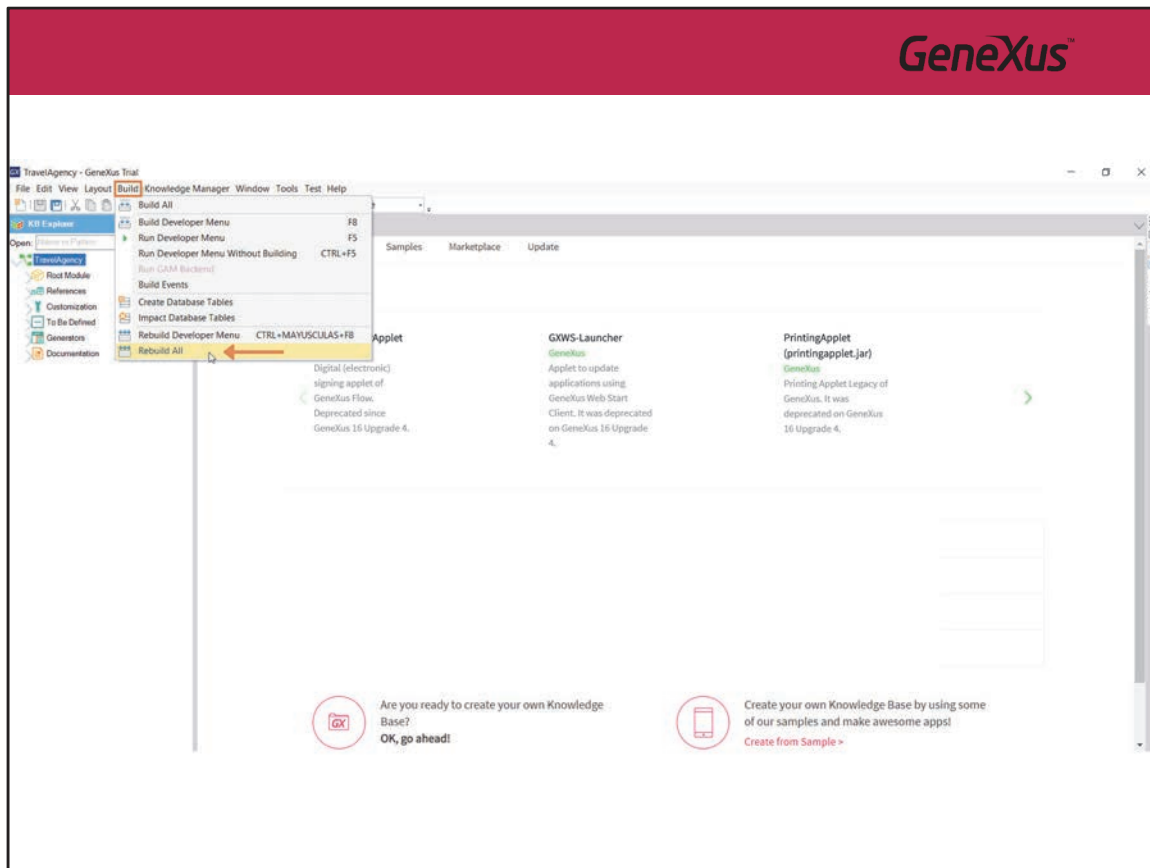
[Validation message display] プロパティ

The screenshot shows a 'Customer' form with fields for Id, Name, Last Name, Address, Phone, Email, and Added Date. The 'Name' field has a red border and a yellow tooltip that says 'Enter the customer name'. The 'Phone' field has a yellow tooltip that says 'The phone is empty'. A configuration dialog titled 'Client side validation behaviour' is open, showing the following settings:

Client side validation behaviour	
Stop on error	No
Validation message position	Right
Validation message overlap adjacent controls	Yes
Validation message display	All at once

An orange arrow points from the 'Validation message display' row to the 'All at once' value.

値が **All at once** の場合は、すべての検証メッセージを同時に画面に表示します。



これらのプロパティはフォームでのやり取りに関連する動作を変えるため、変更する場合は、フォームを持つオブジェクトをすべてもう一度生成する必要があります。そのためには、すべてのオブジェクトを生成する **[すべてリビルド]** オプションを使用する必要があります。

続いて、既に確認したルール以外のいくつかの興味深いルールを紹介します。実行順序を決めるためにそれらがどのように評価されるかや、イベントのトリガーの詳細については別の章で説明します。