

6.5 Lab Solution: Field-level validation in the UI

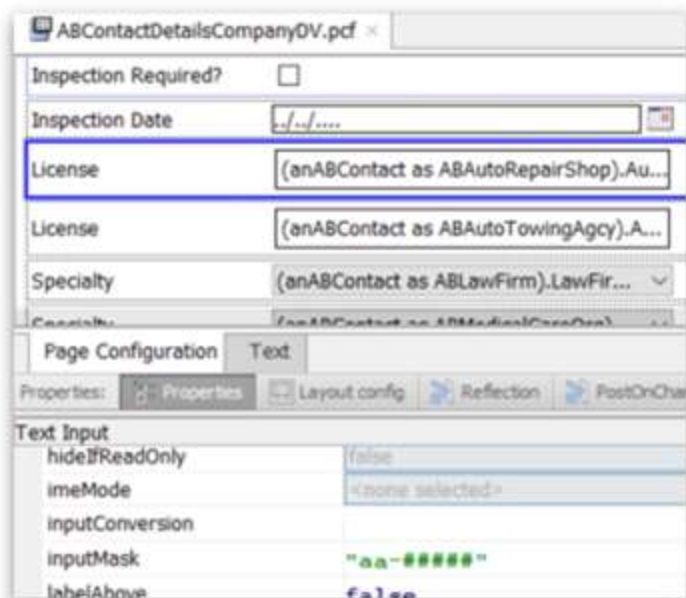


Solution

Exact details on how to complete the lab.

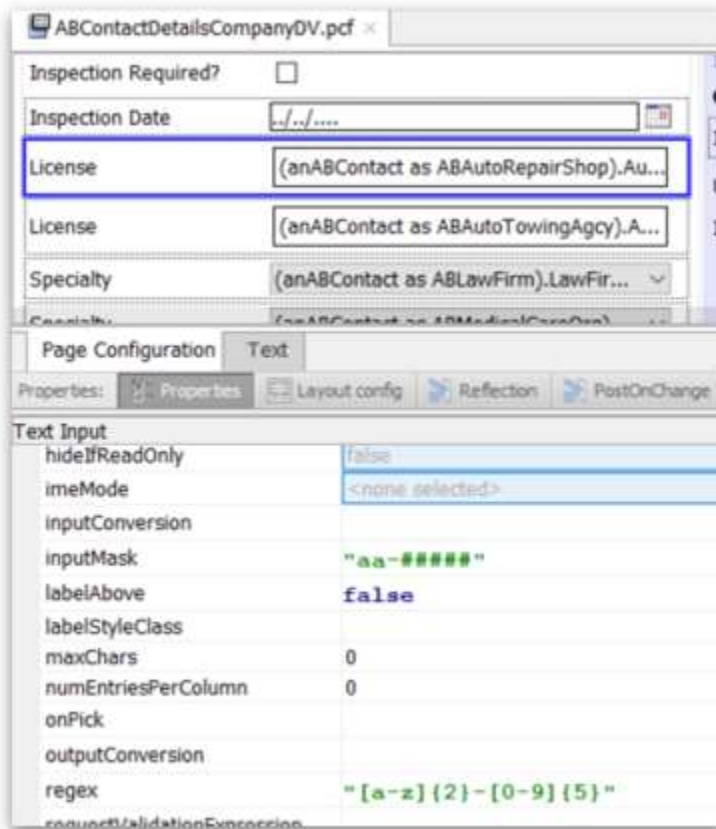
1. Configure the inputMask property of the widget.

- Type "a#-#####" in the inputMask field.



2. Configure the pattern

- Configure the regex property of the widget.



"[a-z]{2}-[0-9]{5}"

3. Configure the validationExpression property of the widget.



```
(anABContact as ABAutoRepairShop).AutoRepairLicense.contains("777") ?  
DisplayKey.get("Ext.License777Invalid"):null
```

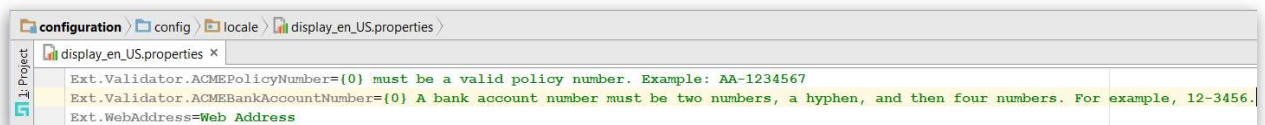
6.6 Lab Solution: Field-level validation in the Data Model



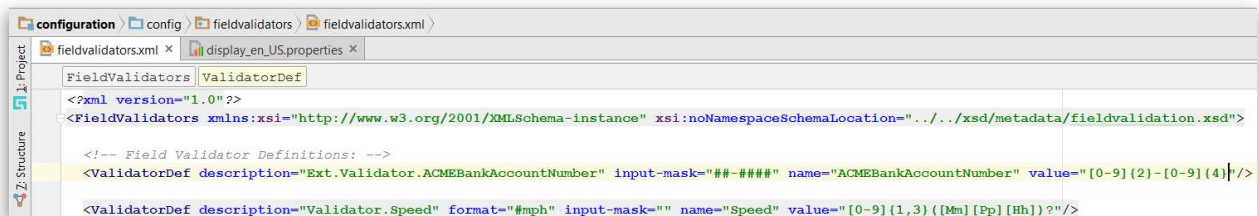
Solution

Exact details on how to complete the lab.

1. Create the display key

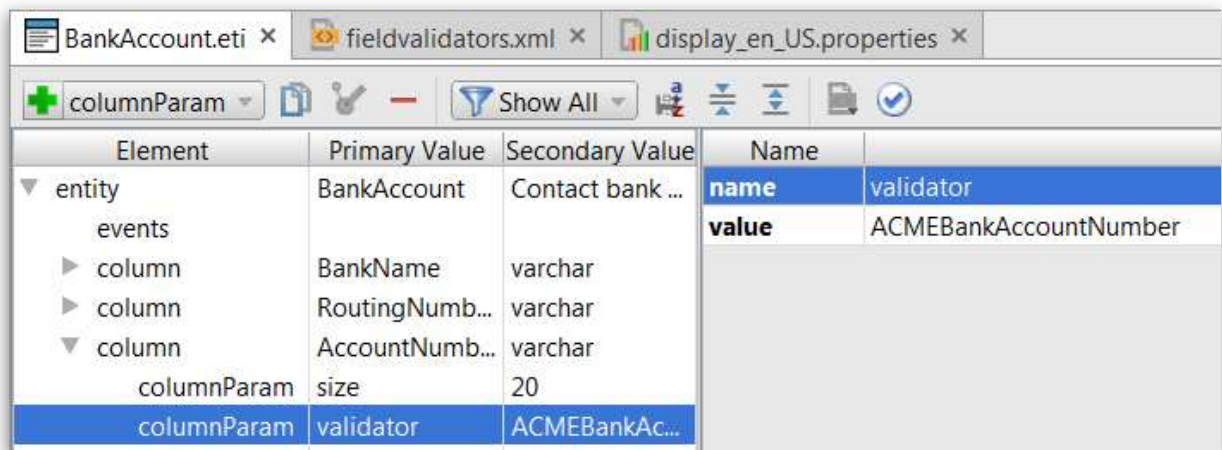


2. Create a field validator



```
<?xml version="1.0"?>
<FieldValidators xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../xsd/metadata/fieldvalidation.xsd">
  <!-- Field Validator Definitions: -->
  <ValidatorDef description="Ext.Validator.ACMEBankAccountNumber" input-mask="###-####" name="ACMEBankAccountNumber" value="[0-9]{2}-[0-9]{4}|"/>
  <ValidatorDef description="Validator.Speed" format="#mph" input-mask="" name="Speed" value="[0-9]{1,3}([Mm][Pp][Hh])?"/>
</FieldValidators>
```

3. Associate the field validator with the AccountNumber field of the BankAccount entity



Element	Primary Value	Secondary Value	Name	
entity	BankAccount	Contact bank ...	name	validator
events			value	ACMEBankAccountNumber
column	BankName	varchar		
column	RoutingNumb...	varchar		
column	AccountNumb...	varchar		
columnParam	size	20		
columnParam	validator	ACMEBankAc...		

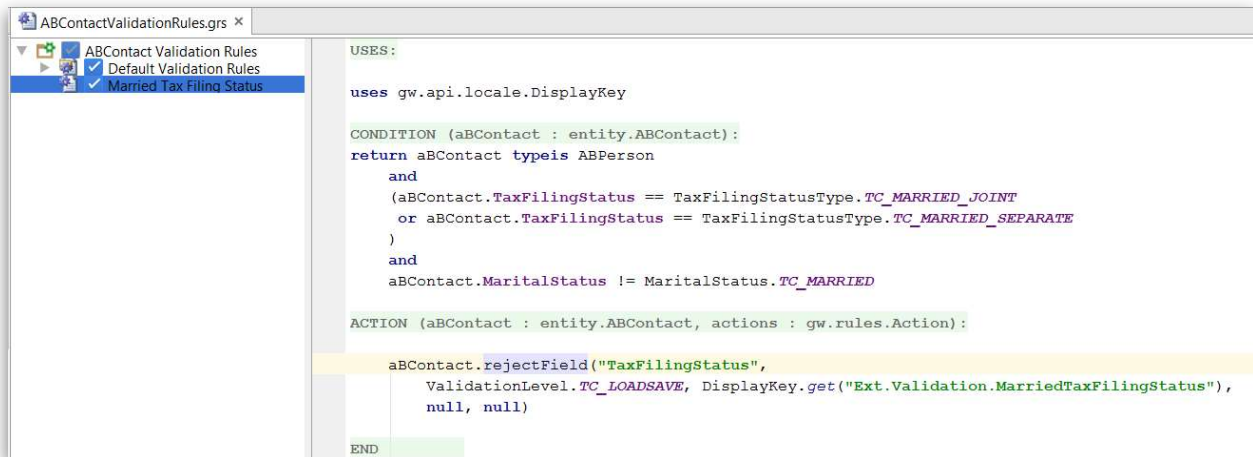
6.7 Lab Solution: Validation rules



Solution

Exact details on how to complete the lab.

1. Create a new Rule in ABContactValidationRules.



- a) You also have the option to copy and paste it from below. However, it is recommended that you type the code based on the screenshot. **Important:** please make sure that you copy and paste the sections individually without the parts highlighted in a darker gray.

```
USES:
uses gw.api.locale.DisplayKey

CONDITION (aBContact : entity.ABContact):
return aBContact typeis ABPerson
    and
    (aBContact.TaxFilingStatus == TaxFilingStatusType.TC_MARRIED_JOINT
    or aBContact.TaxFilingStatus == TaxFilingStatusType.TC_MARRIED_SEPARATE
    )
    and
    aBContact.MaritalStatus != MaritalStatus.TC_MARRIED

ACTION (aBContact : entity.ABContact, actions : gw.rules.Action):
    aBContact.rejectField("TaxFilingStatus",
        ValidationLevel.TC_LOADSAVE,
        DisplayKey.get("Ext.Validation.MarriedTaxFilingStatus"),
        null, null)

END
```

2. Add the display key

- a) Open the display.properties file
- b) Add a new entry as listed below:

```
Ext.Validation.MarriedTaxFilingStatus= Married filing jointly/Married filing separately is only  
allowed when contact is married.
```