camos.

camos Develop
Developer training

Wasele attributes

Prerequisites

- **Knowledge base „Carconfigurator“ at the end of the 3. day of the modeler training**

- **Contents:**
  - Highlight mandatory fields in contact form in color
  - Check mandatory fields for missing entries

Training targets

- **After these exercises you should …**
  - Know usage possibilities for Wasele attributes
  - Define and evaluate Wasele attributes
  - Access Wasele attributes via functions

Wasele attributes

- **Wasele attributes**
  - are used to determine features, components and methods as well as ORM-elements more precisely
  - are defined in the knowledge base properties
  - Data type: Numeric, String, RTF, Date, Currency, Graphic, or HTML
  - are defined with the creating for certain Wasele (feature, component, method and/or ORM)

  - The value of the attribute can be preset and overloaded on concrete Wasele, if necessary

Exercise: Indicate mandatory fields in color

- **Target**
  - The user can send a consultation for the current configuration via email to a car dealer of his choice

- **Problem**
  - Some specifications are optional while other specifications have to be filled in necessarily

- **Solution**
  - This differentiation is controlled by Wasele attributes on the cause variables of the form

Exercise: Indicate mandatory fields in color

- **Create the object class „ContactForm“ under the root**
  - Here you create the following string features
    - Customer_Name
    - Customer_Address
    - Customer_Email
    - Dealer_Name
    - Dealer_Email
    - Remark
  - Maintain the namings of the features

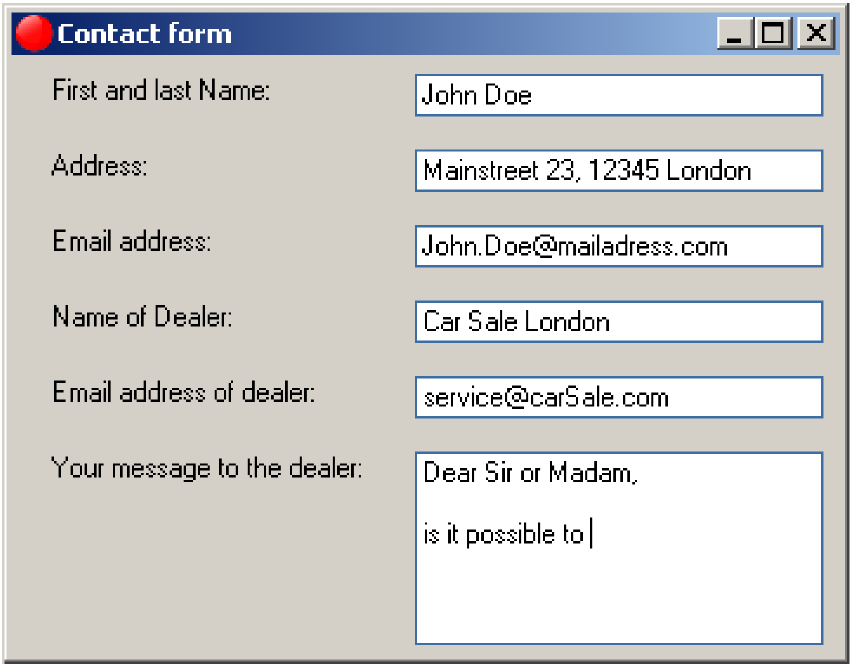
Exercise: Indicate mandatory fields in color

- **Create form „Form“ in class ContactForm**
  - Header: Contact form
  - Editlines or string editor for entry
  - Attention: The names of the editlines have to be identical with the feature names of their cause variables!
  - Naming labels with postfix „:“

- **Create the numerical feature „formhandle“**

- **Create the method New**

```
formhandle := WinOpen('Form');
```



First and last Name:	John Doe
Address:	Mainstreet 23, 12345 London
Email address:	John.Doe@mailadress.com
Name of Dealer:	Car Sale London
Email address of dealer:	service@carSale.com
Your message to the dealer:	Dear Sir or Madam, is it possible to

Exercise: Indicate mandatory fields in color

- **Create a new Wasele attribute (in KNB properties)**
  - Name: „MandatoryField", numeric
  - only for features, default value 0

- **Overload the attribute with the value 1 on the compulsory features:**
  - Remark, Dealer_Email, Customer_Email, Customer_Name

- **All mandatory fields have to be highlighted in light yellow**
  - Step 1: Determine list of all mandatory features
  - Step 2: Color mandatory form elements on form

Exercise: Indicate mandatory fields in color

- **Step 1**
  - Create the method GetMandatoryValues()
    - Disable the side effects
    - The return value is a string list

```
# Determine all mandatory features of the active object
RETURN FeCoGet('FDL', {'Attribute', 'MandatoryField', 1});
```

Exercise: Indicate mandatory fields in color

- **Step 2**
  - Create the method HighlightMandatoryValues()
    - Local variables i, iMax (numerical), Fes[] (string list)

```
FEs[] := GetMandatoryValues();
iMax := MaxIndex(FEs[]);
FOR i := 1 TO iMax DO
  WinSetBgColor(formhandle, FEs[i], {255, 255, 213});
ENDFOR;
```

  - Call HighlightMandatoryValues() in new()

```
formhandle := WinOpen('Form');
HighlightMandatoryValues();
WinStartModal(formhandle);
```

camos.

Exercise: Indicate mandatory fields in color

- **Make contact form callable**
  - Create a component of the class „ContactForm‟ in „start‟

  - Extend the menu „Administration‟ in class „start‟
    - New menu trigger „Contact car dealer‟

```
_ContactForm := 'ContactForm';
_ContactForm := NOVALUE;
```

    - Enabled: _Car <> NOVALUE
    - Assign IconEmail.bmp as icon of the menu item
      - Enable „Icon in toolbar‟

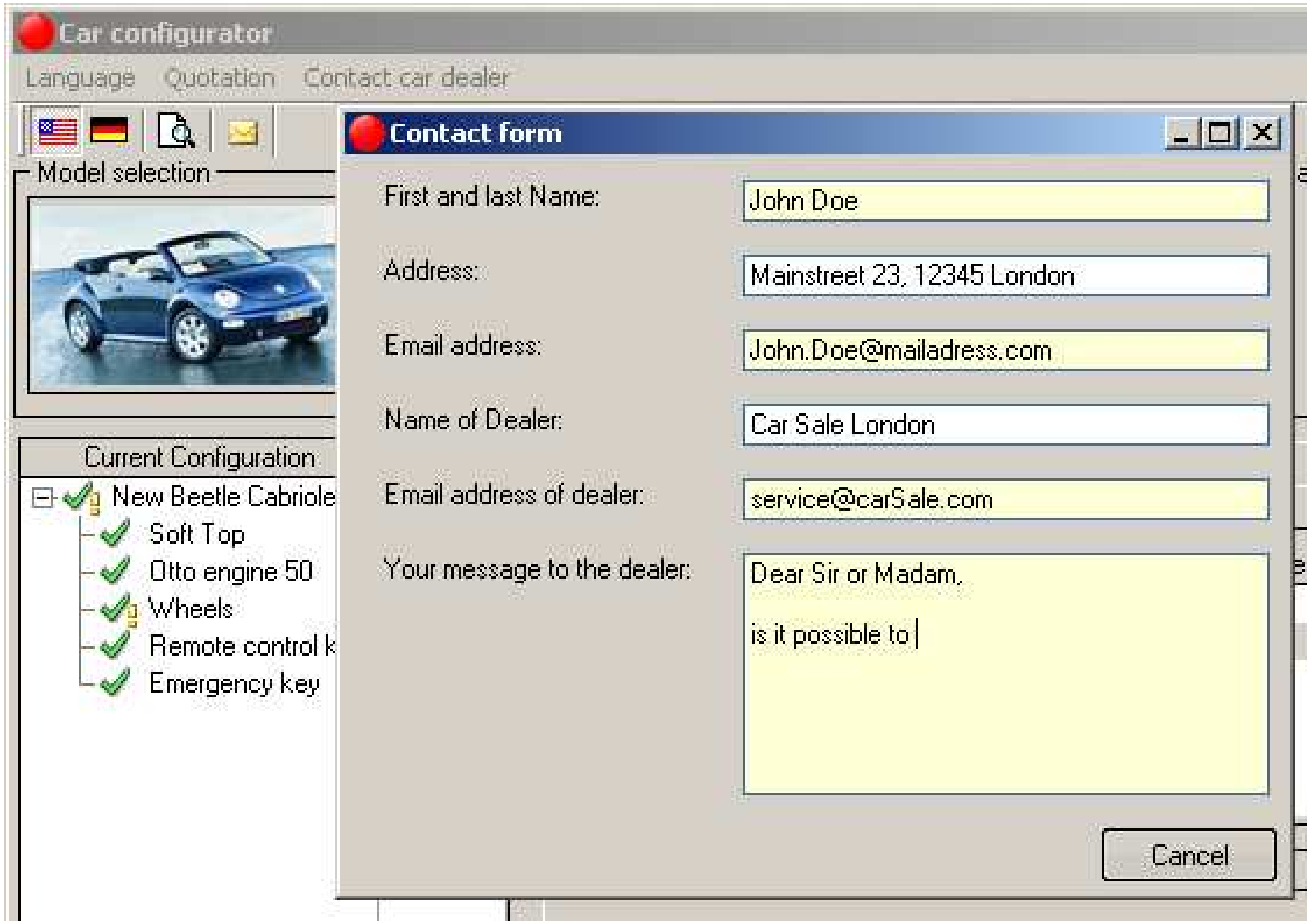
Exercise: Indicate mandatory fields in color

- **Close form**
  - Create a new pushbutton on the form „Form“
  - Name: „Cancel“
  - The selection trigger closes the form

```
WinClose(formhandle);
```

- **Start the CarConfigurator and test the appearance of the contact form**

Result: Indicate mandatory fields in color



First and last Name:	John Doe
Address:	Mainstreet 23, 12345 London
Email address:	John.Doe@mailadress.com
Name of Dealer:	Car Sale London
Email address of dealer:	service@carSale.com
Your message to the dealer:	Dear Sir or Madam, is it possible to

Exercise: Check mandatory fields

- **The entries of the mandatory fields have to be checked before sending the email**
  - Create the method CheckMandatoryValues() in the class „ContactForm“
    - Disable side effects
    - Local variables ok, i and iMax (numerical), Features[] (string list)

```
# Determine mandatory fields or mandatory features
Features[] := GetMandatoryValues();
ok := 1;
iMax := MaxIndex(Features[]);
FOR i := 1 TO iMax DO
 # If the feature has the value NOVALUE...
  IF <<Features[i]>> = NOVALUE THEN
 # ... set ok to zero
    ok := 0;
  ENDIF;
ENDFOR;
RETURN ok;
```

Exercise: Check mandatory fields

- **Email should contain BLOB of the configuration as attachment**
  - Create a predecessor component on „start“ in class „ContactForm“

- **Create the method SendEmail() in ContactForm**
  - Local variables Body (string), OK (numeric)

```
Body := StrConcat(Customer_Name, ' (', Customer_Email, ')^n',
  Customer_Address, '^n^nwrites the following message to
  you:', '^n^n', Remark);
OK := SendMail(Dealer_Email, 'Inquiry with regard to
  Carconfigurator', Body, {'Configuration.sot',
  OT2Bin(@start._Car)});
IF OK THEN
  RETURN 1;
ELSE
  RETURN 0;
ENDIF;
```

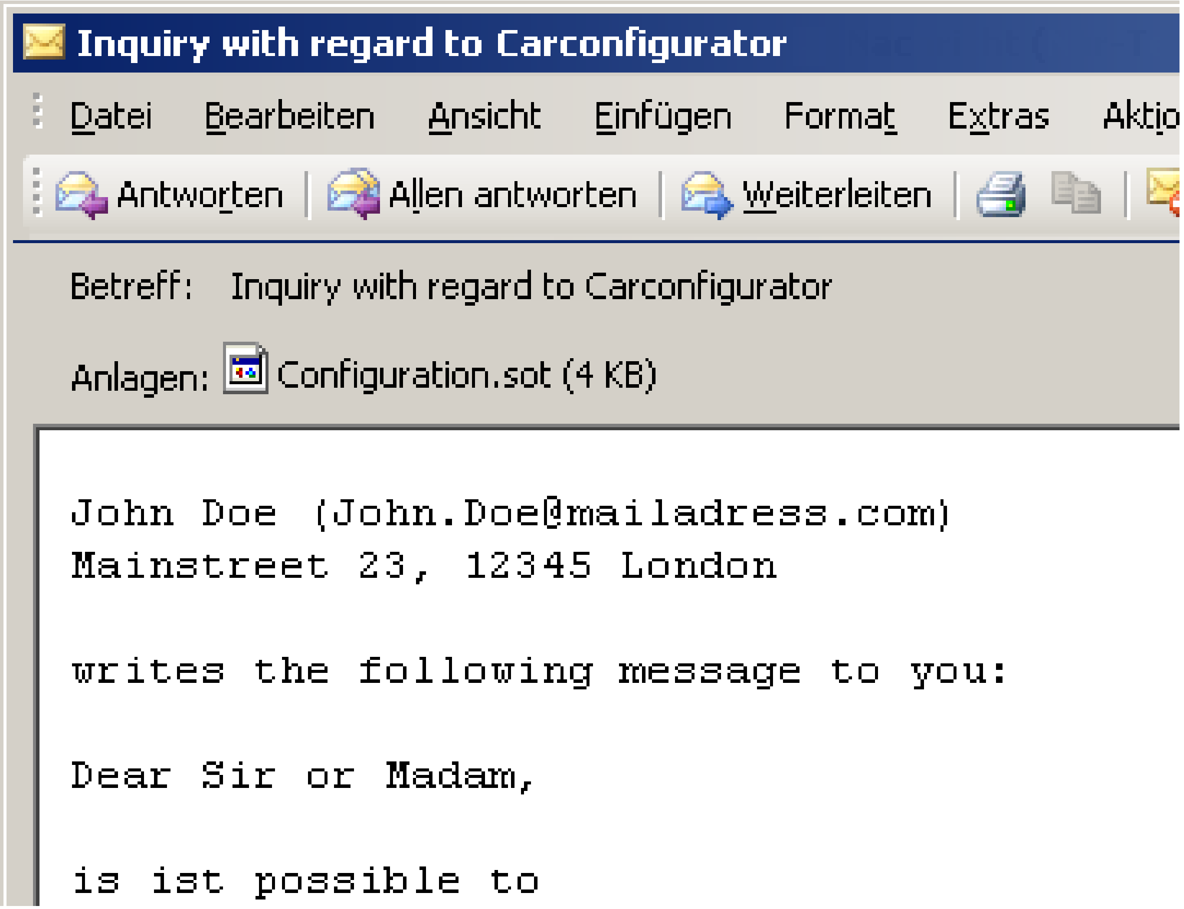
Exercise: Check mandatory fields

- **Create a new pushbutton on the form „Form“**
  - Name: „Send“
  - Selection trigger sends the email

```
IF CheckMandatoryValues() THEN
   IF SendEmail() THEN
     WinMessage('INFO', 'Inquiry sent');
     WinClose(formhandle);
   ELSE
     WinMessage('ERROR', GetLastError());
   ENDIF;
ELSE
   WinMessage('ERROR', 'Fill mandatory fields!');
ENDIF;
```

Result: Check mandatory fields

- Test if the email is only sent with filling all mandatory fields



Betreff: Inquiry with regard to Carconfigurator

Anlagen: Configuration.sot (4 KB)

```
John Doe (John.Doe@mailadress.com)
Mainstreet 23, 12345 London

writes the following message to you:

Dear Sir or Madam,

is ist possible to
```