

SmartComponent Library User-Interface Translation Developer Guide



Consultingwerk Ltd.

Agenda

- **User Interface Translation via Resource Files**
- Translation of messages
- Validation with translatable messages
- Translation of common strings

.NET Resource Files







- .resx files standard way for UI translation in the .NET framework
- .resx files do always contain images (base64 encoded binary) and the location of components on the yellow frame below the design canvas (SmartToolBarController, SmartBusinessEntityAdapter, ...)
- One .resx file created per
 - Form
 - UserControl (e.g. SmartViewerControl)

UI translation with Resource Files

- .resx files may optionally contain all translatable strings used on the design (field labels, tooltip text, ...)
- Which strings are translatable and which ones not (e.g. Tool Keys in the Ribbon) decided by Control vendor, e.g. Infragistics
 - They know better than we do which properties are supposed to be localized because they may be visible to the end user!

One .resx file per language + default

- DatabaseLoginForm.cls is the Form source code
- DatabaseLoginForm.resx is the default .resx file
-nl.resx is the .resx file with the Dutch translation

 DatabaseLoginForm.cls	22.08.2013 06:56	CLS-Datei	37 KB
 DatabaseLoginForm.de.resx	26.05.2013 00:39	.NET Managed Re...	14 KB
 DatabaseLoginForm.fr.resx	26.05.2013 00:39	.NET Managed Re...	15 KB
 DatabaseLoginForm.nl.resx	22.08.2013 06:56	.NET Managed Re...	15 KB
 DatabaseLoginForm.r	22.08.2013 06:56	Progress .R File	151 KB
 DatabaseLoginForm.resx	22.08.2013 06:56	.NET Managed Re...	46 KB

```
<data name="label1.Text" xml:space="preserve">
  <value>Bitte geben Sie Benutzer ID, Kennwort und Domain an</value>
</data>
<data name="label2.Text" xml:space="preserve">
  <value>Datenbank:</value>
</data>
<data name="label3.Text" xml:space="preserve">
  <value>Benutzer ID:</value>
</data>
<data name="label4.Text" xml:space="preserve">
  <value>Kennwort:</value>
</data>
```



```
<data name="label1.Text" xml:space="preserve">
  <value>Geef gebruikersnaam en paswoord in</value>
</data>
<data name="label2.Text" xml:space="preserve">
  <value>Databank:</value>
</data>
<data name="label3.Text" xml:space="preserve">
  <value>Gebruiker:</value>
</data>
<data name="label4.Text" xml:space="preserve">
  <value>Paswoord:</value>
</data>
```



One .resx file per language + default

■ Progress ResourceManager

```
-----  
Purpose: Initializes the Visual Design  
Notes:  
-----*/  
METHOD PRIVATE VOID InitializeComponent ():  
  
/* NOTE: The following method is automatically generated.  
  
We strongly suggest that the contents of this method only be modified using the  
Visual Designer to avoid any incompatible modifications.  
  
Modifying the contents of this method using a code editor will invalidate any support for this file. */  
@VisualDesigner.FormMember (NeedsInitialize="true":U).  
DEFINE VARIABLE resources AS Progress.Util.ResourceManager NO-UNDO.  
resources = NEW Progress.Util.ResourceManager("Consultingwerk.Windows.OpenEdgeLogin.DatabaseLoginForm":U).  
@VisualDesigner.FormMember (NeedsInitialize="true":U).
```

- Picks up the appropriate .resx file for the current user, based on user locale (language settings)
- Search order: **de-DE**, **de**, **default** (optional country specific language support)

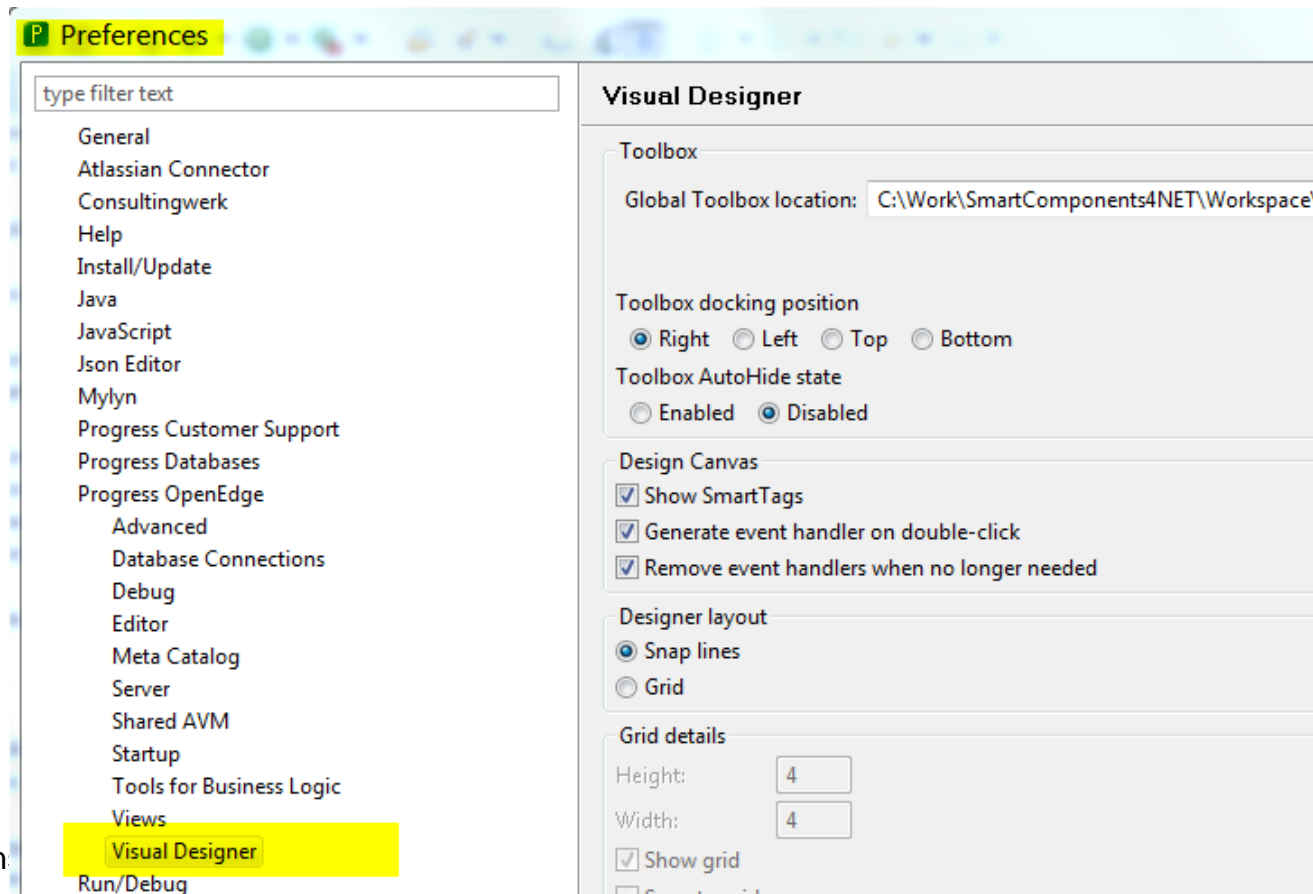
Developer requirements

- It's a good habit to let the Visual Designer put a :U on **all** strings inside InitializeComponent method
- Progress does not distinguish translatable and untranslatable properties here (they can't)

```
/* */  
/* ultraGroupBox1 */  
/* */  
THIS-OBJECT:ultraGroupBox1:AccessibleDescription = resources:GetString("ultraGroupBox1.AccessibleDescription":U)  
THIS-OBJECT:ultraGroupBox1:AccessibleName = resources:GetString("ultraGroupBox1.AccessibleName":U).  
THIS-OBJECT:ultraGroupBox1:Anchor = CAST(resources:GetObject("ultraGroupBox1.Anchor":U), System.Windows.Forms.Ar  
appearance1:BackColor = System.Drawing.SystemColors:ButtonFace.  
appearance1:TextHAlignAsString = resources:GetString("appearance1.TextHAlignAsString":U).  
appearance1:TextVAlignAsString = resources:GetString("appearance1.TextVAlignAsString":U).  
THIS-OBJECT:ultraGroupBox1:Appearance = appearance1.  
THIS-OBJECT:ultraGroupBox1:BackgroundImage = CAST(resources:GetObject("ultraGroupBox1.BackgroundImage":U), System  
THIS-OBJECT:ultraGroupBox1:BackgroundImageLayout = CAST(resources:GetObject("ultraGroupBox1.BackgroundImageLayout":U), System
```


Developer requirements



- It's a good habit to let the Visual Designer put a :U on all strings in the



Developer requirements

- By default field labels, button captions, tool tips, etc. will be stored in the source code, considered untranslatable
- Those strings **can be written** to the resource file to make them translatable
- Therefore the ***Localizable*** property of the Form/UserControl/SmartViewerControl needs to be set to True
- Leave the ***Language*** on “***Default***”

DatabaseLoginForm : Progress.Windows.Form

Properties	Events
BackgroundImageLayout	Tile
CancelButton	buCancel
CausesValidation	True
ContextMenuStrip	(Keine)
ControlBox	True
Cursor	Default
DisposeDialogOnClose	True
DoubleBuffered	False
Enabled	True
▶ Font	Microsoft Sans Serif; 8,25pt
ForeColor	 ControlText
FormBorderStyle	Sizable
HelpButton	False
▶ Icon	 (Symbol)
ImeMode	NoControl
IsMdiContainer	False
KeyPreview	False
Language	(Standard)
Localizable	True
▶ Location	0; 0
Locked	False
MainMenuStrip	(Keine)
MaximizeBox	False

Demo

- Resource File translation in the SmartComponent Library (SmartDB)

UI Translation Process at HAVI

- Developer creates “Localizable” Forms and UserControls
- Authorized translation user performs Translation using Translation maintenance (SmartDB)
- When application user signs in, translations will be fetched from the Backend and the client will write the appropriate .resx files to the local file system

Agenda

- User Interface Translation via Resource Files
- Translation of messages
- Validation with translatable messages
- Translation of common strings

Translation of messages

- Translations are stored in the SmartDB
- Translations may have a default language
- Translations may be stored in a language (SmartDB language table)

Message Maintenance

Message Maintenance

File Start SmartFramework

Navigation: Erster Datensatz, Vorheriger Datensatz, Nächster Datensatz, Letzter Datensatz

Maintenance: Hinzufügen, Ändern, Speichern, Abbrechen, Kopieren, Löschen

Legen Sie eine Spalte hier ab, um nach dieser zu gruppieren.

Typ	Group	Number	Text	Language
?	SFR	1	Are you sure tha...	
?	SFR	1	Sind Sie sicher,...	Deutsch (
?	SFR	2	Current values h...	
✗	VALMSG	1	&1 should not b...	
✗	VALMSG	1	&1 muss gefüllt...	Deutsch (
✗	VALMSG	2	The value "&2" f...	

Message Group: VALMSG Number: 1

Language: <default> Type: Error

Text: &1 should not be empty.

Detail: The field &1 needs to be filled.

Message properties

- Message Group (for logical ordering)
- Message Number (free to define yourself)
- Type: Info, Warning, Error, Question
- Short **Text**
- Message **Detail**
- Message Text and Detail may be using &1...&9 as placeholders for custom values

API for showing a message to the user

■ 1) Retrieve a Message

```
DEFINE VARIABLE oMessageProvider AS Consultingwerk.SmartFramework.IMessageProvider NO-UNDO .
DEFINE VARIABLE oMessage          AS Consultingwerk.SmartFramework.Message          NO-UNDO .

/* Get reference to the Message Provider Service */
oMessageProvider = {Consultingwerk/get-service.i Consultingwerk.SmartFramework.IMessageProvider} .

/* Get the message structure with 0 to 9 message parameters substituted */
oMessage = oMessageProvider:GetMessage ("SFR":U,
                                        1,
                                        "Hello World") .
```

■ 2) Present the Message

```
Consultingwerk.Util.MessageFormHelper:ShowMessage (oMessage) .
```

API for showing a message to the user

- 3) Question message returns Yes/No

```
USING Consultingwerk.Util.*          FROM PROPATH .
USING Consultingwerk.Framework.Enum.* FROM PROPATH .

IF MessageFormHelper:ShowMessage (oMessage) = DialogResultEnum:DialogResultYes THEN
DO:
    /* yes branch */
END.
ELSE DO:
    /* no branch */
END.
```

Agenda

- User Interface Translation via Resource Files
- Translation of messages
- Validation with translatable messages
- Translation of common strings

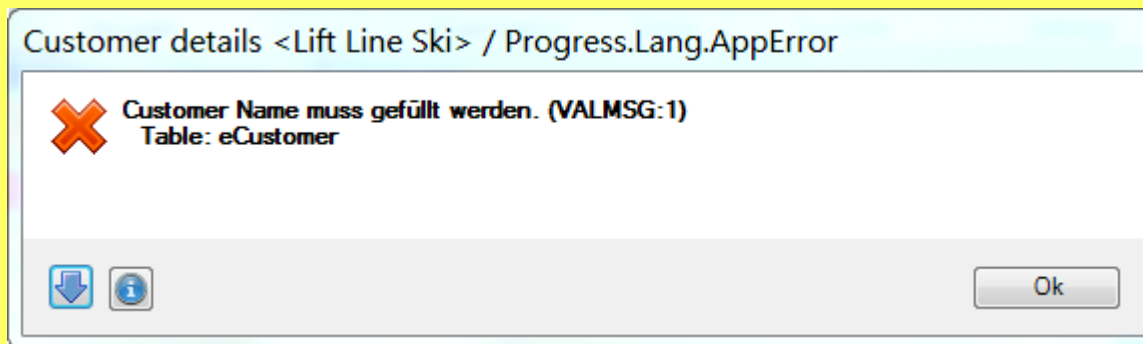
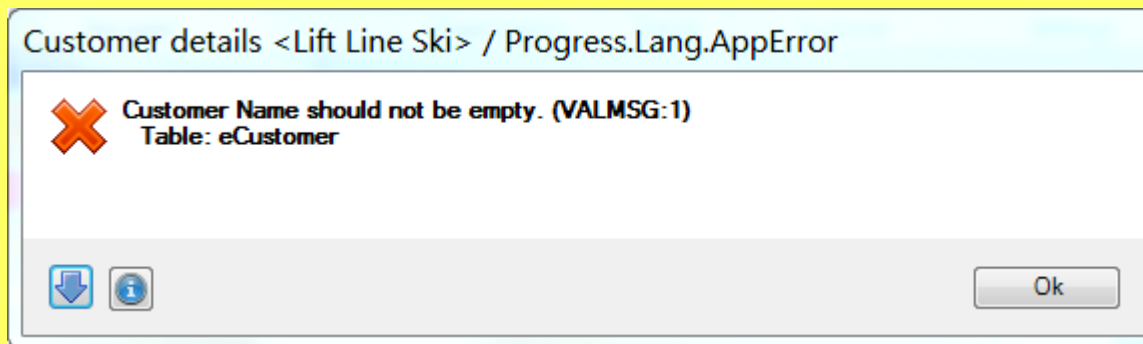
Validation with translatable messages

- The usage of the ERROR-STRING attribute of the before table buffer is used for validation messages (Business Entity or Data Access object)
- Does not support “Message” structure
- Consultingwerk.SmartFramework.MessageFormatter returns message from Message Provider for this purpose

Validation with translatable messages

```
MessageFormatter: GetMessage ("VALMSG":U, 1, "Customer Name").
```

- Returns message VALMSG 1 with Customer Name as a parameter



Sample using „Validate“ methods

```

/*-----
Purpose: Provides a hook for high level data validation before Update
operations
Notes:   Invoked during SaveChanges (). When the ERROR flag of the ProDataset
is set, the Update operation will be cancelled before writing back
the data to the database using the DataAccess object
-----*/
METHOD OVERRIDE PUBLIC VOID ValidateData ():

FOR EACH eCustomer ON ERROR UNDO, THROW:

    Validate:IsNotNullOrEmpty (BUFFER eCustomer:HANDLE, "Name":U,
        MessageFormatter:GetMessage ("VALMSG":U, 1, "Customer Name")).

    IF
    Validate:IsNotNullOrEmpty (BUFFER eCustomer:HANDLE, "SalesRep":U,
        MessageFormatter:GetMessage ("VALMSG":U, 1, "Sales Rep"))

    THEN
    Validate:CanFind (BUFFER eCustomer:HANDLE,
        "SalesRep":U,
        "Consultingwerk.SmartComponentsDemo.OERA.Sports2000.SalesRepBusinessEntity":U,
        "eSalesRep":U,
        SUBSTITUTE ("FOR EACH eSalesRep WHERE eSalesRep.SalesRep = &1":U, QUOTER (eCustomer.Sales
        MessageFormatter:GetMessage ("VALMSG":U, 2, "Sales Rep", eCustomer.Salesrep)) .

    END.

END METHOD.

```

Agenda

- User Interface Translation via Resource Files
- Translation of messages
- Validation with translatable messages
- Translation of common strings

Translation of common strings

- Common strings (e.g. value in calculated fields etc.) are either translated
 - Using the original word used by the developer
 - Using a key
 - Optionally using a scope, e.g. application module, similar to the message group

Translation maintenance in the SmartDB

Translation Maintenance

File Start SmartFramework

Navigation: First record, Previous record, Next record, Last record

Maintenance: Add record, Update record, Save changes, Cancel update, Copy record, Delete record

Drag a column header here to group by that column.

Translation Scop	Translation Key	Language
SmartToolBarC...	TableIOToolTableIO...	Deutsch (DE)
SmartToolBarC...	TableIOToolTableIO...	Deutsch (DE)
SmartToolBarC...	TableIOToolTableIO...	Deutsch (DE)
		Deutsch (DE)
		Deutsch (DE)
		Deutsch (DE)
		Deutsch (DE)
		Deutsch (DE)
SmartToolBarC...	NavigationToolNavig...	Deutsch (DE)
SmartToolBarC...	TableIOToolTableIO...	Deutsch (DE)
SmartToolBarC...	NavigationToolNavig...	Deutsch (DE)
SmartUpdatable...	SaveRecordPromt_...	
Attachment		Deutsch (DE)
SmartToolBarC...	TableIOToolTableIO...	Deutsch (DE)

Language: Deutsch (DE)

Scope: Attachment

Key: n/a

Original Text: Attachments

Translated Text: Anlagen

API for retrieving translations

- Translations are retrieved from the Consultingwerk.SmartFramework.ITranslationProvider service
- Translations will be cached on the client to provide better performance
- Same API if used on client of AppServer

- Simplified API using **translate.i** include file

```
/*-----  
File      : translate.i  
Purpose   : Provides String Translation based on an original string  
  
Syntax    : Static String: {Consultingwerk/translate.i &scope="" &string="'Question':U"}  
           Variable:      {Consultingwerk/translate.i &scope=cScope &string=cString}  
           No scope:      {Consultingwerk/translate.i &string="'Question':U"}  
  
           By key:        {Consultingwerk/translate.i &scope="" &key="'xyz':U"}  
  
Author(s) : Mike Fechner / Consultingwerk Ltd.  
Created   : Fri Jan 11 00:33:45 CET 2013  
-----*/
```

```
ASSIGN cKey      = GUID /* ensure a group/tab for this Control only */  
       cCaption = {Consultingwerk/translate.i &scope="'Attachment':U" &string="'Attachments':U"}  
       .
```

Questions



Thank you!