

# ELOwf

# Entwicklerleitfaden

[Stand: 11.11.2016 | Programmversion: 10.01.000]

## Inhalt

1	Übersetzung / Lokalisierung.....	3
1.1	Übersetzungstabelle benutzen.....	3
1.2	properties-Dateien benutzen.....	4
1.3	Vergleich .....	6
1.4	Übersetzungen bearbeiten.....	7
1.5	Dynamische Sprachauswahl .....	7
2	Datumsfelder.....	8
2.1	Das Format.....	8
2.2	Internationalisierung & Zeitzonen .....	8
3	Numerische Felder und Betragsfelder .....	9
3.1	Der Unterschied.....	9
3.2	Verhalten.....	9
3.3	Format.....	9
3.4	Formeln.....	10
3.5	Spontane Formeln.....	10
3.6	Skripting .....	11
4	Linkfelder .....	12
5	Zeichnungen .....	13
6	Validierung .....	15
6.1	Einleitung.....	15
6.2	Grundverhalten .....	15
6.3	Eigene Validierungsnachrichten.....	16
6.4	Eigene Validierungsfunktionen (JS_VAL_....).....	17

6.5	Eigene Filterfunktionen (JS_FILTER_...)	18
6.6	Kann ich je nach Anwendungsfall die Validierung ein- oder ausschalten?	18
7	Listen	19
7.1	Übersicht	19
7.2	Dynamische Stichwortliste	19
7.3	Dynamische Stichwort-Map	24
8	Eigene Dateien hinzufügen	26
8.1	CSS-Dateien	26
9	Ereignisse	27
9.1	onInit()	27
9.2	inputChanged(elem)	27
9.3	saveClicked()	27
9.4	nextClicked(id)	28
10	Verschiedenes	29
10.1	Drucken	29
10.1.1	Textbereiche automatisch erweitern	29
10.2	Einstellbare Tastaturbefehle	30

## 1 Übersetzung / Lokalisierung

Es stehen mehrere Möglichkeiten zur Verfügung, um Workflow-Formulare zu lokalisieren bzw. internationalisieren, damit Sprachtexte und Tooltips automatisch in der Clientsprache übersetzt werden.

### 1.1 Übersetzungstabelle benutzen

Die "klassische" Vorgehensweise, um Texte zu internationalisieren, ist Sprachtexte und Tooltips in dem Volltext zu speichern. Damit werden die Texte anhand der Übersetzungstabellen, die in den Clients verfügbar sind, automatisch ersetzt.

Guten Tag

Properties of the selected cell

Field type	Input
Text	
Variable name	IX_MAP_FOOBAR
Keyword list	
URL	
Display	
Tooltip	Dies ist ein Tooltip
Validation	notempty
Validation message	Dies ist eine beliebige Validierungsnachricht

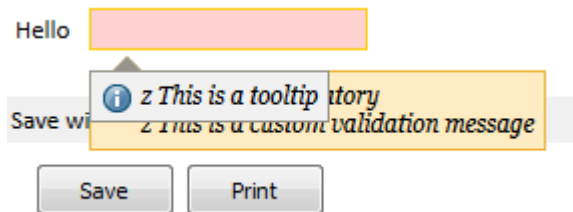
Die Übersetzungstabelle würde in etwa so aussehen:

Translation table		
System language en ▼	de	fr
z This is a tooltip	Dies ist ein Tooltip	Ceci est une infobulle
z This is a custom validation message	Dies ist eine beliebige Validierungsnachricht	Ceci est message de validation person...
z Hello	Guten Tag	Bonjour



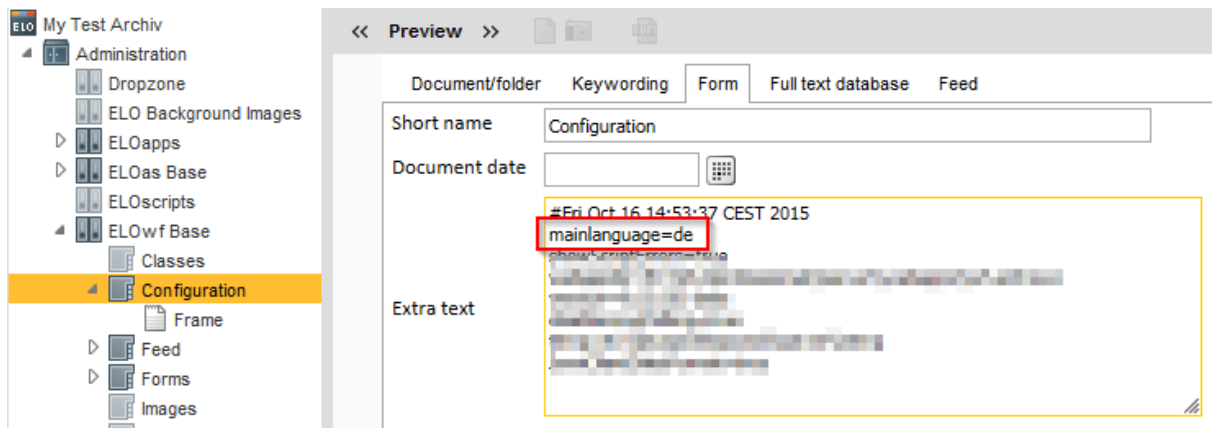
**Hinweis:** Die vorangestellte "z" in dem Screenshot oben wurde nur hinzugefügt, um die Texte für die Anzeige zu sortieren.

Das Ergebnis in einem englischsprachigen Client wäre also:



Der Screenshot ist etwas überfüllt, aber es sollte Ihnen die zugrundeliegende Idee vermitteln. Falls keine Übersetzungen für einen bestimmten Text gefunden werden, wird das Original behalten.

Wie Sie in diesem Beispiel erkennen können, ist die Referenzsprache für ELOwf nicht unbedingt die Systemsprache. Die Standardsprache wird durch "mainlanguage" in dem ELOwf Ordner *Configuration* definiert.



Diese Sprache wird bei der Suche nach Übersetzungen als "Referenz" verwendet.

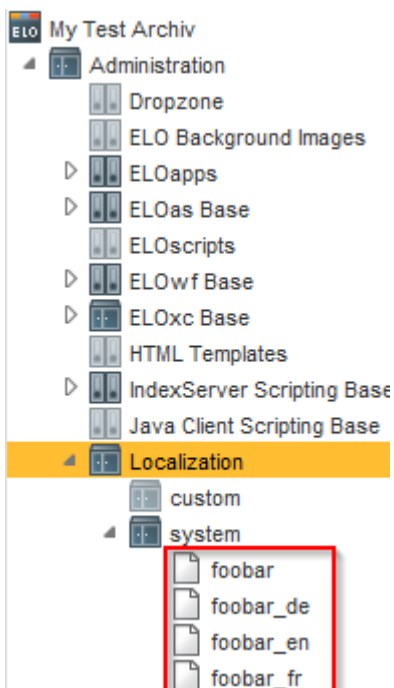
## 1.2 properties-Dateien benutzen

Die "neue" Vorgehensweise, um Texte zu lokalisieren, ist es, Übersetzungsschlüssel in dem Formular zu verwenden, und dann .property-Dateien zu benutzen, um die Übersetzungen auszuliefern. Damit können Sie auch maßgeschneidert übersetzte Hinweise per Skripting erstellen, was nicht auf andere Weise möglich wäre.

The screenshot shows a form field on the left with the text "foo.greeting" and an empty input box. To the right is a "Properties of the selected cell" panel. The properties are as follows:

Properties of the selected cell	
Field type	Input
Text	<input type="text"/>
Variable name	IX_MAP_FOOBAR
Keyword list	<input type="text"/>
URL	<input type="text"/>
Display	<input type="text"/>
Tooltip	foo.mytooltip
Validation	notempty
Validation message	foo.myvalidationmsg
Formula	<input type="text"/>
Shortcut	<input type="checkbox"/>
Character count	20 <input type="checkbox"/>
	<input type="checkbox"/> Read-only
Form columns	1

Die .property-Dateien können unter dem Ordner Administration\Localization\system abgelegt werden.



Jede dieser 4 Dateien ist eine .properties-Datei für die jeweilige Sprache. Die "foobar" ohne ein nachgestelltes Sprachkürzel wird per Default benutzt, falls eine unerwartete/unbekannte Sprache nachgefragt wird.



Hinweis: Sie können jeden beliebigen Dokumentennamen (Sprachkürzel davon ausgenommen) für die Sprachdateien benutzen. Alle Schlüssel werden zur Laufzeit in den jeweiligen Speicher für die Sprache geladen.

Der Inhalt von "foobar\_en.properties" könnte so aussehen:

```
foobar_en.properties x
1 foo.greeting=Hello
2 foo.mytooltip=This is a tooltip
3 foo.myvalidationmsg=This is a validation message
4
5 bar.mynotification=That can be called inside a script!
```

Sie können auch eine Teilmenge von diesen Properties importieren, um sie fürs Skripten zur Verfügung zu stellen. Um das zu ermöglichen, geben Sie die Präfix für den Übersetzungsschlüssel in der unten angezeigten Eigenschaft ein. In diesem Beispiel werden alle Schlüssel, die mit "bar" anfangen, für das Formular verfügbar gemacht.

Global form settings

Keywording form	11: Test	▼
Template name	translations	
Languages	de.en.fr	▼
Translation keys	bar	
<input type="checkbox"/> Limited variable access		
<input type="checkbox"/> Realign columns		

Sie sind in Skripten über "elo.locale.store" aufrufbar. In diesem Fall können Sie eine MessageBox mit einer übersetzten Nachricht einfach so erstellen:

```
eloAlert(elo.locale.store['bar.mynotification'])
```

## 1.3 Vergleich

Als Faustregel gilt: Sie können Übersetzungstabellen benutzen, wenn Sie nur ein paar Formulare in einer Sprache übersetzen müssen. Wenn Sie mehrere größere Formulare haben, oder in mehreren Sprachen übersetzen müssen, wird die auf Properties basierte Lösung empfohlen.

Die Nutzung von Properties-Dateien hat die folgenden Vorteile:

- Skripte können auch auf übersetzte Texte zugreifen, zum Beispiel auf Hinweiskästchen.

- Je nach Kontext muss manchmal ein Wort in einer Sprache in unterschiedliche Wörter einer anderen Sprache übersetzt werden. Das ist nicht mit Übersetzungstabellen möglich.
- Properties-Dateien werden von professioneller Übersetzungs-Software unterstützt.
- Properties-Dateien werden in dem ELO-Archiv versioniert.

Das mag in kleinen Archiven mit niedrigen Anforderungen für die Lokalisierung überflüssig sein, aber der Nutzen steigt zunehmend, wenn diese Anforderungen zunehmen.

## 1.4 Übersetzungen bearbeiten

Aus Performancegründen speichert ELOwf Übersetzungen in einem Cache. Nach der Bearbeitung der Übersetzungstabellen oder Properties-Dateien, laden Sie das Modul ELOwf neu, damit die Änderungen übernommen werden. Führen Sie nach der Änderung der Properties-Dateien außerdem einen Neustart des Indexserver durch.

## 1.5 Dynamische Sprachauswahl

Bei der Eingabe von Sprachen in den Formulareinstellungen für "languages" (siehe Bild) gilt:

The screenshot shows a 'Global form settings' window. It contains several fields: 'Keywording form' with a dropdown set to '11: Test', 'Template name' with the text 'translations', 'Languages' with a dropdown set to 'de.en.fr' (this field is highlighted with a red rectangle), and 'Translation keys' which is empty. Below these fields are two checkboxes: 'Limited variable access' and 'Realign columns', both of which are unchecked.

Ein Kombinationsfeld wird dem Formular hinzugefügt, mit dem man innerhalb des Formulars selbst die Sprache wechseln kann.

The screenshot shows a form with a text input field containing 'Hello'. Below it is a button labeled 'Save without passing forward'. At the bottom are two buttons: 'Save' and 'Print'. To the right of the 'Hello' field is a language selection dropdown menu (highlighted with a red rectangle) with a list of options: 'English', 'Deutsch', 'English' (highlighted in blue), and 'Français'.

## 2 Datumsfelder

Obwohl es nicht auf dem ersten Blick ersichtlich ist, kann jedes Datumsfeld zusätzliche Informationen zur Uhrzeit beinhalten. Sie können zum Beispiel "12. November 2015" im Kalender auswählen, aber Sie können auch eine Uhrzeit eingeben, wie beispielsweise "12. November 2015 07:08 Uhr". Bitte beachten Sie: Sekunden werden nicht gespeichert.

### 2.1 Das Format

- Bei der Verwendung des ELO Java Clients haben die Formulare das gleiche Format. Dieses Format hat entweder ein bestimmtes Muster oder ist Locale-abhängig (in der Standardeinstellung).
- In anderen Fällen benutzt das Formular entweder das deutsche Format ("dd.MM.yyyy HH:mm") - es sei denn, die Sprache ist Ungarisch, in diesem Fall wird ("yyyy.MM.dd HH:mm") benutzt.

Mit anderen Worten: Falls Sie nicht den Java Client verwenden, können Sie andere Datums-/Zeitformate nicht benutzen.

Es empfiehlt sich, für das Datumsfeld beim Schreiben, Einfügen oder Aktualisieren das gleiche Format zu benutzen. Es ist allerdings möglich, auf die folgenden Alternativen auszuweichen:

- Das Alternativformat, das in dem Java Client angegeben wurde
- Das universelle Format ISO 8601 (z.B. "2014-09-08T08:02:17")

### 2.2 Internationalisierung & Zeitzonen

Bitte beachten Sie: Bei Anmeldung mit einem Client in einer anderen Zeitzone, werden IX\_DOCDATE und IX\_CREATEDATE für die örtliche Zeitzone angepasst.

Beispiel: Ein Dokument mit einem Erstellungsdatum vom "12. November 2015 07:08" wird zu "12. November 2015 10:08", wenn ein Benutzer sich an einem Client in einer drei Stunden späteren Zeitzone anmeldet. Dies ist auch der Fall bei einem "puren" Datum, hier wird die Zeit als 0 Uhr interpretiert. "12. November 2015" wird zum Beispiel gleich wie "12. November 2015 um 00:00" behandelt.

**Alle andere Datumswerte in dem Formular bleiben unverändert.**



## 3 Numerische Felder und Betragsfelder

### 3.1 Der Unterschied

Der Hauptunterschied zwischen Betragsfeldern und numerischen Felder besteht darin, dass Betragsfelder Trennzeichen benutzen, währenddessen "num"-Felder diese nicht haben. Betragsfelder werden typischerweise für große Zahlen oder Währungen eingesetzt. Die Tausender- und Dezimaltrennzeichen sind auch sprachabhängig. Ein Betragsfeld wie "123.456,78" auf Deutsch wird also als "123,456.78" im Englischen dargestellt.

### 3.2 Verhalten

"amount nk:2" verhält sich folgendermaßen (in der Client-Spracheinstellung Deutsch):

- "12345 -> "12.345,00" <= stellt ein Tausendertrennzeichen dar
- "12.345" -> "12.345,00" <= interpretiert "." als Tausendertrennzeichen
- "12,345" -> "12,35"

"num nk:2" verhält sich folgendermaßen (in der Client-Spracheinstellung Deutsch):

- "12345" -> "12345,00" <= Kein Tausendertrennzeichen
- "12.345" -> "12,35" <= interpretiert sowohl "," als auch "." als Dezimaltrennzeichen
- "12,345" -> "12,35"

Betragsfelder können Eingaben mit oder ohne Tausendertrennzeichen akzeptieren, werden aber die Daten immer mit Trennzeichen anzeigen.

Wenn der folgende Flag im Header-Skript gesetzt wird:

```
ELO.Configuration.Amount.noThousandSep = true;
```

Dann werden "amount"-Felder ähnlich zu "num"-Felder, die das letzte Trennzeichen als ein Dezimaltrennzeichen interpretieren, egal, ob es sich um einen Punkt "." oder ein Komma "," handelt.

### 3.3 Format

Was wird für die Tausender-/Dezimaltrennzeichen benutzt?

- Wenn der Benutzer den ELO Java Client einsetzt, wird es in den technischen Voreinstellungen der Konfiguration eingestellt. Dort haben Sie zwei Optionen: "Standard" (empfehlenswert, ist sprachabhängig). Alternativ dazu können Sie eigene Trennzeichen angeben (nicht empfehlenswert, weil das Trennzeichen immer gleich bleibt, auch wenn Sie die Sprache ändern).
- In anderen Fällen wird der Standard der Client-Spracheinstellung angewandt.

Falls Sie sich nicht sicher sind, welches Trennzeichen benutzt wird, können Sie die Einstellung in dem Formularheader ermitteln. Diese sind in den folgenden Variablen gespeichert:

```
ELO.Configuration.Amount.ThousandSep = ...  
ELO.Configuration.Amount.DecimalSep = ...
```

### 3.4 Formeln

Numerische Eingabefelder können über beliebige Formeln berechnet werden.

The screenshot shows a form editor interface. On the left, there is a preview of a form with three input fields labeled 'A:', 'B:', and 'A+B:'. The 'A:' field contains the value '11.111,00', the 'B:' field contains '222,00', and the 'A+B:' field contains '11.333,00'. A blue box highlights the 'A+B:' field. To the right of the form preview is a 'Properties of the selected cell' dialog box. This dialog has several fields: 'Field type' (Eingabefeld), 'Text' (empty), 'Variable name' (IX\_MAP\_SUM), 'Keyword list' (empty), 'Display' (empty), 'Tooltip' (empty), 'Validation' (amount nk:2), 'Formula' (IX\_MAP\_A + IX\_MAP\_B), 'Shortcut' (empty), 'Character count' (20), 'Form columns' (1), and a 'Read-only' checkbox which is checked. A blue arrow points from the 'Formula' field in the dialog to the 'A+B:' field in the form preview.

Die Sonderfunktion ESum(IX\_MAP\_N) als Formel kann benutzt werden, um alle Werte zu summieren. Es ist dasselbe wie  $IX\_MAP\_N1 + IX\_MAP\_N2 + \dots$

### 3.5 Spontane Formeln

Sie können in Formulare "=" eingeben, gefolgt mit einer mathematischen Formel wie "= 0.21 \* 1234 + 57", um etwas dynamisch zu berechnen. Drücken Sie danach einfach die Eingabetaste oder verlassen Sie das Feld; das Ergebnis wird automatisch berechnet.

Diese Inline-Formeln sind auf numerische Operationen beschränkt. Andere Variablen stehen nicht zur Verfügung.

## 3.6 Skripting

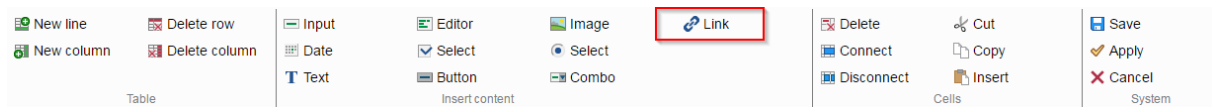
Wenn Sie Werte aus Feldern durch das Skripten holen, benutzen Sie in jedem Fall "\$num(...)" und nicht "\$val(...)". Die erste gibt ein Float zurück, wobei die Zweite einen "ELO-formatierten" String zurückgibt: Das heißt, kein Tausendertrennzeichen und ein Komma als Dezimaltrennzeichen, egal welche Sprache angegeben wird!

Enthält die Eingabe den Wert "12.345,67", gibt \$val("IX\_...") den String "12345,67" zurück.

Numerische Werte und Betragswerte werden ebenfalls immer in "ELO\_PARAMS" als ELO-formatierte Strings gespeichert.

## 4 Linkfelder

Ab ELO 10 können Sie auch Links in Formulare einfügen.

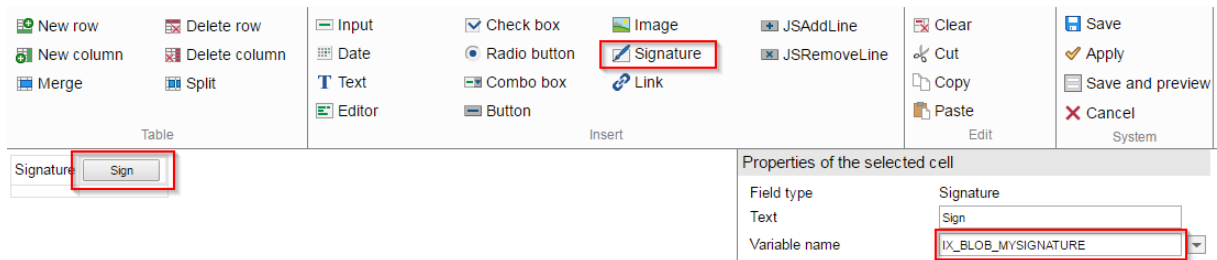


Es stehen zwei Arten von Links zur Verfügung:

- Ist die URL eine GUID, wird beim Klick das Dokument in einem neuen Fenster/Tab geöffnet.
- Ist die URL eine "normale" Internetadresse, wird die zugehörige Webseite in einem neuen Fenster/Tab geladen.

Wird die URL in dem Formulardesigner leer gelassen, zeigt das Formular einen Link auf das aktuelle Dokument an.

## 5 Zeichnungen



Das Signatur-Widget ist in ELOwf 10.01 und späteren Versionen verfügbar. Das Widget ermöglicht es dem Benutzer, ein Formular zu unterschreiben.

Da die Daten, die in einer Signatur gespeichert werden, größer als in den Indexfeldern und Map-Felder erlaubt sind, müssen "BLOB" (Binary Large Object)-Felder eingesetzt werden. Diese werden anhand von Variablen erkannt, die mit "IX\_BLOB\_" anfangen, gefolgt von einem beliebigen Namen.

Signature

Save without passing forward



---

Signed by: Administrator

Signature 

24.Nov.16 10.51

Save without passing forward

Es ist möglich, mehrere Signaturen auf einem Formular anzubringen, die von dem gleichen oder verschiedenen Benutzern unterzeichnet sind. Die einzige Beschränkung hierbei ist, dass dieses Widget nicht in Kombination mit dynamischen "JS\_ADDLINE"-Zeilen verwendet werden kann.

## 6 Validierung

Properties of the selected cell

Field type	Input
Text	<input type="text"/>
Variable name	<input type="text" value="IX_GRP_FOO"/>
Keyword list	<input type="text"/>
URL	<input type="text"/>
Display	<input type="text"/>
Tooltip	<input type="text"/>
Validation	<input type="text"/>
Validation message	<input type="text"/>
Formula	<input type="text"/>
Shortcut	<input type="text"/>
Character count	<input type="text" value="20"/> <input type="text"/>
	<input type="checkbox"/> Read only
Form columns	<input type="text" value="1"/>

### 6.1 Einleitung

Die Einschränkungen der Indexfelder auf einer Verschlagwortungsmaske und die Validierung in den ELOWf Formularfeldern sind im Grunde zwei ganz verschiedene Funktionalitäten.

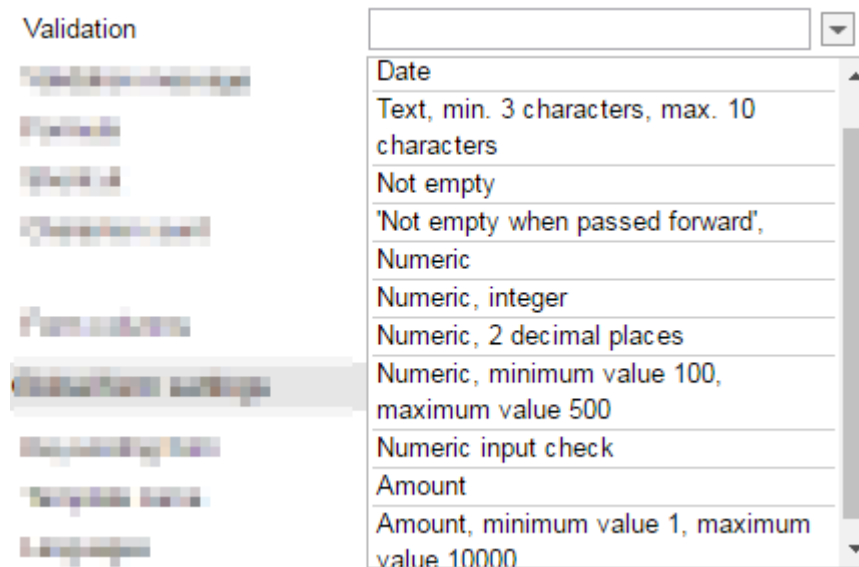
Die Einschränkungen, die in der Verschlagwortungsmaske angegeben sind, werden von *ELOWf und dem Indexserver ignoriert (\*)*. Sie können per Default alles mögliche in jedem Indexfeld speichern, auch wenn es als eine Nummer, Datum, oder alles andere definiert wurde.

Die Verschlagwortungsmasken werden ausschließlich in den Clients (Java, Web, Windows) benutzt, um die Inhalte zu validieren/formatieren, jedoch aber *nicht* von ELOWf, der eine andere Validierung einsetzt.

(\*) ...mit Ausnahme der Minimal-/Maximallänge, die jedoch die zwei einzigen Eigenschaften einer Verschlagwortungsmaske sind, die zwingend auf ein Feld beschränkt sind.

### 6.2 Grundverhalten

Sie können pro Feld eine Reihe vordefinierter Einschränkungen angeben:



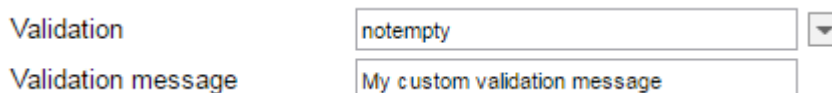
Um konsistent zu bleiben, sollen diese Einschränkungen gleich sein wie die in der Verschlagwortungsmaske – aber erforderlich ist das nicht.

Ein paar weitere Hinweise:

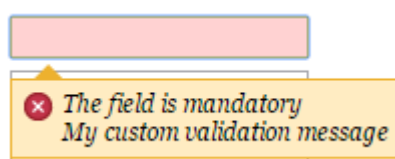
- Für numerische Werte bestimmt Min./Max. in der Verschlagwortungsmaske die Anzahl der Stellen, wobei diese Einstellung in dem Formulardesigner den Minimal-/Maximalwert bestimmt.
- Die Validierung 'nicht leer wenn weitergeleitet' wurde geschaffen, damit das Formular mit leeren Feldern "temporär" gespeichert werden kann, die zwingend gefüllt werden müssen, wenn das Formular innerhalb eines Workflows weitergeleitet wird.

## 6.3 Eigene Validierungsnachrichten

Zusätzlich zu den Standard-Validierungsnachrichten können Sie eigene Nachrichten schreiben.



Wird zu:



Die Nachricht wird zusätzlich zu den anderen Fehlermeldungen angezeigt. Es ist auch möglich, Übersetzungsschlüssel für die Meldung zu benutzen.



## 6.4 Eigene Validierungsfunktionen (JS\_VAL\_...)

Angepasste Funktionen können verwendet werden, um Eingaben zu validieren.

```
JS_VAL_myValidation(fieldName, fieldValue, validationParam)
```

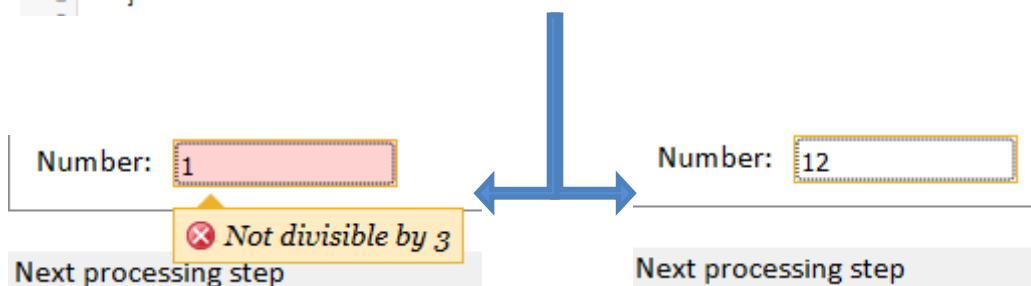
Für ein ungültiges Feld sollte die Funktion den Fehler als nicht-leeren String ausgeben.

Number:

Properties of the selected cell	
Field type	Eingabefeld
Text	
Variable name	IX_MAP_A
Keyword list	
Display	
Tooltip	
Validation	JS_VAL_DIV:3
Formula	
Shortcut	
Character count	14
	<input type="checkbox"/> Read-only
Form columns	1

### Edit form scripts

```
1 <script type='text/javascript'>
2
3 function JS_VAL_DIV(name, value, param) {
4     var num = +value;
5     if (num % param != 0) {
6         return "Not divisible by " + param;
7     }
8 }
```



Das kann zusätzlich zu anderen Validierungseinschränkungen vorgenommen werden. Diese Funktionalität ist nur dadurch eingeschränkt, indem es synchron sein muss.

## 6.5 Eigene Filterfunktionen (JS\_FILTER\_...)

Gibt der Benutzer etwas in einem Feld ein, kann ein Filter eingesetzt werden, um den Wert umzuwandeln.

Die vordefinierte Funktion JS\_FILTER\_NUMBER beispielsweise erlaubt es dem Benutzer, nur Ziffern einzugeben.

Eigene Filter können eingesetzt werden, um nur Zahlen zu erlauben, die Eingaben in Großbuchstaben umzuwandeln, ein bestimmtes Format zu erzwingen oder jede beliebige andere Art von Texttransformation zu übernehmen.

Um das zu machen, geben Sie "JS\_FILTER\_MeineFunktion" oder "JS\_FILTER\_MeineFunktion:Parameter" in dem Feld "Validierung" ein. Die folgende Funktion wird aufgerufen, wenn ein Benutzer etwas in dem Feld eingibt oder einfügt.

```
JS_FILTER_MyFunc(vorne, eingegeben, danach, param)
```

Hier ist ein Beispiel, wie Sie die Eingaben in Großbuchstaben umwandeln können:

```
function JS_FILTER_Uppercase(vorne, eingegeben, danach) {  
    return eingegeben.toUpperCase();  
}
```

In diesem Fall wird der Rückgabewert ein String, der die Eingaben in modifizierter Form darstellt. Es ist allerdings auch möglich, die Angaben vor und nach der Eingaben zu modifizieren. Sie erreichen das, indem sie ein Array mit den neuen Werten zurückgeben ([vorne, eingegeben, danach]).

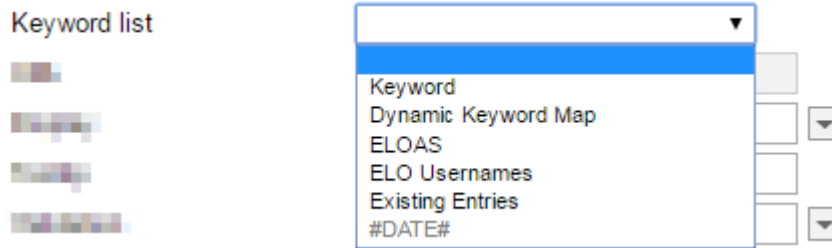
## 6.6 Kann ich je nach Anwendungsfall die Validierung ein- oder ausschalten?

Nein. Die Validierung findet immer statt. Es ist auch sinnvoll. Auch wenn Sie einen Workflow mit einer "Abbrechen"-Operation vorliegen haben, muss das Formular immerhin gespeichert werden. Und um das speichern zu können, muss es gültige Daten beinhalten.

Anstatt die Validierung zu entfernen, empfehlen wir die Anwendung von "onNextClicked(...)", um einfach Werte ggs. mit Dummy-Werten zu füllen, bevor die Validierung stattfindet.

## 7 Listen

### 7.1 Übersicht



Eine Erklärung für diese Listentypen:

- **Stichwort:** Das kann eine statische oder dynamische Stichwortliste sein, je nachdem, was für das Feld in der Verschlagwortungsmaske definiert wurde.
- **Dynamische Stichwortliste:** Diese Liste wird typischerweise benutzt, um Werte für Map-Felder zu holen.
- **ELOAS:** Ruft ein ELOas-Ruleset auf und gibt die Ergebnisse als eine Liste zurück.
- **ELO Benutzernamen:** Ruft eine Liste von Benutzern / Gruppen ab.
- **Bestehende Einträge:** Wählen Sie diese Option aus, erhält der Benutzer eine Autovervollständigungs-Liste, die aus den vorher in dem Feld eingetragenen Werten in anderen Dokumenten basiert.

### 7.2 Dynamische Stichwortliste

Ein dynamisches Stichwortfeld ist ein Feld, das mehrere Spalten Inhalt darstellen kann, typischerweise dynamisch gefiltert nach einem oder mehreren anderen Feldwerten. Es sieht folgendermaßen aus:

ab	
Code	Name
GA	Gabon
LY	Libyan Arab Jamahiriya
SA	Saudi Arabia
SY	Syrian Arab Republic
AE	United Arab Emirates
ZW	Zimbabwe

Dynamische Stichwortlisten sind ein relativ komplexes Feature in ELO, das für die Funktionalität Skripting-Kenntnisse voraussetzt. Sie sind im ELO Java Client und dem ELO Web Client verfügbar.

Sie können Folgendes damit tun:

- Mehrere Spalten von Daten anzeigen
- Inhalte dynamisch abrufen (zum Beispiel: externe Datenbanken, dynamisch berechnete Daten, ELO Skripte...)
- Daten können gefiltert werden, nicht nur in einem Feld, sondern auch abhängig von beliebig anderen Feldern (zum Beispiel: was der Benutzer eingegeben hat und eine Kategorie, die der Benutzer zuvor ausgewählt hat)
- Füllen weiterer anderer Felder mit Eintragsauswahl (inklusive der Nur-Lesen-Felder)

Die Verwendung einer dynamischen Stichwortliste wird direkt in der Verschlagwortungsmaske definiert:

**Add or edit index fields for keywording form: Biggy**

**Fields**

- Field 1
- Field 2
- Field 3
- Field 4
- Field 5
- Field 6
- Field 7
- Field 8
- Field 9
- Field 10
- Field 11
- Field 12
- Field 13
- Field 14
- Field 15
- Field 16
- Field 17
- Field 18
- Field 19
- Field 20
- Field 21
- Field 22
- Field 23
- Field 24
- Field 25
- Field 26
- Field 27
- Field 28
- Field 29
- Field 30
- Field 31
- Field 32
- Field 33
- Field 34
- Field 35
- Field 36
- Field 37
- Field 38
- Field 39
- Field 40
- Field 41
- Field 42
- Field 43
- Field 44
- Field 45
- Field 46
- Field 47
- Field 48
- Field 49
- Field 50
- Field 51
- Field 52
- Field 53
- Field 54
- Field 55
- Field 56
- Field 57
- Field 58
- Field 59
- Field 60
- Field 61
- Field 62
- Field 63
- Field 64
- Field 65
- Field 66
- Field 67
- Field 68
- Field 69
- Field 70
- Field 71
- Field 72
- Field 73
- Field 74
- Field 75
- Field 76
- Field 77
- Field 78
- Field 79
- Field 80
- Field 81
- Field 82
- Field 83
- Field 84
- Field 85
- Field 86
- Field 87
- Field 88
- Field 89
- Field 90
- Field 91
- Field 92
- Field 93
- Field 94
- Field 95
- Field 96
- Field 97
- Field 98
- Field 99
- Field 100
- Field 101
- Field 102
- Field 103
- Field 104
- Field 105
- Field 106
- Field 107
- Field 108
- Field 109
- Field 110
- Field 111
- Field 112
- Field 113
- Field 114
- Field 115
- Field 116
- Field 117
- Field 118
- Field 119
- Field 120
- Field 121
- Field 122
- Field 123
- Field 124
- Field 125
- Field 126
- Field 127
- Field 128
- Field 129
- Field 130
- Field 131
- Field 132
- Field 133
- Field 134
- Field 135
- Field 136
- Field 137
- Field 138
- Field 139
- Field 140
- Field 141
- Field 142
- Field 143
- Field 144
- Field 145
- Field 146
- Field 147
- Field 148
- Field 149
- Field 150
- Field 151
- Field 152
- Field 153
- Field 154
- Field 155
- Field 156
- Field 157
- Field 158
- Field 159
- Field 160
- Field 161
- Field 162
- Field 163
- Field 164
- Field 165
- Field 166
- Field 167
- Field 168
- Field 169
- Field 170
- Field 171
- Field 172
- Field 173
- Field 174
- Field 175
- Field 176
- Field 177
- Field 178
- Field 179
- Field 180
- Field 181
- Field 182
- Field 183
- Field 184
- Field 185
- Field 186
- Field 187
- Field 188
- Field 189
- Field 190
- Field 191
- Field 192
- Field 193
- Field 194
- Field 195
- Field 196
- Field 197
- Field 198
- Field 199
- Field 200
- Field 201
- Field 202
- Field 203
- Field 204
- Field 205
- Field 206
- Field 207
- Field 208
- Field 209
- Field 210
- Field 211
- Field 212
- Field 213
- Field 214
- Field 215
- Field 216
- Field 217
- Field 218
- Field 219
- Field 220
- Field 221
- Field 222
- Field 223
- Field 224
- Field 225
- Field 226
- Field 227
- Field 228
- Field 229
- Field 230
- Field 231
- Field 232
- Field 233
- Field 234
- Field 235
- Field 236
- Field 237
- Field 238
- Field 239
- Field 240
- Field 241
- Field 242
- Field 243
- Field 244
- Field 245
- Field 246
- Field 247
- Field 248
- Field 249
- Field 250
- Field 251
- Field 252
- Field 253
- Field 254
- Field 255
- Field 256
- Field 257
- Field 258
- Field 259
- Field 260
- Field 261
- Field 262
- Field 263
- Field 264
- Field 265
- Field 266
- Field 267
- Field 268
- Field 269
- Field 270
- Field 271
- Field 272
- Field 273
- Field 274
- Field 275
- Field 276
- Field 277
- Field 278
- Field 279
- Field 280
- Field 281
- Field 282
- Field 283
- Field 284
- Field 285
- Field 286
- Field 287
- Field 288
- Field 289
- Field 290
- Field 291
- Field 292
- Field 293
- Field 294
- Field 295
- Field 296
- Field 297
- Field 298
- Field 299
- Field 300
- Field 301
- Field 302
- Field 303
- Field 304
- Field 305
- Field 306
- Field 307
- Field 308
- Field 309
- Field 310
- Field 311
- Field 312
- Field 313
- Field 314
- Field 315
- Field 316
- Field 317
- Field 318
- Field 319
- Field 320
- Field 321
- Field 322
- Field 323
- Field 324
- Field 325
- Field 326
- Field 327
- Field 328
- Field 329
- Field 330
- Field 331
- Field 332
- Field 333
- Field 334
- Field 335
- Field 336
- Field 337
- Field 338
- Field 339
- Field 340
- Field 341
- Field 342
- Field 343
- Field 344
- Field 345
- Field 346
- Field 347
- Field 348
- Field 349
- Field 350
- Field 351
- Field 352
- Field 353
- Field 354
- Field 355
- Field 356
- Field 357
- Field 358
- Field 359
- Field 360
- Field 361
- Field 362
- Field 363
- Field 364
- Field 365
- Field 366
- Field 367
- Field 368
- Field 369
- Field 370
- Field 371
- Field 372
- Field 373
- Field 374
- Field 375
- Field 376
- Field 377
- Field 378
- Field 379
- Field 380
- Field 381
- Field 382
- Field 383
- Field 384
- Field 385
- Field 386
- Field 387
- Field 388
- Field 389
- Field 390
- Field 391
- Field 392
- Field 393
- Field 394
- Field 395
- Field 396
- Field 397
- Field 398
- Field 399
- Field 400
- Field 401
- Field 402
- Field 403
- Field 404
- Field 405
- Field 406
- Field 407
- Field 408
- Field 409
- Field 410
- Field 411
- Field 412
- Field 413
- Field 414
- Field 415
- Field 416
- Field 417
- Field 418
- Field 419
- Field 420
- Field 421
- Field 422
- Field 423
- Field 424
- Field 425
- Field 426
- Field 427
- Field 428
- Field 429
- Field 430
- Field 431
- Field 432
- Field 433
- Field 434
- Field 435
- Field 436
- Field 437
- Field 438
- Field 439
- Field 440
- Field 441
- Field 442
- Field 443
- Field 444
- Field 445
- Field 446
- Field 447
- Field 448
- Field 449
- Field 450
- Field 451
- Field 452
- Field 453
- Field 454
- Field 455
- Field 456
- Field 457
- Field 458
- Field 459
- Field 460
- Field 461
- Field 462
- Field 463
- Field 464
- Field 465
- Field 466
- Field 467
- Field 468
- Field 469
- Field 470
- Field 471
- Field 472
- Field 473
- Field 474
- Field 475
- Field 476
- Field 477
- Field 478
- Field 479
- Field 480
- Field 481
- Field 482
- Field 483
- Field 484
- Field 485
- Field 486
- Field 487
- Field 488
- Field 489
- Field 490
- Field 491
- Field 492
- Field 493
- Field 494
- Field 495
- Field 496
- Field 497
- Field 498
- Field 499
- Field 500
- Field 501
- Field 502
- Field 503
- Field 504
- Field 505
- Field 506
- Field 507
- Field 508
- Field 509
- Field 510
- Field 511
- Field 512
- Field 513
- Field 514
- Field 515
- Field 516
- Field 517
- Field 518
- Field 519
- Field 520
- Field 521
- Field 522
- Field 523
- Field 524
- Field 525
- Field 526
- Field 527
- Field 528
- Field 529
- Field 530
- Field 531
- Field 532
- Field 533
- Field 534
- Field 535
- Field 536
- Field 537
- Field 538
- Field 539
- Field 540
- Field 541
- Field 542
- Field 543
- Field 544
- Field 545
- Field 546
- Field 547
- Field 548
- Field 549
- Field 550
- Field 551
- Field 552
- Field 553
- Field 554
- Field 555
- Field 556
- Field 557
- Field 558
- Field 559
- Field 560
- Field 561
- Field 562
- Field 563
- Field 564
- Field 565
- Field 566
- Field 567
- Field 568
- Field 569
- Field 570
- Field 571
- Field 572
- Field 573
- Field 574
- Field 575
- Field 576
- Field 577
- Field 578
- Field 579
- Field 580
- Field 581
- Field 582
- Field 583
- Field 584
- Field 585
- Field 586
- Field 587
- Field 588
- Field 589
- Field 590
- Field 591
- Field 592
- Field 593
- Field 594
- Field 595
- Field 596
- Field 597
- Field 598
- Field 599
- Field 600
- Field 601
- Field 602
- Field 603
- Field 604
- Field 605
- Field 606
- Field 607
- Field 608
- Field 609
- Field 610
- Field 611
- Field 612
- Field 613
- Field 614
- Field 615
- Field 616
- Field 617
- Field 618
- Field 619
- Field 620
- Field 621
- Field 622
- Field 623
- Field 624
- Field 625
- Field 626
- Field 627
- Field 628
- Field 629
- Field 630
- Field 631
- Field 632
- Field 633
- Field 634
- Field 635
- Field 636
- Field 637
- Field 638
- Field 639
- Field 640
- Field 641
- Field 642
- Field 643
- Field 644
- Field 645
- Field 646
- Field 647
- Field 648
- Field 649
- Field 650
- Field 651
- Field 652
- Field 653
- Field 654
- Field 655
- Field 656
- Field 657
- Field 658
- Field 659
- Field 660
- Field 661
- Field 662
- Field 663
- Field 664
- Field 665
- Field 666
- Field 667
- Field 668
- Field 669
- Field 670
- Field 671
- Field 672
- Field 673
- Field 674
- Field 675
- Field 676
- Field 677
- Field 678
- Field 679
- Field 680
- Field 681
- Field 682
- Field 683
- Field 684
- Field 685
- Field 686
- Field 687
- Field 688
- Field 689
- Field 690
- Field 691
- Field 692
- Field 693
- Field 694
- Field 695
- Field 696
- Field 697
- Field 698
- Field 699
- Field 700
- Field 701
- Field 702
- Field 703
- Field 704
- Field 705
- Field 706
- Field 707
- Field 708
- Field 709
- Field 710
- Field 711
- Field 712
- Field 713
- Field 714
- Field 715
- Field 716
- Field 717
- Field 718
- Field 719
- Field 720
- Field 721
- Field 722
- Field 723
- Field 724
- Field 725
- Field 726
- Field 727
- Field 728
- Field 729
- Field 730
- Field 731
- Field 732
- Field 733
- Field 734
- Field 735
- Field 736
- Field 737
- Field 738
- Field 739
- Field 740
- Field 741
- Field 742
- Field 743
- Field 744
- Field 745
- Field 746
- Field 747
- Field 748
- Field 749
- Field 750
- Field 751
- Field 752
- Field 753
- Field 754
- Field 755
- Field 756
- Field 757
- Field 758
- Field 759
- Field 760
- Field 761
- Field 762
- Field 763
- Field 764
- Field 765
- Field 766
- Field 767
- Field 768
- Field 769
- Field 770
- Field 771
- Field 772
- Field 773
- Field 774
- Field 775
- Field 776
- Field 777
- Field 778
- Field 779
- Field 780
- Field 781
- Field 782
- Field 783
- Field 784
- Field 785
- Field 786
- Field 787
- Field 788
- Field 789
- Field 790
- Field 791
- Field 792
- Field 793
- Field 794
- Field 795
- Field 796
- Field 797
- Field 798
- Field 799
- Field 800
- Field 801
- Field 802
- Field 803
- Field 804
- Field 805
- Field 806
- Field 807
- Field 808
- Field 809
- Field 810
- Field 811
- Field 812
- Field 813
- Field 814
- Field 815
- Field 816
- Field 817
- Field 818
- Field 819
- Field 820
- Field 821
- Field 822
- Field 823
- Field 824
- Field 825
- Field 826
- Field 827
- Field 828
- Field 829
- Field 830
- Field 831
- Field 832
- Field 833
- Field 834
- Field 835
- Field 836
- Field 837
- Field 838
- Field 839
- Field 840
- Field 841
- Field 842
- Field 843
- Field 844
- Field 845
- Field 846
- Field 847
- Field 848
- Field 849
- Field 850
- Field 851
- Field 852
- Field 853
- Field 854
- Field 855
- Field 856
- Field 857
- Field 858
- Field 859
- Field 860
- Field 861
- Field 862
- Field 863
- Field 864
- Field 865
- Field 866
- Field 867
- Field 868
- Field 869
- Field 870
- Field 871
- Field 872
- Field 873
- Field 874
- Field 875
- Field 876
- Field 877
- Field 878
- Field 879
- Field 880
- Field 881
- Field 882
- Field 883
- Field 884
- Field 885
- Field 886
- Field 887
- Field 888
- Field 889
- Field 890
- Field 891
- Field 892
- Field 893
- Field 894
- Field 895
- Field 896
- Field 897
- Field 898
- Field 899
- Field 900
- Field 901
- Field 902
- Field 903
- Field 904
- Field 905
- Field 906
- Field 907
- Field 908
- Field 909
- Field 910
- Field 911
- Field 912
- Field 913
- Field 914
- Field 915
- Field 916
- Field 917
- Field 918
- Field 919
- Field 920
- Field 921
- Field 922
- Field 923
- Field 924
- Field 925
- Field 926
- Field 927
- Field 928
- Field 929
- Field 930
- Field 931
- Field 932
- Field 933
- Field 934
- Field 935
- Field 936
- Field 937
- Field 938
- Field 939
- Field 940
- Field 941
- Field 942
- Field 943
- Field 944
- Field 945
- Field 946
- Field 947
- Field 948
- Field 949
- Field 950
- Field 951
- Field 952
- Field 953
- Field 954
- Field 955
- Field 956
- Field 957
- Field 958
- Field 959
- Field 960
- Field 961
- Field 962
- Field 963
- Field 964
- Field 965
- Field 966
- Field 967
- Field 968
- Field 969
- Field 970
- Field 971
- Field 972
- Field 973
- Field 974
- Field 975
- Field 976
- Field 977
- Field 978
- Field 979
- Field 980
- Field 981
- Field 982
- Field 983
- Field 984
- Field 985
- Field 986
- Field 987
- Field 988
- Field 989
- Field 990
- Field 991
- Field 992
- Field 993
- Field 994
- Field 995
- Field 996
- Field 997
- Field 998
- Field 999
- Field 1000

**Field information**

Name: Dyn List

Name (locale key):

Group: **DYN\_LIST**

Min. input length: 0 Max. input length: 255

Input type: Text

Default value:

Rights: Everyone

External data:

Quick info:

Quick info (locale key):

Dynamic keyword list: **CountryCodes**

Tab assignment: Basis

☐ Only accept data from keyword list

☐ Translated keyword list

☐ Automatically add \* before search term

☐ Automatically add \* after search term

☐ Index field with high priority

☐ Show index field when checking in

☒ Search with [F7] allowed

☐ Pass on input to child entries

☒ Allow input inheritance from parent element

☐ Not split into tokens

☒ Normal access

☐ Read-only

☐ Not visible

**Keyword list**

**Position fields**

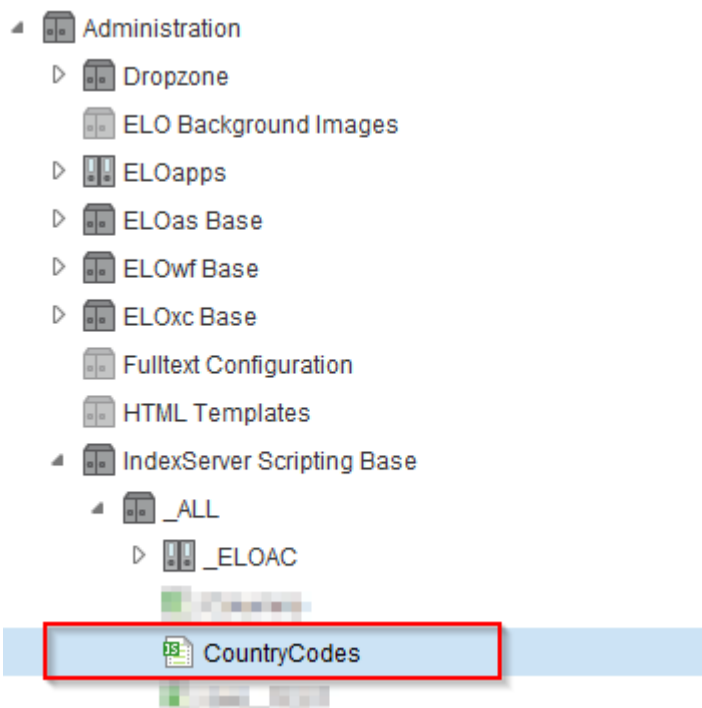
**Buttons:** OK, Cancel, New field, Delete

Um diese Listen in einem Formular zu verwenden, müssen sie in dem Formulareditor folgendermaßen deklariert werden:

Properties of the selected cell

Field type	Input
Text	<input type="text"/>
Variable name	IX_MAP_FOO <input type="button" value="v"/>
Keyword list	Keyword <input type="button" value="v"/>
Group name	DYN_LIST
	<input checked="" type="checkbox"/> Autofill
	<input type="checkbox"/> Only list values allowed

Das entsprechende Skript "CountryCodes" muss natürlich vorhanden sein und in dem Ordner Administration\IndexServer Scripting Base liegen.



Hier ist ein Beispiel für ein Skript, der eine statische Liste von Länderkürzel und -namen bereitstellt:

```
importPackage(Packages.de.elo.ix.jscript);  
importPackage(Packages.de.elo.ix.scripting);
```

```
function getDataIterator() {
    try {
        log.info("getDataIterator(SimpleDatabaseQuery)");
        return new DynamicKeywordDataProvider(new CountryCodes());
    } finally {
        log.info(")getDataIterator");
    }
}

function CountryCodes() {
    var index = 0;
    var results = [];

    /* Hilfefunktion, die eine Länderliste filtert */
    this.filterCountries = function(filter) {
        log.info("filter: " + filter)
        filter = filter.toLowerCase()
        results = [];
        for (var i=0; i<isoCountries.length; i++){
            if (isoCountries[i].cname.toLowerCase().indexOf(filter) >= 0) {
                results.push([isoCountries[i].ccode, isoCountries[i].cname]);
            }
        }
        log.info("Nach Filter: " + results.length);
    }

    /* Wird bei der Initialisierung einer dynamischen Liste aufgerufen (durch
    Deklaration in der Verschlagwortungsmaske) */
    this.open = function(ec, sord, focus) {
        log.info("öffnen");

        this.target = focus;
        // TODO: In diesem Fall wird das erste Indexfeld ausgewählt,
        // idealerweise sollte aber ein anderes ausgewählt werden
        var filter = sord.objKeys[0].data[0] || "";

        this.filterCountries(filter);
    }

    /* Wird aufgerufen, wenn eine dynamische Liste durch das ELOwf-Feld "Dyn.
    keywording map" initialisiert wird */
    this.openMap = function(ec, map, focus) {
        log.info("openMap");
        log.info(JSON.stringify(map));

        this.target = focus;
        var filter = map[focus] || "";
    }
}
```

```
this.filterCountries(filter);
}

this.close = function() {
    log.info("schließen()");
}

this.getNextRow = function() {
    var row = results[index];
    index++;
    log.info("getNextRow(" + row + ")");
    return row;
}

/* Die Benennung der Spalten */
this.getHeader = function() {
    log.info("getHeader()");
    return ["Code", "Name"];
}

/* Das Ziel des ausgewählten Eintrags */
this.getKeyNames = function() {
    log.info("getKeyNames()");
    return [this.focus, "EIN_ANDERES_INDEXFELD"];
}

this.hasMoreRows = function() {
    log.info("hasMoreRows()");
    return (index < results.length - 1);
}

/* Entweder wird eine Fehlermeldung zurückgegeben, oder bei Erfolg leer
gelassen */
this.getMessage = function() {
    log.info("getMessage()");
    return "";
}

this.getTitle = function() {
    log.info("getTitle()");
    return "Country Codes";
}
}
```

```
var isoCountries = [  
  {'ccode' : 'AF', 'cname' : 'Afghanistan'},  
  {'ccode' : 'AX', 'cname' : 'Aland Islands'},  
  {'ccode' : 'AL', 'cname' : 'Albania'},  
  /* ... */  
  {'ccode' : 'ZW', 'cname' : 'Zimbabwe'}  
]
```

See also:

<http://forum.elo.com/supportweb/eloimages/en-elo-dynamic-keyword-lists.19901.pdf>

<http://www.forum.elo.com/jforum/posts/list/3839.page#19317>

## 7.3 Dynamische Stichwort-Map

Eine dynamische Stichwort-Map ist ähnlich wie im vorherigen Beispiel, basiert aber auf Map-Feldern. Hierbei sollen Sie das Zielskript direkt angeben, wie im Screenshot unten abgebildet.

Properties of the selected cell

Field type	Input
Text	<input type="text"/>
Variable name	<input type="text" value="IX_MAP_FOO"/>
Keyword list	<input type="text" value="Dynamic Keyword Map"/>
Script name	<input type="text" value="CountryCodes"/>
Filter	<input type="text"/>
	<input checked="" type="checkbox"/> Autofill
	<input type="checkbox"/> Only list values allowed

### Hinweise

Wenn ein Eintrag in einer dynamischen Stichwortliste ausgewählt wird, wird die folgende Ereignisfunktion in dem Headerskript aufgerufen:

```
onDynListItemSelected(eintrag)
```





Bitte beachten Sie: Es ist in dynamischen Stichwortlisten nicht möglich, "Autovervollständigen" mit "Nur Listeneinträge erlaubt" zu kombinieren. Der Hauptgrund hierfür ist, dass dynamische Stichwortlisten von mehreren Eingabefeldern abhängen können, sowie mehrere Eingabefelder modifizieren können. Es ist aktuell nicht möglich, eine Validierung in mehreren Feldern gleichzeitig zu unterstützen.

Alle Listen können auch innerhalb der Skripte angestoßen werden.

```
/**
 * Ruft eine bestimmte Regel in ELOas auf.
 */
function $listAs(skriptName, param2, param3, beiErfolg, beiFehler) {

/**
 * $listKw("IX_GRP_DYN_FIELD", ...) holt die Daten aus der Stichwort-
 * Liste, die in dem angegebenen Feld definiert wurde
 */
function $listKw(swldid, beiErfolg, beiFehler) {

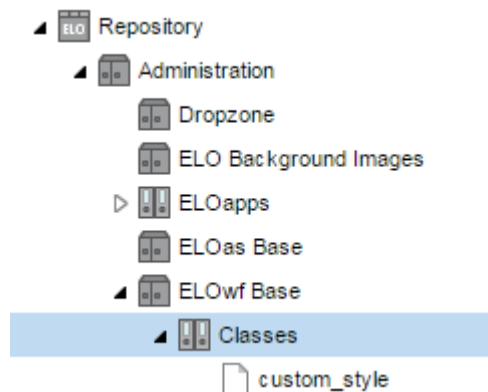
/**
 * $listDyn("MeinSkript", "foo", ["bar"], ...) holt die Daten vom
 * entsprechenden Skript in 'IndexServer Scripting Base'.
 * Das Skript wird mit "foo" als Fokusname aufgerufen und {"foo": ...,
 * "bar": ...} als Map-Daten, auch mit den Wildcards "{i}" und "{*}"
 * ersetzt
 */
function $listDyn(skriptName, fokusfeld, filterfelder, beiErfolg,
beiFehler) {
```

## 8 Eigene Dateien hinzufügen

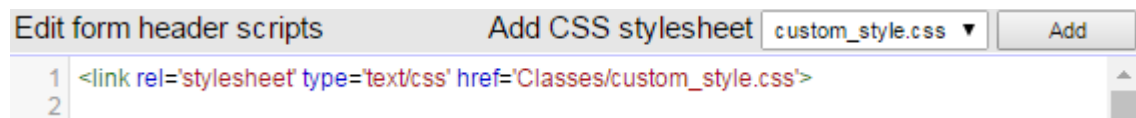
### 8.1 CSS-Dateien

Es ist möglich, Formulare umzugestalten, indem Sie Inline-CSS in dem Formular-Header hinzufügen.

Falls Sie umfangreichere Gestaltungsmöglichkeiten benötigen, sollen Sie am besten CSS-Dateien unter ¶Administration¶ELOwf Base¶Classes ablegen.



Nachdem Sie die Dateien dort abgelegt haben, müssen Sie ELOwf neu laden, damit diese neuen Dateien erkannt werden. Danach sind sie über den Header des Formulareditors verfügbar:



Wählen Sie ein Stylesheet und klicken Sie auf *Hinzufügen*, um die entsprechende Zeile im Header hinzuzufügen.

## 9 Ereignisse

Dieses Kapitel beschreibt einige Ereignisse, die in ELowf-Formulare abgefeuert werden.

### 9.1 onInit()

Dieses Ereignis wird einmal aufgerufen, wenn das Formular geladen wird.

### 9.2 inputChanged(elem)

Dieses Ereignis wird immer aufgerufen, wenn der Benutzer die Inhalte des Formulars ändert (nach jedem Tastendruck). Das Eingabefeld wird als Argument mitgeliefert. Das Ereignis wird nicht aufgerufen, wenn ein Skript einen Wert ändert.

### 9.3 saveClicked()

Das Ereignis 'saveClicked' wird aufgerufen, wenn ein Formular gespeichert wird. Genauer gesagt, es wird aufgerufen, bevor das Formular validiert und gespeichert wird.

Dieses Ereignis eignet sich, um zusätzliche Validierungen durchzuführen oder andere Variablen in dem Formular zu setzen, bevor es gespeichert wird.

Beispiel:

```
function saveClicked() {
    if( $num("IX_GRP_PRICE") > 10000 && $val("IX_GRP_TYPE") == "GUT&GÜNSTIG" ) {
        eloAlert("Sorry, das ist zu teuer!");
        return false; // Nicht speichern und abbrechen
    }
    else {
        return true; // ok, weiter mit Validierung und Speichern
    }
}
```

Ab ELowf 10.1 können auch asynchrone Prozesse ausgeführt werden, indem sie Promises zurückgeben:

```
function doSomething(resolve, reject) {
    // Daten asynchron holen oder verifizieren
    // 'resolve()' aufrufen, falls erfolgreich abgeschlossen
    // oder 'reject()' falls Prozess abgebrochen werden soll
}

function saveClicked() {
    return new Promise(doSomething);
}
```

Das Promise ist ein verzögertes Ergebnis, das verwendet wird, um den Prozess fortzuführen, falls es beschlossen oder abgelehnt wird.

## 9.4 nextClicked(id)

Dieses Ereignis funktioniert identisch zu "saveClicked", aber es wird aufgerufen, falls das Formular weiter zum nächsten Workflowknoten weitergeleitet wird. Es kann ebenfalls "true" zurückgeben, um weiter zu machen, "false", um abubrechen, oder ein Promise für eine verzögerte Auslösung.

Der einzige funktionelle Unterschied zwischen diesem und dem vorherigen Ereignis ist ein zusätzlicher Parameter "id", der die ID des Workflowknotens referenziert, zu dem der Workflow weitergeleitet wird.

Statt der ID ist es auch möglich, den internen Knotennamen wiederherzustellen:

```
var internalNodeName = ELO_PARAMS["NEXT_" + id].split('\t')[1]
```

## 10 Verschiedenes

### 10.1 Drucken



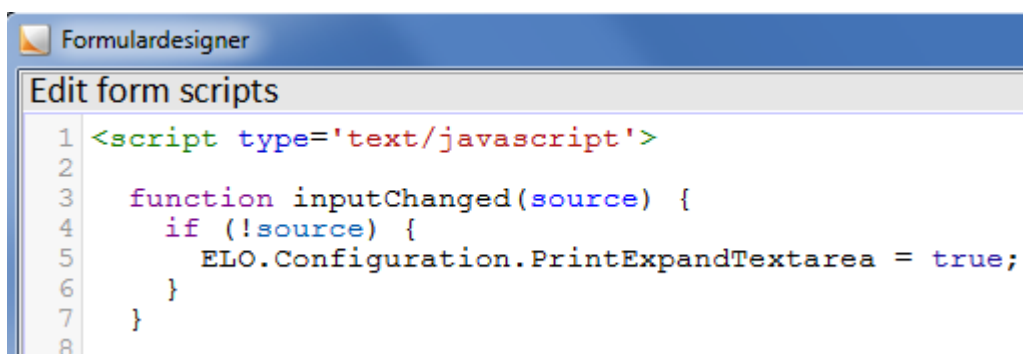
**Bitte beachten Sie:** Die folgenden Funktionen werden nur ausgeführt, wenn der Benutzer über die Schaltfläche *Drucken* etwas ausdruckt. Die Features werden nicht funktionieren, falls der Benutzer über *Strg+P* ausdruckt, oder über das Browsermenü geht.

#### 10.1.1 Textbereiche automatisch erweitern

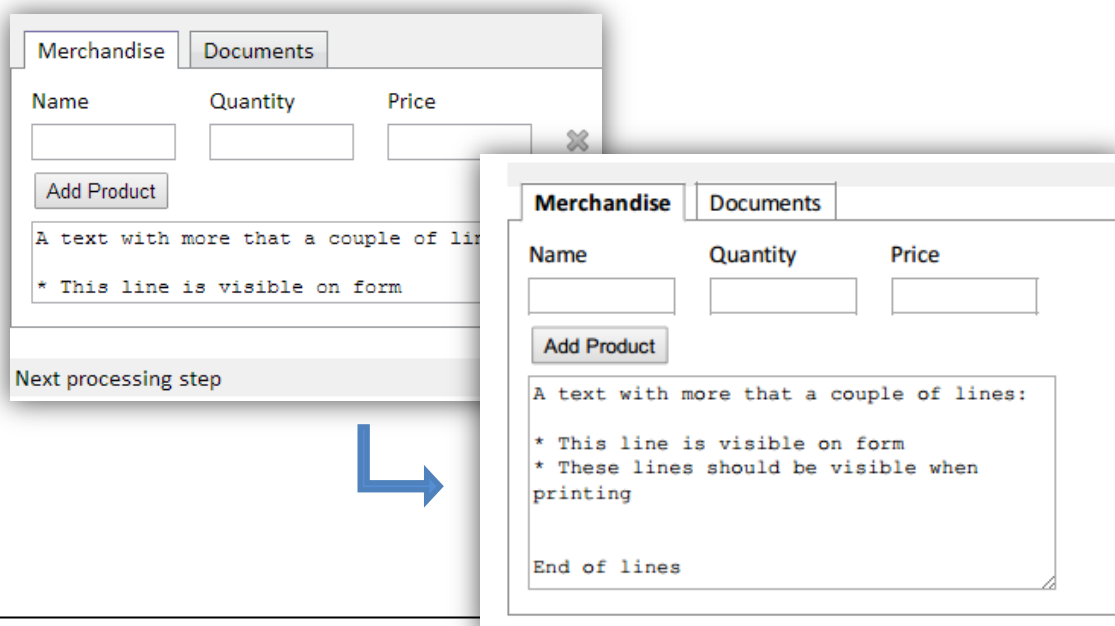
Beim Druck kann man das folgende Flag setzen, um die vollständigen Inhalte der Textbereiche zu sehen:

```
ELO.Configuration.PrintExpandTextarea = true;
```

Fügen Sie dieses Flag in der Ereignisfunktion `inputChanged()` ein.



Nach Ausführung werden alle Textbereiche ausreichend größer gemacht.



### 10.1.1.1 Alle Reiter drucken

Falls Sie alle Reiter standardmäßig ausdrucken wollen, können Sie das folgende Flag setzen:

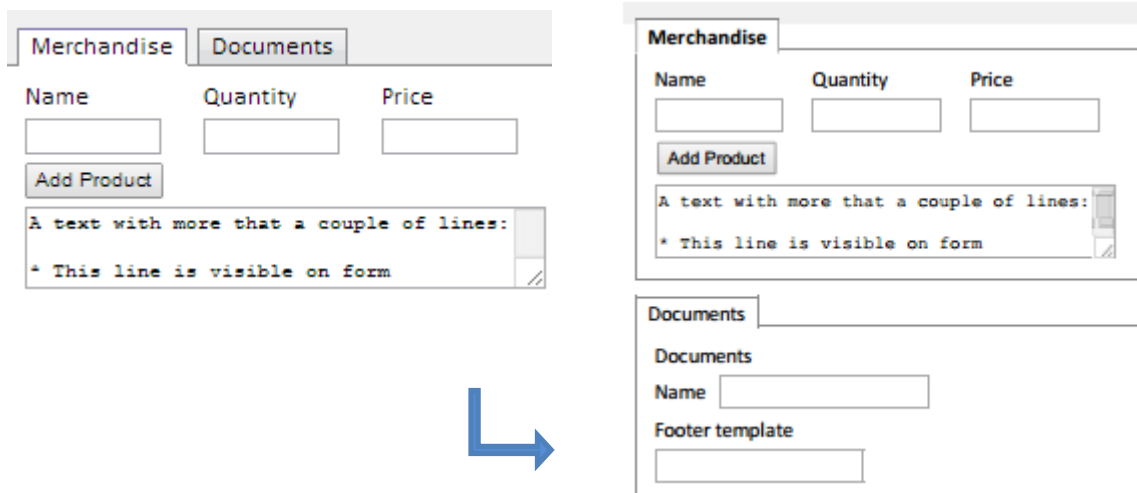
```
ELO.Configuration.PrintAllTabs = true;
```

Fügen Sie dieses Flag in der Ereignisfunktion `inputChanged()` ein.

**Edit form scripts**

```
1 <script type='text/javascript'>
2
3   function inputChanged(source) {
4     if (!source) {
5       ELO.Configuration.PrintAllTabs = true;
6     }
7   }
8 }
```

Abb. 1: JavaScript-Code: Funktion 'inputChanged'



## 10.2 Einstellbare Tastaturbefehle



**Bitte beachten Sie:** Diese Funktionen werden eingeschränkt unterstützt, je nach angewendetem Client und Browser.

Tastaturkürzel funktionieren nur, wenn das Formular direkt geöffnet wird, oder im ELO Web Client in Google Chrome oder im Internet Explorer. In Firefox wird stattdessen das Browsermenü über die Alt-Schaltfläche geöffnet, und der Java Client hat überhaupt keine Unterstützung für diese Funktionalität. Andere Kombinationen von Browsern/Clients werden nicht supportet.

Es ist möglich, Tastaturkürzel einzustellen, um den Fokus direkt auf ein bestimmtes Feld zu setzen:

**Properties of the selected cell**

Field type	Input
Text	<input type="text"/>
Variable name	<input type="text" value="IX_GRP_FOO"/>
Keyword list	<input type="text"/>
URL	<input type="text"/>
Display	<input type="text"/>
Tooltip	<input type="text"/>
Validation	<input type="text"/>
Validation message	<input type="text"/>
Formula	<input type="text"/>
Keyboard shortcut	<input type="text" value="a"/>
Character count	<input type="text" value="20"/>
	<input type="checkbox"/> Read-only
Form columns	<input type="text" value="1"/>

**Global form settings**

Keywording form	<input type="text" value="7: Biggy Form DE"/>
Name	<input type="text" value="basic"/>
Languages	<input type="text"/>
Translation keys	<input type="text"/>
	<input type="checkbox"/> Limited variable access
	<input type="checkbox"/> Realign columns

**Current cell contents**

```
<div elonodename="INPUT" style="left: 0px; top: 0px;"><input size="20" name="IX_GRP_FOO" title="" type="text"></div>
```

Drückt der Benutzer *ALT+a*, springt der Fokus zu dem entsprechenden Feld.

