

Historie

Version	Bearbeitet von	Datum	Beschreibung
1.0	A. Lange	26.11.2020	Erste Version
1.1	A. Lange	29.11.2020	Daten als Liste anzeigen zugefügt
1.2	A. Lange	16.03.2021	%%WEEKDAY%%, %%WEEKDAYPLUSONE%%, %%WEEKDAYPLUSTWO%%, %%WEEKDAYPLUSTHREE%%, %%WEEKDAYPLUSFOUR%%, %%WEEKDAYPLUSFIVE%%, %%WEEKDAYPLUSSIX%% in den Makros zugefügt
1.3	A. Lange	17.06.2021	Neuer Template-Befehl PARAM, LOCATION, CHECKDATA zugefügt
2.0	A. Lange	08.04.2023	Überarbeitung komplettes Dokument Aktualisierungen zur neuen Version
2.1	A. Lange	21.04.2023	Neue Kommandos OFFSET und LAYER
2.2	A. Lange	29.04.2023	Raster im Template-Editor
2.3	A. Lange	05.05.2023	CEILING und FLOOR in Formatierung von Datenfeldern
2.5	A. Lange	17.03.2024	Fehlerkorrekturen ATEXT erweitert um BGCOLOR und BGDISTANCE DATAMATRIX zugefügt PARAM erweitert um GROUP, LOCATIONINCLUSION, LOCATIONEXCLUSION TEXT erweitert um BGCOLOR
2.6	A. Lange	29.11.2024	CALC für die Berechnung von Zahlenwerten zugefügt
2.7	A. Lange	03.09.2025	SET (Unterdatensätze) zugefügt

Die Beschreibung bezieht sich auf die Versionen:

ESLEasy_Manager ab Version 1.5.2.1

ESLEasy_Client ab Version 1.3.2.0

ESLEasy_Preview ab Version 1.1.3.0

ESLEasy_Server ab Version 1.4.1.0

Inhaltsverzeichnis

Historie	1
Allgemeines	5
Template-Editor.....	5
Quellcode-Editor	6
Einrücken von Quellcode.....	6
Suchen und Ersetzen	6
Einsetzen von Sonderzeichen	6
Quellcodeschrift vergrößern/verkleinern	6
Quellcode-Interpreter	6
Grafischer Editor.....	8
Raster.....	8
Markieren	8
Verschieben	9
Verändern über die Anfasser	9
Rechte Maustaste.....	9
Template-Objekte.....	9
Steuerung des grafischen Editors.....	10
Objekte sperren (<i>Lock</i>)	11
Objekte ausrichten (<i>Kontextmenü</i>)	11
Tag-Wert ersetzen.....	11
Object löschen.....	12
Objekt-Inspektor	12
Daten	13
Daten verknüpfen.....	13
Daten-Karteireiter	13
Daten formatieren.....	14
Datenfelder in den Template-Quellcode einfügen.....	15
Daten abrufen	15
Daten als Liste anzeigen und auswählen.....	16
Mehrere DataID's auf einem Label.....	16
Location Tags.....	18

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

Dynamische Tag-Namen.....	20
Projektdatei	20
Template speichern / kopieren	21
ESL-Label-Daten.....	21
Aktuelle Ansicht an ein Label senden.....	22
Neues Template erstellen (<i>Config</i>).....	22
Rückgängig- Wiederholen	23
WHEN- und MAKRO-Dialog	23
Hilfe	25
Vorschau.....	25
Anzeige aktualisieren	26
Befehlsreferenz	26
CANVAS.....	26
BACKGROUNDCOLOR	26
QUALITY	27
TEXTQUALITY	27
LINKEDDATAIDS	28
OFFSET	28
LAYER.....	28
TEXT	29
FONT	31
IMAGE.....	32
RECTANGLE.....	33
LINE.....	35
ELLIPSE.....	36
POLYGON	37
PIE	38
ARC	39
BARGRAPH.....	41
BARCODE	43
DATAMATRIX	45
QRCODE	47
QRCODE_VCARD.....	47

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

PRICE.....	49
NFC (<i>Hanshow</i>)	51
NFC (<i>Ontime</i>)	52
LOCATION	53
CHECKDATA	53
PARAM.....	54
ATEXT (<i>Attribute Text</i>)	55
COCKPITCHART	57
END	59
Dynamische Ausrichtung von Objekten	59
Aufbau der dynamischen Befehle	60
Variablen	61
SETVARIABLE	61
CLEARVARIABLE	61
CLEARALLVARIABLE	61
Variable abfragen (<i>Beispiel</i>).....	61
Makros.....	62
Makrofelder.....	63
DATAVALUE	64
SPLITREPEAT	64
REGEX	65
CALC.....	65
PAGE	65
WHEN	66
Kombination von WHEN-Bedingungen	66
Liste der WHEN-Befehle	67
IF	71
Datenfelder.....	72
Formatierung von Datenfeldern.....	72

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

Allgemeines

Im ersten Teil dieses Dokuments wird die Erstellung von Templates für ESL-Easy über den Template-Editor beschrieben. Der Template-Editor ist Bestandteil des ESL-Easy-Managers.

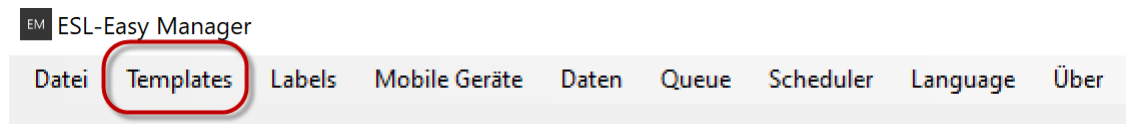
Templates können auch extern mit einem Texteditor erstellt werden. Es handelt sich um Textdateien, die den Sourcecode für die Generierung der grafischen Inhalte für die Labels enthält. Es wird aber empfohlen, den Template-Editor für die Erstellung zu verwenden.

Alle Templates müssen im konfigurierten Ordner für ESL-Easy abgelegt werden. Im Standard ist dies der Ordner **\ESLEasy\Templates**.

Im zweiten Teil folgt die Beschreibung der Befehle, die in einem Template verwendet werden können.

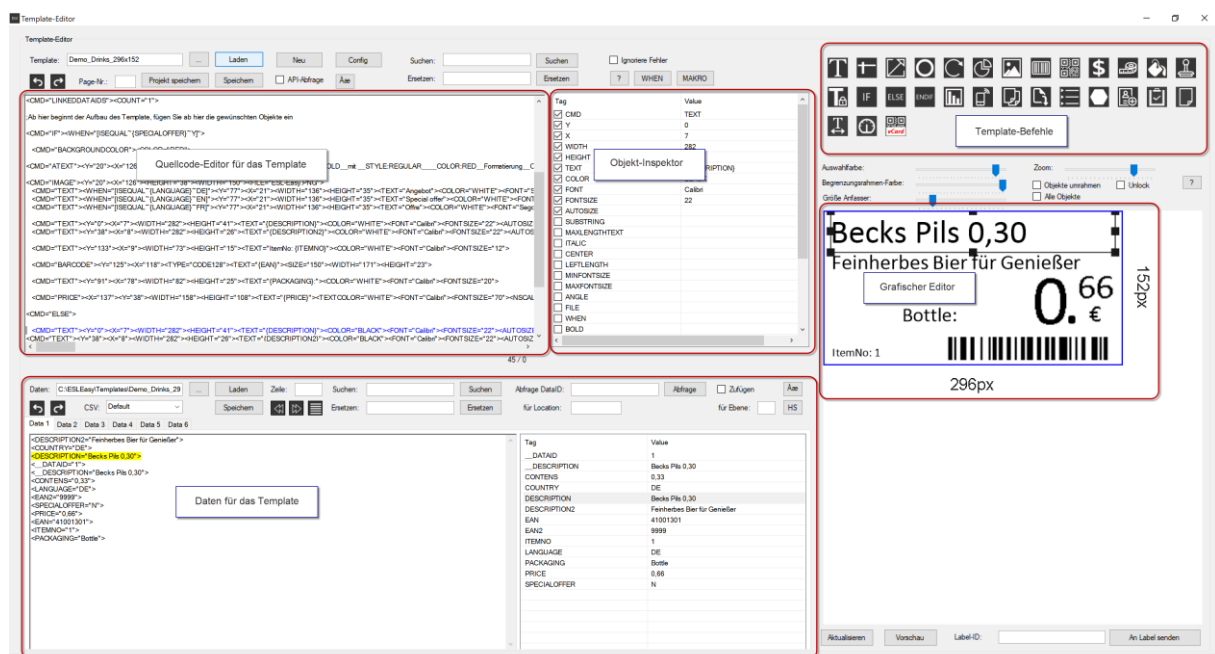
Template-Editor

Sie gelangen in den Template-Editor über das Hauptmenü des ESL-Easy-Managers:



Der Template-Editor ermöglicht Ihnen, Templates komfortabel zu erstellen und mit Nutzdaten das Ergebnis direkt angezeigt zu bekommen.

Der Template-Editor teilt sich auf verschiedene Bereiche auf:



Quellcode-Editor

Ein Template kann direkt im Quellcode-Editor erstellt werden. Alle Eingaben werden als Ergebnis im grafischen Editor angezeigt.

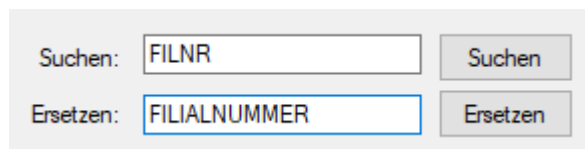
Jede Änderung an einem Template über den Quellcode wird erst mit einer kurzen Verzögerung (*wenn Sie keine Taste drücken*) grafisch ausgegeben, um das System während der Eingaben nicht unnötig zu belasten.

Einrücken von Quellcode

Im Quellcode-Editor kann markierter Text mit der **TAB**-Taste nach rechts eingerückt werden. Mit **Shift-TAB** kann der Text wieder zurück verschoben werden.

Suchen und Ersetzen

Sie können den Quelltext oder die Nutzdaten durchsuchen bzw. durch einen anderen Text ersetzen.

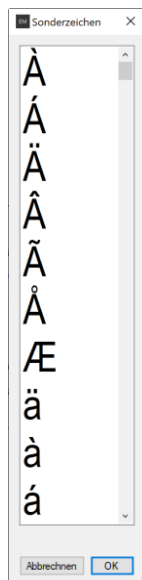


Suchen:

Ersetzen:

Einsetzen von Sonderzeichen

Über den Button  kann eine Liste mit Sonderzeichen eingeblendet werden, die dann in den Quellcode bzw. in die Daten übernommen werden können.



Quellcodeschrift vergrößern/verkleinern

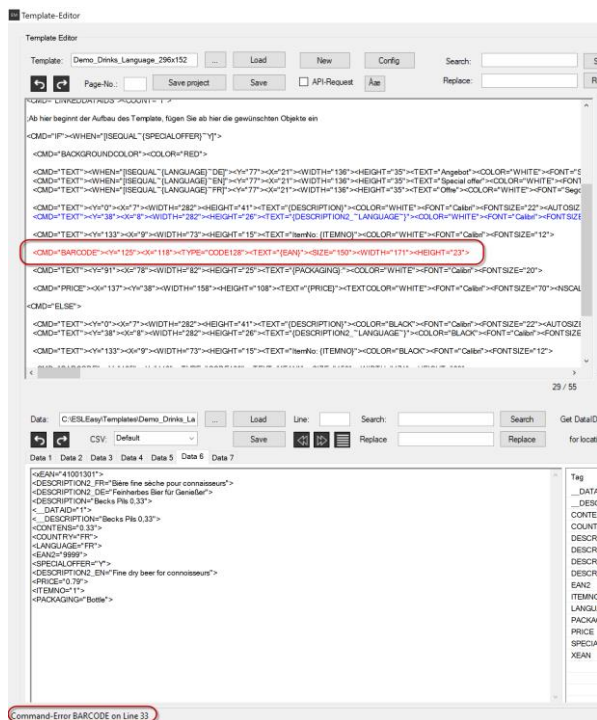
Sie können mit **Strg** + dem **Mausrad** die Schrift im Quellcode-Editor und im Datenbereich vergrößern und verkleinern.

Quellcode-Interpreter

Tritt bei der Verarbeitung des Templates ein Fehler auf, wird die fehlerhafte Zeile rot markiert und in der Status-Leiste eine Fehlermeldung angezeigt.

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025



Bitte beachten Sie: Es können auch Probleme bei der Verarbeitung von Quellcodezeilen auftreten, die sich durch inhaltliche Interpretation von Daten ergeben und keine genaue Fehleranalyse des Interpreters erlauben. Werden Objekte nicht dargestellt, prüfen Sie bitte die Eigenschaften des Objektes. Besonders **IF**-Bedingungen und **WHEN**-Angaben stellen eine große Fehlerquelle dar.

```
<CMD="BARCODE"><Y="125"><X="118"><TYPE="CODE128"><TEXT="{EAN}"><SIZE="150"><WIDTH="171"><HEIGHT="23">
```

Command-Error BARCODE on Line 33

Hier gibt es das verknüpfte Datenfeld **{EAN}** in den Nutzdaten nicht (*es fehlt <EAN="...">*)

```
<DESCRIPTION2_FR="Bière fine sèche pour connaisseurs">
<DESCRIPTION2_DE="Feinherbes Bier für Genießer">
<DESCRIPTION="Becks Pils 0,33">
<__DATAID="1">
<__DESCRIPTION="Becks Pils 0,33">
<CONTENS="0.33">
<COUNTRY="FR">
<LANGUAGE="FR">
<EAN2="9999">
<SPECIALOFFER="Y">
<DESCRIPTION2_EN="Fine dry beer for connoisseurs">
<PRICE="0.79">
<ITEMNO="1">
<PACKAGING="Bottle">
```

Mit ☐ Ignoriere Fehler kann die Ausführung des Templates bei einem Fehler fortgeführt werden. Für den späteren Echt-Einsatz müssen alle Fehler in einem Template beseitigt sein (*diese Funktion ist nur für die Entwicklung gedacht*).

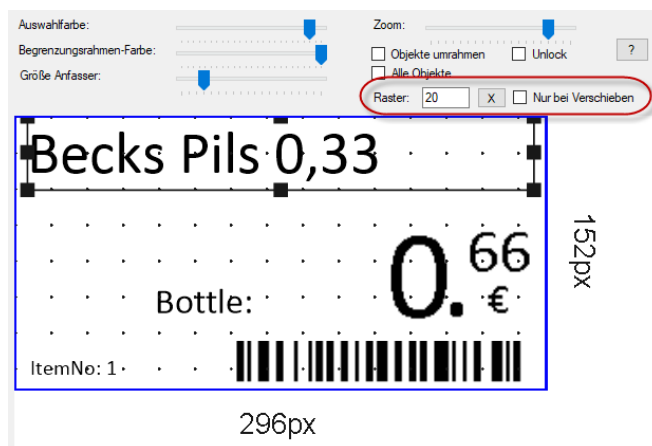
Grafischer Editor

Die angezeigten Objekte im grafischen Editor können dort direkt mit der Maus verändert werden, um z.B. Position und Größe anzupassen.

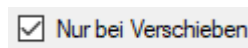


Raster

Um Objekte einfacher zu positionieren, kann ein Raster eingeblendet werden.



Geben Sie dazu den Raster-Abstand in Pixel ein (*einen Wert größer 1*) und drücken Sie Enter. Das Raster wird durch Punkte im Label-Bereich angezeigt. Sie können nun die Objekte nur noch auf dem Raster verschieben.



Ist „**Nur bei Verschieben**“ ausgewählt, können die Objekte frei in der Größe geändert werden. Nur beim Verschieben von Objekten wird das Raster als Grundlage genommen. Ist „**Nur bei Verschieben**“ nicht ausgewählt, werden auch die Größenänderungen von Objekten am Raster durchgeführt.

Wenn Sie als Rasterwert **0** oder **1** eingeben, oder den **X** Button betätigen, wird das Raster gelöscht. Danach können die Objekte wieder frei positioniert und verändert werden.

Markieren

Strg: Gedrückt halten zum Markieren mehrerer Objekte

Strg + Shift: Ein Objekt aus mehreren markierten Objekten ausschließen

Alt: Solange **Alt** gedrückt bleibt, wird das aktuell markierte Objekte angesprochen, auch wenn die Maus das Objekt selbst nicht trifft

Shift: Mehrere übereinander liegende Objekte werden nacheinander einzeln selektiert

Ein Objekt kann durch Anklicken mit der Maus oder durch Positionieren des Cursors im Quellcode-Editor auf einem Objekt-Befehl markiert werden. Lässt sich ein Objekt schwer durch die Maus markieren, ist es einfacher, die passende Quellcode-Zeile auszuwählen, um das gewünschte Objekt zu markieren.

Verschieben

Strg: Die markierten Objekte können nur horizontal verschoben werden

Strg + Shift: Die markierten Objekte können nur vertikal verschoben werden

Verändern über die Anfasser

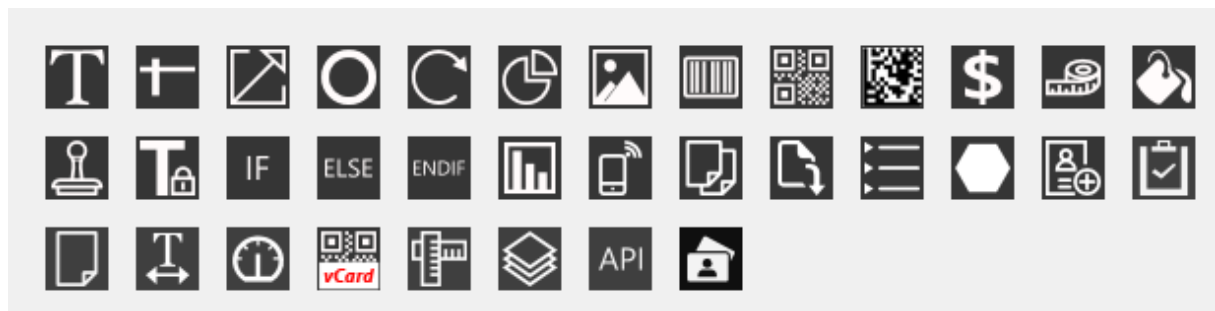
Bei Linien kann die **Strg**-Taste zusätzlich gedrückt werden. Dann wird die Linie automatisch horizontal oder vertikal ausgerichtet, je nachdem, ob sich die Maus gerade näher horizontal oder vertikal an der Linie befindet.

Rechte Maustaste

Ruft das Kontextmenü auf, über das z.B. die Objekte ausgerichtet werden können (*siehe Objekte ausrichten*).

Template-Objekte

Alle verfügbaren Objekte für die Template-Gestaltung können über die **Icons** ausgewählt werden.



Wenn Sie die Maus über einem Icon platzieren, wird Ihnen eine Erklärung zum Icon angezeigt.

Das gewählte Objekt wird als neue Quellcode-Zeile an der aktuellen Cursor-Position (*der Cursor befindet sich im Quellcode-Bereich nach den Template-Parametern*)

```
;Ab hier beginnt der Aufbau des Template, fügen Sie ab hier die gewünschten Objekte ein
```

```
<CMD="IF"><WHEN="[ISEQUAL~{SPECIALOFFER}~Y]">
```

```
<CMD="BACKGROUNDCOLOR"><COLOR="RED">
```

bzw. am Ende des Template-Quellcodes eingefügt und die möglichen Parameter im Objekt-Inspektor angezeigt. Jedes Objekt bekommt nach der Auswahl Default-Werte zugewiesen, damit das Objekt sofort im grafischen Editor erscheint. Sie können danach die Eigenschaften des Objekts nach Ihren Bedürfnissen anpassen.

Tag	Value
☒ CMD	TEXT
☒ Y	0
☒ X	0
☒ WIDTH	50
☒ HEIGHT	25
☒ TEXT	Text
☒ COLOR	BLACK
☒ FONT	Calibri
☒ FONTSIZE	22
☒ BOLD	FALSE
☐ AUTOSIZE	
☐ SUBSTRING	
☐ MAXLENGTHTEXT	
☐ ITALIC	
☐ CENTER	
☐ LEFTLENGTH	
☐ MINFONTSIZE	
☐ MAXFONTSIZE	
☐ ANGEL	
☐ WHEN	
☐ POSITION	
☐ LTRIM	
☐ MAXLENGTH	
☐ WHOLEWORD	

Neue Objekte werden immer an der zuletzt aktiven Mausposition eingefügt, wenn dies möglich ist. Ansonsten werden sie an die Default-Position gesetzt.

Die Eigenschaften der Objekte werden in der **Befehlsreferenz** beschrieben.

Steuerung des grafischen Editors

Im grafischen Editor lassen sich verschiedene Einstellungen für die Ausgabe vornehmen:

Mit den Schieberegler „**Auswahlfarbe**“ kann die Farben für Umrandungen von Objekten angepasst werden (z.B. sinnvoll, wenn die Hintergrundfarbe gleich der Umrandungsfarbe ist wie z.B. schwarz).

Über „**Zoom**“ vergrößern oder verkleinern Sie die Anzeige.

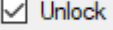
Mit „**Objekte umrahmen**“ können Sie die Objekte mit den physischen Parametern der Begrenzungen anzeigen lassen, was den vorgegebenen Platz für ein Objekt und Überlappungen zu anderen Objekten aufzeigt.

„**Alle Objekte**“ zeigt alle im Quelltext vorhandenen Objekte an, unabhängig davon, ob das Objekt aktuell angezeigt werden würde (z.B. durch eine **WHEN-Bedingung** aktuell ausgeschlossen).

Die Anfasser der Objekte können mit „**Größe Anfasser**“ in der Größe verändert werden.

Objekte sperren (Lock)

Jedes grafische Objekt hat eine Eigenschaft **<LOCK>**, die im Objektinspektor oder direkt im Sourcecode gesetzt werden kann. Damit wird dieses Objekt im grafischen Editor gesperrt, so dass es dort nicht mehr ausgewählt und verändert werden kann. Um diese Sperre wieder aufzuheben, muss das Objekt im Quellcode-Editor (*Zeile*) ausgewählt werden, um das **<LOCK>** wieder zu entfernen.

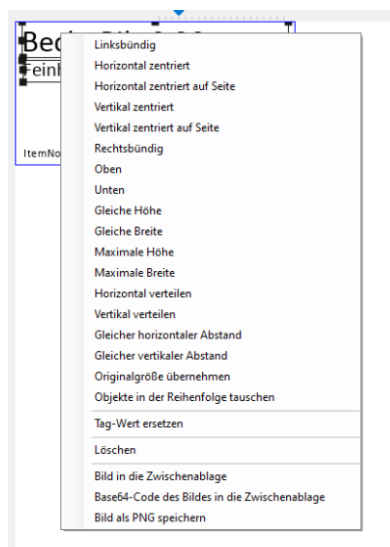
Wird  ausgewählt, hebt dies die Sperren aller Objekte temporär auf, so dass wieder alle Objekte selektiert werden können. Die Sperren der einzelnen Objekte bleiben erhalten, so dass diese wieder nach dem Deaktivieren von „**Unlock**“ aktiv sind.

Objekte ausrichten (Kontextmenü)

Sie können Objekte vom System ausrichten lassen:

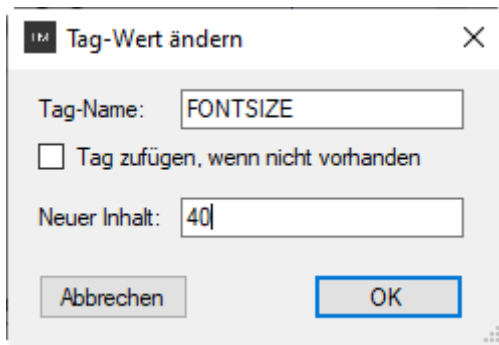
- Ist nur ein Objekt markiert, wird dieses immer in Bezug zu den **Label-Rändern** ausgerichtet
- Sind mehrere Objekte markiert, erfolgt die Ausrichtung am **zuletzt** markierten Objekt

Den Ausrichtungs-Dialog erhalten Sie durch Drücken der **rechten** Maustaste über dem grafischen Editor:



Tag-Wert ersetzen

Soll für alle markierten Objekte ein bestimmtes Tag gleichgesetzt werden (z.B. *gleiche Schriftgröße*), so kann dies über den Dialog „**Tag-Wert ersetzen**“ erfolgen:



Geben Sie den Tag-Namen ein, der in allen Objekten gesetzt werden soll (z.B. **FONTSIZE**) und den gewünschten Wert.

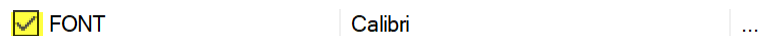
„**Tag hinzufügen, wenn nicht vorhanden**“ fügt das Tag in allen Objekten zu, wenn es bisher nicht vorhanden war. Ohne diese Auswahl wird der Wert in den Objekten nur ersetzt, wenn das Tag bereits im Objekt vorhanden ist.

Object löschen

Mit „**Löschen**“ werden die ausgewählten Objekte aus dem Template entfernt.

Objekt-Inspektor

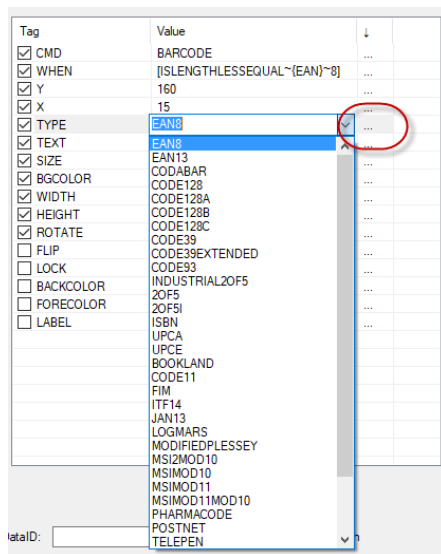
Im Objekt-Inspektor werden alle verfügbaren Eigenschaften des aktuellen Objektes angezeigt. Nur markierte Eigenschaften sind aktuell aktiv:



Einzelne Werte eines Objektes können über den Objekt-Inspektor geändert werden (*auch direkt im Quellcode möglich*).

Tag	Value		
☒ CMD	TEXT	...	
☒ Y	113	...	
☒ X	366	...	
☒ TEXT	{@@PREIS0.00}	...	
☒ WIDTH	270	...	
☒ HEIGHT	90	...	
☒ COLOR	BLACK	...	
☒ FONT	Calibri	...	
☒ FONTSIZE	85	...	
☒ RIGHT		...	
☐ AUTOSIZE		...	
☐ SUBSTRING		...	
☐ MAXLENGTHTEXT		...	
☐ ITALIC		...	
☐ CENTER		...	
☐ LEFTLENGTH		...	
☐ MINFONTSIZE		...	
☐ MAXFONTSIZE		...	
☐ ANGEL		...	
☐ WHEN		...	
☐ BOLD		...	
☐ POSITION		...	
☐ LTRIM		...	
☐ ...		...	

Eigenschaft eines Objektes, die eine Auswahl erlauben, können über „...“ abgerufen werden:



Daten

Der Daten-Bereich ermöglicht es, Testdaten für die Verknüpfung im Template zu erzeugen. Es können aktuelle Daten aus den importierten Datensätzen abgefragt oder manuell Daten erzeugt werden. Um ein Template mit allen möglichen Datenkombinationen testen zu können, können mehrere Datenbereiche (*Tags*) angelegt werden, die unterschiedliche Testdaten enthalten können. So können Sie z.B. Daten erzeugen, die lange und kurze Texte enthalten, um zu prüfen, ob der Text richtig dargestellt wird. Auch lassen sich Daten mit unterschiedlichem Inhalt erzeugen, um z.B. die Darstellung für ein Angebot zu testen (**WHEN-Bedingung**). Die erzeugten Testdaten können zu einem Template als Projektdatei gespeichert werden, um diese Daten jederzeit später wieder zur Verfügung zu haben (*gleicher Dateiname mit der Endung .TPL*). Die angelegten Testdaten beeinflussen die echten Daten in der Datenbank nicht (*es werden keine Stammdatenänderungen vorgenommen*).

Daten verknüpfen

Im Template werden viele Ausgaben mit Daten verknüpft, die später durch die ESL-Easy-Produktdatenbank oder aus externen Quellen kommen. Um das Template bereits bei der Erstellung mit realistischen Daten testen zu können, können Daten für die Verknüpfung bereitgestellt werden.

Testdaten werden immer in der Form `<Feldname="Inhalt">` erzeugt, z.B. `<PRICE="12,99">`.

Im Template kann auf den Feldnamen mit `{Feldname}` zugegriffen werden.

Beispiel:

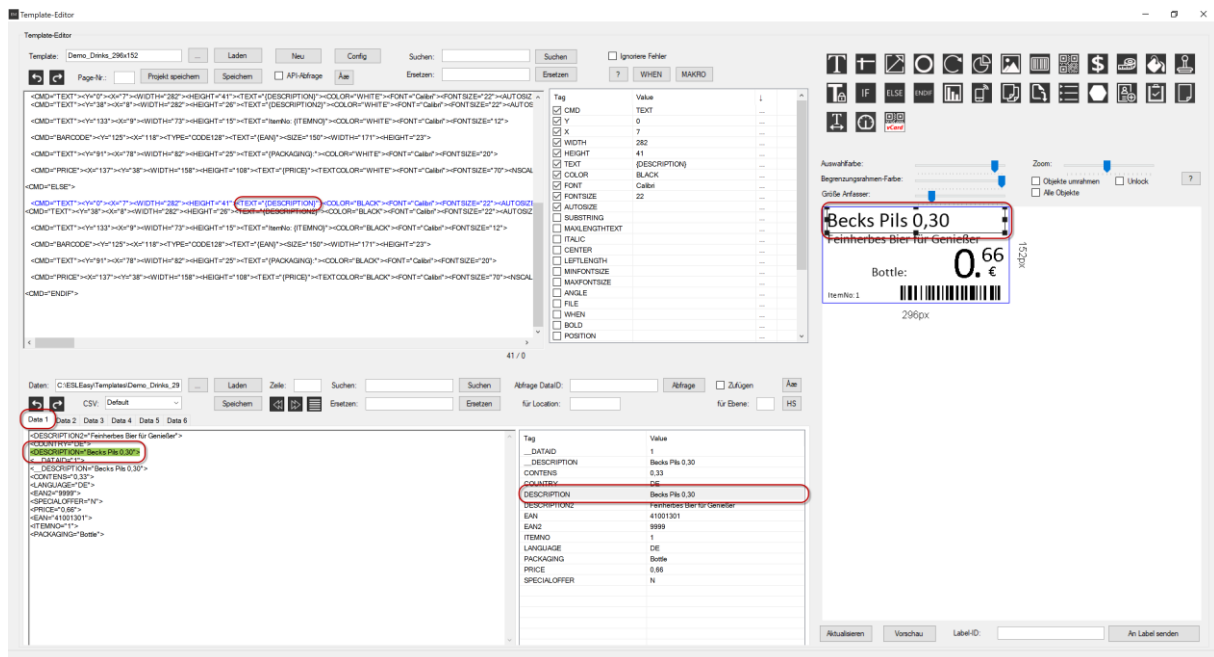
```
<CMD="TEXT"><Y="133"><X="9"><WIDTH="73"><HEIGHT="15"><TEXT="ItemNo:
{ITEMNO}"><COLOR="BLACK"><FONT="Calibri"><FONTSIZE="12">
```

Daten-Karteireiter

Um das Template mit verschiedenen Daten testen zu können, können mehrere Karteireiter geöffnet werden, die jeweils eigene Daten enthalten. Der aktivierte Karteireiter wird dann zu den aktuellen Daten, die mit dem Template verknüpft werden.

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025



Um einen neuen Karteireiter anzulegen, betätigen Sie die **rechte** Maustaste im Datenbereich:

Neuer Daten-Tab
Tab schließen
Alle Tabs schließen
Daten formatieren
Tag in die Zwischenablage kopieren
Value in die Zwischenablage kopieren

und wählen „**Neuer Daten-Tab**“

Daten formatieren

Um die Datensätze untereinander anzuordnen (*formatieren*), wählen Sie im Kontextmenü (*rechte Maustaste*) „**Daten formatieren**“ aus. Jedes Datenfeld wird dadurch in einer eigenen Zeile dargestellt.



Die Daten werden zusätzlich in einer Liste neben den Quelldaten dargestellt. Wenn Sie dort einen Datensatz auswählen, wird der dafür zugrunde liegende Teil der Quelldaten markiert:

<FARBE>	Name	Inhalt
<PLU>	_DATAID	140819-1-0-0
<BASISMENGE="100">	DESCRIPTION	BiFi Roll - STÜCK
<PROSTARTIKELNAME="BiFi Roll">	ALKOHOL	
<CURRENTPAGE="1">	ALKOHOLSTRING	
<PASST_ZU>	ANZAHEINZELTEILE	1
<LASTROWNPAGE="1">	ANZAHLUMVERPACKUNGEN	0
<QUERYTIME="2020-11-06T09:13:23.9814899+00:00">	ARTIKELBEZEICHNUNG	BiFi Roll
<STAMMWÜRZE>	ARTIKELBEZEICHNUNGERWEITERT	
<PROSTPRODUKTNAME="Roll">	AUSBAU_KURZ	
<FIRSTROWNPAGE="1">	AUSPRAEGUNG	45g
<_DESCRIPTION="BiFi Roll - STÜCK">	BASISMENGE	100
<AUSPRAEGUNG="45g">	BASISMENGE-EINHEIT	g
<HERKUNFT_KURZ>	BASISPREIS	100g = 2,64
<BRUTTOLADENVERKAUFPREIS="1,19">	BESONDERHEITEN	
<ALKOHOL>	BRUTTOLADENVERKAUFPREIS	1,19
<PAGESIZE="1">	BRUTTOPREIS	1,19
<ANZAHEINZELTEILE="1">	BRUTTOSONDERPREIS	0
<BRUTTOPREIS="1,19">	CURRENTPAGE	1
<EAN13="4251097410289">	DESCRIPTION	BiFi Roll
<PAGESIZE="1000">		
<GESCHMACK>		
<PFANDGESAMT="0">		
<WAHRUNGSSYMBOL="€">		
<REBSORTE>		
<ANZAHLUMVERPACKUNGEN="0">		
<STORER_PREISVORTEIL>		

Datenfelder in den Template-Quellcode einfügen

Sie können im Quellcode für das Template einen Bereich markieren (z.B. den Inhalt eines Text-Tags), oder den Cursor an eine Position setzen, an der das Datenfeld eingefügt werden soll:

<TEXT="Text">

Wenn Sie nun einen **Doppelklick** auf ein Datenfeld in der Liste durchführen, wird der markierte Bereich im Template-Quellcode durch das Datenfeld `{Datenfeldname}` ersetzt bzw. das Datenfeld an der Cursor-Position eingefügt:

COUNTRY	DE
DESCRIPTION	Becks Pils 0,30
DESCRIPTION2	Feinherbes Bier für Genießer

Ergebnis: <TEXT="{DESCRIPTION}">

Daten abrufen

Sie können Daten auch direkt aus der **ESL-Easy-Produktdatenbank**, aus **EXCEL** oder **CSV**-Dateien oder einem angebundenen System abrufen:

Abfrage DataID:	105860	Abfrage	☐ Zufügen
für Location:	10343	für Ebene:	

ESL-Easy ruft die Daten entsprechend ab und gibt sie im aktuellen Tab aus.

Für **CSV**-Dateien kann die Kodierung der Datei beim Einlesen vorgegeben werden:

Default	▼
Default	
ASCII	
UTF8	
UTF7	
UTF32	
Unicode	
BigEndianUnicode	

Bei **EXCEL** oder **CSV**-Dateien sind meist mehrere Datensätze enthalten. Um auf die einzelnen Datenzeilen zugreifen zu können, können Sie die Zeilennummer eingeben oder durch die Datenzeilen blättern.

Zeile:





Bitte beachten Sie: Um **EXCEL**-Dateien einlesen zu können, wird ein lokal installiertes **Microsoft EXCEL** benötigt!

Daten als Liste anzeigen und auswählen

Zeile:





Sie können sich die zu importierenden Daten in einer Auswahlliste anzeigen lassen.

Ansicht Daten

	ItemNo	StoreNo	Description	Description2	Contents	Packaging	Price	EAN	EAN2	SpecialOffer
▶	1	1	Becks Pils 0,33	Feinherbes Bier für Genießer	0,33	Bottle	0,79	41001301	9999	N
	1	2	Becks Pils 0,33	Feinherbes Bier für Genießer	0,33	Bottle	0,79	41001301	9999	N
	1	3	Becks Pils 0,33	Feinherbes Bier für Genießer	0,33	Bottle	0,79	41001301	9999	N
	2	1	Cachaca Pitu 0,70	Perfekt für Caipirinha	0,75	Bottle	9,89	4000269003428		N
	2	2	Cachaca Pitu 0,70	Perfekt für Caipirinha	0,75	Bottle	8,99	4000269003428		N
	2	3	Cachaca Pitu 0,70	Perfekt für Caipirinha	0,75	Bottle	7,99	4000269003428		J
	3	1	Jägermeister 0,7	Der Likör für Kenner mit dem Hirsch	0,7	Bottle	11,39	4067700011015	6666	N
	3	2	Jägermeister 0,7	Der Likör für Kenner mit dem Hirsch	0,7	Bottle	10,99	4067700011015	6666	J
	3	3	Jägermeister 0,7	Der Likör für Kenner mit dem Hirsch	0,7	Bottle	11,39	4067700011015	6666	N

Abbrechen OK

Mehrere DataID's auf einem Label

Es ist möglich, dass **mehrere DataID's** innerhalb eines Templates benötigt werden, wenn z.B. mehrere Artikel auf einem Label ausgegeben werden sollen. Da in diesem Fall mehrere DataID's abgerufen werden müssen, und die zurückgegebenen Datenfelder für jede Data-ID gleich heißen, können die Daten in Ebenen aufgeteilt werden.

Abfrage DataID: Abfrage ☒ Zufügen

für Location: für Ebene:

In diesem Fall werden die Daten aus der aktuellen Abfrage in die Nutzdaten als **2. Ebene** hinzugefügt. In den Daten wird dies durch ein an das Datenfeld angehangene „_2“ erkenntlich:

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

```
<VPACKBEZK_2="Kist"><EINZELPREISNORMAL_2="0"><ARTBEZ_2="Augustiner Hell 20x0,5"><BASISEINHEIT_2="1"><BASISPREIS="2.00">
<BASIS="20"><PREIS="17.990000"><LEERGARTNR_2="990310"><AE="0"><EWMW="M"><BASISEINHEIT_2="1"><EINZELPREIS_2="0.950000">
<_DATAID="105860-0_20"><EAN_2="4105250022003"><_DATAID_2="105760-0_20"><PREISNORMAL="0"><EAN="41057544"><BASIS_2="20">
<_DESCRIPTION="Schweiger Schmäckerl Weiße 20x0,5"><INHVKEBEZK_2="I"><_DESCRIPTION_2="Augustiner Hell 20x0,5">
<LEERGARTNR="990310"><VKEEINZ_2="1"><ARTNR="105860"><AE_2="0"><FILNR_2="343"><ENDTIME"><INHVKE_2="0.5000">
<TS_VONDATUM="10.03.2020 00:00:00"><BASISEINHEIT="1"><PREIS_2="16.990000"><VPACKBEZLEINZ="Flasche"><VKE_2="20"><KZVL="V">
<KZBASISART="2"><LIEFNR"><PREISNORMAL_2="0"><KZSONDERVK="0"><BASISPREIS_2="1.90"><AEE="1"><ARTNR_2="105760"><AEE_2
="1"><WGR_2="10"><EINZELPREIS="1.000000"><LEERGPREIS_2="0.080000"><VPACKBEZLEINZ_2="Flasche"><VKEEINZ="1">
<INHVKEBEZK="I"><ARTBEZ="Schweiger Schmäckerl Weiße 0,5"><ENDTIME_2"><BASISPREIS="1.80"><EWMW_2="M"><BASISEINHEIT="1">
<KZVL_2="V"><VKE="20"><FILNR="343"><KZBASISART_2="2"><TS_BISDATUM"><TS_BISDATUM_2"><VPACKBEZ="Kiste"><LIEFNR_2">
<ARTBEZ="Schweiger Schmäckerl Weiße 20x0,5"><VPACKBEZK="Kist"><ARTBEZ_2="Augustiner Hell 0,5"><WGR="10"><TS_VONDATUM_2
="10.01.2020 00:00:00"><BASISPREIS_2="1.70"><INHVKE="0.5000"><LEERGPREIS="3.100000"><KZSONDERVK_2="0"><LEERGPREIS_2
="3.100000"><EINZELPREISNORMAL="0"><VPACKBEZL_2="Kiste"><LEERGPREIS="0.080000">
```

Damit kann im Template auf den Preis der zweiten Ebene mit **{PREIS_2}** zugegriffen werden.

Im Template können Sie über den Namen des Datenfeldes mit angehangener Ebene (z.B. **ITEMNO_2**) auf den Inhalt zugreifen, oder alternativ den Befehl **LAYER** verwenden (siehe **LAYER Kommando**).

SET (Unterdatensätze)

Sind in den Daten zu einer **DATAID** Informationen zu Unterdatensätzen (z.B. *Set-Artikel*) angegeben (**<_SETx_DATAID="">**), werden alle Datenfelder für die SET-DATAID's zusätzlich bereitgestellt. Allen Datenfeldern wird **_SETx_** vorangestellt (z.B. **<_SET1_DESCRIPTION="">**).

Im Template-Editor kann angegeben werden, ob die Unterdatensätze für die Template-Entwicklung zusätzlich geladen werden sollen:

Im Template kann dann auf diese Datenfelder wie auf jedes andere Datenfeld zugegriffen werden.

Wenn die Nummerierung der SET-Unterdaten nicht bekannt ist (z.B. *wurden die Set-Unterdatensätze von einem externen ERP-System nicht in aufsteigender Reihenfolge beginnend mit 1, sondern z.B. mit 1,3,5,7 angegeben*), kann im Template auf alle Unterdatensätze mit

```
<_SET#x_Datenfeld>
```

zugegriffen werden. Das **X** hinter der **Raute** gibt dann den Unterdatensatz in sortierter Reihenfolge an, auf den zugegriffen werden soll.

Beispiel:

Vorhandene Unterdatenfelder:

```
<_SET1_DESCRIPTION="abc">
<_SET3_DESCRIPTION="abc">
<_SET5_DESCRIPTION="abc">
```

Um nun auf den **zweiten** Unterdatensatz zuzugreifen, kann entweder **{_SET3_DESCRIPTION}** oder **{_SET#2_DESCRIPTION}** verwendet werden, da der Unterdatensatz **<_SET3_DESCRIPTION="abc">** in der Reihenfolge der **zweite** Unterdatensatz ist.

Location Tags

Im Location-Dialog (*Standorte*) können Daten pro Location (*Standort*) zusätzliche Datenfelder hinterlegt werden, die immer automatisch zur Verfügung stehen (z.B. *wenn ein Label in einer Location verlinkt wird*). Mit diesen Daten kann in der gleichen Weise gearbeitet werden, wie mit Stammdaten. Die Tags werden den Stammdaten-Tags zugefügt, als würden sie Teil der Stammdaten sein.

Standorte					
	Standort	Interne Nr.	Bezeichnung	Subnetz	DataObject
▶	1	1	Demo Germany	192.168.8.	<LANGUAGE="DE">
	2	2	Demo England		<LANGUAGE="EN">
	3	3	Demo French		<LANGUAGE="FR">
*					

Mit einem Doppelklick auf **DataObject** wird der Pflege-Dialog für Location-Tags aufgerufen:

[illegible]

Beispiel für den Einsatz von Location-Tags:Abfrage der Data-ID **1** aus der Location **1**:

Abfrage DataID: ☐ Zufügen

für Location: für Ebene:

Tag	Value
__DATAID	200001
__DESCRIPTION	XS: 1,20 m/10 mm
ARTIKELNR	200001
FARBE	schwarz
GROESSE	XS: 1,20 m/10 mm
LANGUAGE	DE
UVP	7,99

Abfrage der Data-ID **1** aus der Location **2**:

Abfrage DataID: ☐ Zufügen

für Location: für Ebene:

Tag	Value
__DATAID	200001
__DESCRIPTION	XS: 1,20 m/10 mm
ARTIKELNR	200001
FARBE	schwarz
GROESSE	XS: 1,20 m/10 mm
LANGUAGE	EN
UVP	7,99

Dynamische Tag-Namen

Tag-Namen für die Ausgabe auf den Labels bzw. für **WHEN**-Bedingungen können dynamisch zusammengebaut werden, um z.B. den Inhalt von Location-Tags mit in den Tag-Namen einzubeziehen.

Der Zugriff auf den Inhalt eines Tags wird mit **+Tag-Name+** gekennzeichnet.

Beispiel:

```
<WHEN=" [ ISNOTEMPTY~ENERGY_CLASS_2017_+LANGUAGE+ ] ">
```

Die WHEN-Bedingung erfolgt auf ein dynamisch zusammengebauten Tag-Namen aus dem statischen Teil **ENERGY_CLASS_2017_** und dem Inhalt des Location-Tags **LANGUAGE**. Steht in **LANGUAGE = „DE“**, ist der Tag-Name **ENERGY_CLASS_2017_DE**. Es wird entsprechend der Inhalt des Tags **ENERGY_CLASS_2017_DE** für die Ausgabe auf dem Label verwendet.

Projektdatei

Es ist möglich, die abgerufenen bzw. manuell erstellten Daten zu einem Template, zu speichern. Damit stehen diese Daten bei der nächsten Bearbeitung des Templates wieder zur Verfügung, ohne diese erneut erstellen zu müssen. Geben Sie dazu unter **Daten:** als Dateiname den Namen des Templates mit der Endung **.prj** ein und drücken Sie **„Speichern“**.

Template:

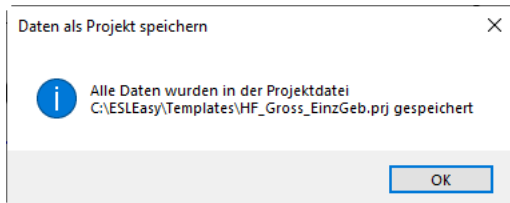
Page-Nr.:

```
<CMD="TEXT"><WHENAND="([ISNOTFLOATEQUAL~([EINZELPREIS]~0))([ISNOTFLOATEQUAL~([EINZELPREIS]~0])><Y="2"
<CMD="TEXT"><Y="210"><X="366"><TEXT="{@EINZELPREIS0.00}"><W
<CMD="TEXT"><WHEN="([ISNOTFLOATEQUAL~([PREIS]~0])><Y="155"><X
<CMD="TEXT"><WHENAND="([ISNOTFLOATEQUAL~([PREIS]~0))([ISNOTFLO
<CMD="TEXT"><Y="113"><X="366"><TEXT="{@PREIS0.00}"><WIDTH=
<CMD="IF"><WHEN="([ISNOTFLOATEQUAL~([EAN]~0])>
<CMD="BARCODE"><WHEN="([ISLENGTHGREATER~([EAN]~8])><Y
<CMD="BARCODE"><WHEN="([ISLENGTHLESSEQUAL~([EAN]~8])><Y="11
<CMD="ENDIF">
<CMD="TEXT"><Y="355"><X="459"><TEXT="{ARTNR}"><WIDTH="173"><I
<CMD="TEXT"><WHEN="([ISEQUAL~([EWMW]~M])><Y="298"><X="80"><TE
<CMD="TEXT"><WHEN="([ISEQUAL~([EWMW]~E])><Y="298"><X="80"><TE
<CMD="IMAGE"><WHEN="([ISEQUAL~([PFEL]~OBEN])><Y="319"><X="6"><
<CMD="IMAGE"><WHEN="([ISEQUAL~([PFEL]~LINKS])><Y="319"><X="6"><
<CMD="IMAGE"><WHEN="([ISEQUAL~([PFEL]~RECHTS])><Y="319"><X="6"><
<CMD="IMAGE"><WHEN="([ISEQUAL~([PFEL]~UNTEN])><Y="319"><X="6">
<CMD="ELSE">
<CMD="BACKGROUNDCOLOR"><COLOR="RED">
<CMD="TEXT"><WHEN="([ISNOTEMPTY~ARTBEZ])><Y="6"><X="8"><TEX
<CMD="TEXT"><WHEN="([ISEMPTY~ARTBEZ])><Y="6"><X="8"><TEXT="U
```

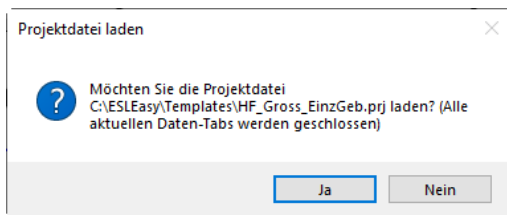
Daten:

CSV:

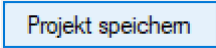
Es wird eine Projektdatei für das Template angelegt.



Beim nächsten Laden des Templates werden Sie gefragt, ob die Projektdatei geladen werden soll.

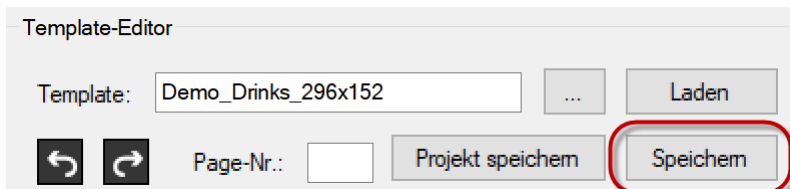


Bei „JA“ werden die Daten aller Dateireiter geladen.

Mit dem Button  wird automatisch eine Projektdatei (**.TPL**) zum aktuellen Template-Namen angelegt.

Template speichern / kopieren

Sie sollten die Änderungen im Template in regelmäßigen Abständen speichern.



Das aktuelle Template wird immer unter dem angegebenen Namen abgelegt. Um ein Template zu kopieren, können Sie es einfach unter einem anderen Namen speichern.

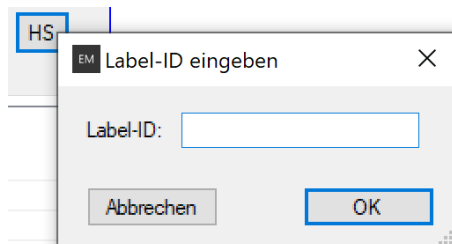
ESL-Label-Daten

ESL-Labels stellen je nach Hersteller unterschiedliche Parameter wie Batteriestand, Funkqualität, Seriennummer, Größe, Auflösung, usw. zur Verfügung. Auf die Daten des aktuellen (*verlinkten*) ESL-Labels kann im Template zugegriffen werden, gleich den normalen Daten.

ESL-Label-Parameter beginnen immer mit **__HS_Parametername**. Bei **Ontime**-Labels kann z.B. mit dem Parameter **__HS_BATTERY** auf den Batteriestand in Prozent zugegriffen werden. Damit wäre es z.B. möglich, ein „**Low battery**“ Symbol zusätzlich auf dem Label auszugeben, wenn der Batteriestand unter 5 % gesunken ist:



Eine Liste der verfügbaren Label-Parameter kann über den Button  abgerufen werden.



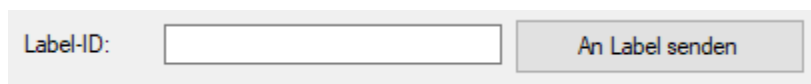
Nach der Eingabe einer Label-ID werden die verfügbaren Parameter in die Liste der Daten eingefügt.

Tag	Value
__HS_APSIGNAL	46
__HS_BARCODE	800208852
__HS_BATTERY	100
__HS_BATTERYLEVEL	5
__HS_CHIPSOLUTION	2
__HS_CURRENTSYSTEMTIME	1641384954
__HS_CURRENTTIME	12536042
__HS_DESCRIPTION	ZKC26V-N
__HS_DEVICETYPE	1
__HS_HEIGHT	152
__HS_LASTCOMMUTIME	2022-01-05 13:15:54
__HS_MAC	A0A3B8BD9AD0
__HS_MODEL	ZKC26V-N
__HS_PICNUM	7
__HS_SOFTVERSION	2.2.6
__HS_STATE	1

Die Bedeutung der Werte ist Herstellerabhängig.

Aktuelle Ansicht an ein Label senden

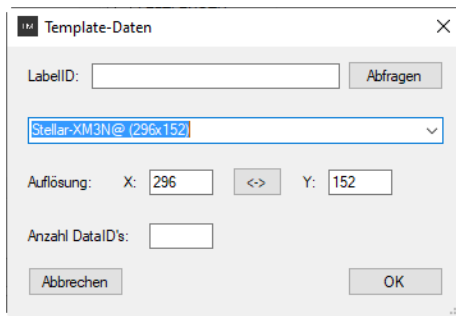
Um die Darstellung auf einem Label betrachten zu können, ist es möglich, die aktuelle Ansicht direkt zu einem Label zu senden.



Bitte beachten Sie, dass nur die aktuelle Ansicht zum Label gesendet wird, evtl. Pages werden hier nicht berücksichtigt (*siehe Befehlsreferenz*). Sollten **NFC**-Daten im Template vorhanden sein, werden diese mit an das Label übertragen.

Neues Template erstellen (Config)

Wenn Sie ein neues Template anlegen, benötigt ESL-Easy die Information von Ihnen, für welchen Label-Typ bzw. für welche Auflösung das Template erstellt werden soll. Dazu erscheint ein Dialog:



Sie können einen Label-Typ aus der Liste auswählen, ein Label scannen oder eine Label-ID eingeben. In diesem Fall wird die Auflösung zum Label anhand des gescannten Labels ermittelt und vorgegeben.

Templates werden immer für eine Auflösung (*Pixel horizontal / Pixel vertikal*) erzeugt. Wird später ein Label gescannt, werden nur Templates zur Auswahl angeboten, die zur Auflösung des Labels passen (*Ausnahme ist die Übersteuerung dieser Vorgabe in der App*).

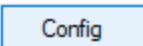
Wenn Sie Daten (*DataID's*) mit dem Template verknüpfen, können Sie die Anzahl der für dieses Template benötigten Data-ID's vorgeben. Diese werden im Befehl

```
.Connected DataID's
<CMD="LINKEDDATAIDS"><COUNT="1">
```

eingetragen (*siehe Befehlsreferenz*).

Wenn Sie ein Template erstellen, welches mit einer unterschiedlichen Anzahl von DataID's arbeitet (so könnte sich z.B. je nach Anzahl der gescannten Artikel das Layout dynamisch anpassen), können Sie mehrere Angaben mit Komma trennen. Beim Verlinken des Labels wird dann geprüft, ob die Anzahl der erfassten Artikel mit einer der Angaben in **LINKEDDATAIDS** übereinstimmt:

<CMD="LINKEDDATAIDS"><COUNT="1, 2, 3, 4"> -> Es müssen **1, 2, 3** oder **4 DataID's** für dieses Template erfasst werden.

Der Dialog lässt sich jederzeit nachträglich über  nochmals aufrufen, um nachträglich Änderungen am Template vorzunehmen (*die Werte können aber auch direkt im Quellcode angepasst werden*).

Rückgängig- Wiederholen

Sie können Änderungen rückgängig machen, oder diese wieder herstellen über die Buttons:



WHEN- und MAKRO-Dialog

WHEN-Bedingungen und **MAKROS** können jeweils über einen Dialog in das Objekt eingefügt werden:



ESL-Easy Template-Editor und Befehlsreferenz

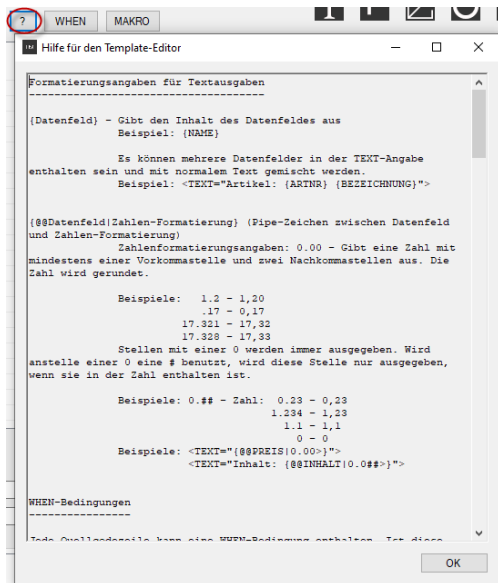
Datum: 03.09.2025

When-Bedingung auswählen		
Bedingung	Beschreibung	Parameter
[ISTAG"Parameter1"]	Abfrage, ob das angegebene Datenfeld in den verbundenen Daten vorhanden ist.	Parameter 1: Name des Datenfeldes
[ISNOTTAG"Parameter1"]	Abfrage, ob das angegebene Datenfeld in den verbundenen Daten nicht vorhanden ist.	Parameter 1: Name des Datenfeldes
[ISEQUAL"Parameter1"Parameter2]	Abfrage, ob Parameter 1 und Parameter 2 gleich sind (String-Vergleich).	String(Parameter 1) = String(Parameter 2)
[ISNOTEQUAL"Parameter1"Parameter2]	Abfrage, ob Parameter 1 und Parameter 2 nicht gleich sind (String-Vergleich).	String(Parameter 1) <> String(Parameter 2)
[ISINTEQUAL"Parameter1"Parameter2]	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob beide gleich sind.	Int(Parameter 1) = Int(Parameter 2)
[ISNOTINTEQUAL"Parameter1"Parameter2]	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob beide ungleich sind.	Int(Parameter 1) <> Int(Parameter 2)
[ISFLOATEQUAL"Parameter1"Parameter2]	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob beide gleich sind.	Float(Parameter 1) = Float(Parameter 2)
[ISNOTFLOATEQUAL"Parameter1"Parameter2]	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob beide ungleich sind.	Float(Parameter 1) <> Float(Parameter 2)
[ISLESSTHAN"Parameter1"Parameter2]	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob Parameter 1 kleiner Parameter 2 ist.	Int(Parameter 1) < Int(Parameter 2)
[ISFLOATLESSTHAN"Parameter1"Parameter2]	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob Parameter 1 kleiner Parameter 2 ist.	Int(Parameter 1) < Int(Parameter 2)
[ISLESSEQUAL"Parameter1"Parameter2]	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob Parameter 1 kleiner gleich Parameter 2 ist.	Int(Parameter 1) <= Int(Parameter 2)
[ISFLOATLESSEQUAL"Parameter1"Parameter2]	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob Parameter 1 kleiner gleich Parameter 2 ist.	Int(Parameter 1) <= Int(Parameter 2)
[ISGREATERTHAN"Parameter1"Parameter2]	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob Parameter 1 größer Parameter 2 ist.	Int(Parameter 1) > Int(Parameter 2)
[ISFLOATGREATERTHAN"Parameter1"Parameter2]	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob Parameter 1 größer Parameter 2 ist.	Int(Parameter 1) > Int(Parameter 2)
[ISGREATEREQUAL"Parameter1"Parameter2]	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob Parameter 1 größer gleich Parameter 2 ist.	Int(Parameter 1) >= Int(Parameter 2)
[ISFLOATGREATEREQUAL"Parameter1"Parameter2]	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob Parameter 1 größer gleich Parameter 2 ist.	Int(Parameter 1) >= Int(Parameter 2)
[ISLENGTHGREATERTHAN"Parameter1"Parameter2]	Es wird geprüft, ob die String-Länge von Parameter 1 (Anzahl Zeichen) größer der Angabe in Parameter 2 ist.	String(Parameter 1).Length > Int(Parameter 2)
[ISLENGTHLESSEQUAL"Parameter1"Parameter2]	Es wird geprüft, ob die String-Länge von Parameter 1 (Anzahl Zeichen) größer gleich der Angabe in Parameter 2 ist.	String(Parameter 1).Length >= Int(Parameter 2)
[ISLENGTHLESSTHAN"Parameter1"Parameter2]	Es wird geprüft, ob die String-Länge von Parameter 1 (Anzahl Zeichen) kleiner der Angabe in Parameter 2 ist.	String(Parameter 1).Length < Int(Parameter 2)
[STARTSWITH"Parameter1"Parameter2]	Es wird geprüft, ob Parameter 1 mit der Angabe in Parameter 2 beginnt.	
[NOTSTARTSWITH"Parameter1"Parameter2]	Es wird geprüft, ob Parameter 1 nicht mit der Angabe in Parameter 2 beginnt.	
[ENDSWITH"Parameter1"Parameter2]	Es wird geprüft, ob Parameter 1 mit der Angabe in Parameter 2 endet.	
[NOTENDSWITH"Parameter1"Parameter2]	Es wird geprüft, ob Parameter 1 nicht mit der Angabe in Parameter 2 endet.	
[CONTAINS"Parameter1"Parameter2]	Es wird geprüft, ob Parameter 2 in Parameter 1 vorkommt.	
[CONTAINSNOT"Parameter1"Parameter2]	Es wird geprüft, ob Parameter 2 nicht in Parameter 1 vorkommt.	
[ISEMPTY"Parameter1"]	Es wird geprüft, ob das angegebene Datenfeld leer ist.	Parameter 1: Name des Datenfeldes
[ISNOTEMPTY"Parameter1"]	Es wird geprüft, ob das angegebene Datenfeld nicht leer ist.	Parameter 1: Name des Datenfeldes
[ISVALUEEMPTY"Parameter1"]	Es wird geprüft, ob der String leer ist.	String(Parameter 1) == leer
[ISNOTVALUEEMPTY"Parameter1"]	Es wird geprüft, ob der String nicht leer ist.	String(Parameter 1) != leer
[ISFILE"Parameter1"]	Es wird geprüft, ob das in Parameter 1 angegebene File vorhanden ist.	Parameter 1: Dateiname
[ISNOTFILE"Parameter1"]	Es wird geprüft, ob das in Parameter 1 angegebene File nicht vorhanden ist.	Parameter 1: Dateiname
[ISDATEEQUAL"Parameter1"Parameter2]	Es wird geprüft, ob das in Parameter 1 angegebene Datum gleich dem Datum in Parameter2 ist.	(DateTime)Parameter1 = (DateTime)Parameter2
[ISDATENOTEQUAL"Parameter1"Parameter2]	Es wird geprüft, ob das in Parameter 1 angegebene Datum ungleich dem Datum in Parameter2 ist.	(DateTime)Parameter1 <> (DateTime)Parameter2
[ISDATEGREATERTHAN"Parameter1"Parameter2]	Es wird geprüft, ob das in Parameter 1 angegebene Datum größer dem Datum in Parameter2 ist.	(DateTime)Parameter1 > (DateTime)Parameter2
[ISDATELESSTHAN"Parameter1"Parameter2]	Es wird geprüft, ob das in Parameter 1 angegebene Datum kleiner dem Datum in Parameter2 ist.	(DateTime)Parameter1 < (DateTime)Parameter2

Makro auswählen	
Makro	Beschreibung
%%TODAY%%	Aktuelles Datum im Format ddMMyyyy (04.01.2020)
%%TODAYMINUSONE%%	Aktuelles Datum - 1 Tag im Format dd.MM.yyyy (03.01.2020) (Abhängig der Culture des PC)
%%TODAYMINUSTWO%%	Aktuelles Datum - 2 Tage im Format dd.MM.yyyy (02.01.2020) (Abhängig der Culture des PC)
%%TODAYMINUSTHREE%%	Aktuelles Datum - 3 Tage im Format dd.MM.yyyy (01.01.2020) (Abhängig der Culture des PC)
%%TODAYMINUSFOUR%%	Aktuelles Datum - 4 Tage im Format dd.MM.yyyy (31.12.2019) (Abhängig der Culture des PC)
%%TODAYMINUSFIVE%%	Aktuelles Datum - 5 Tage im Format dd.MM.yyyy (30.12.2019) (Abhängig der Culture des PC)
%%TODAYPLUSONE%%	Aktuelles Datum + 1 Tag im Format dd.MM.yyyy (03.01.2020) (Abhängig der Culture des PC)
%%TODAYPLUSTWO%%	Aktuelles Datum + 2 Tage im Format dd.MM.yyyy (02.01.2020) (Abhängig der Culture des PC)
%%TODAYPLUSTHREE%%	Aktuelles Datum + 3 Tage im Format dd.MM.yyyy (01.01.2020) (Abhängig der Culture des PC)
%%TODAYPLUSFOUR%%	Aktuelles Datum + 4 Tage im Format dd.MM.yyyy (31.12.2019) (Abhängig der Culture des PC)
%%TODAYPLUSFIVE%%	Aktuelles Datum + 5 Tage im Format dd.MM.yyyy (30.12.2019) (Abhängig der Culture des PC)
%%TOMORROW%%	Morgiges Datum im Format dd.MM.yyyy (01.01.2020) (Abhängig der Culture des PC)
%%YESTERDAY%%	Gestriges Datum im Format dd.MM.yyyy (30.12.2019) (Abhängig der Culture des PC)
%%FIRSTDAYOFTHETWEEK%%	Erster Tag der aktuellen Woche im Format ddMMyyyy (30.12.2019) (Abhängig der Culture des PC)
%%WEEK%%	Kalenderwoche des aktuellen Datums
%%DAY%%	Aktueller Tag im Format dd (04)
%%MONTH%%	Aktueller Monat im Format MM (01)
%%SHORTYEAR%%	Aktuelles Jahr im Format yy (20)
%%LONGYEAR%%	Aktuelles Jahr im Format yyyy (2020)
%%YEAR%%	Aktuelles Jahr im Format yyyy (2020)
%%TIME%%	Aktuelle Uhrzeit im Format HH:mm:ss (08:55:25)
%%HOUR%%	Aktuelle Stunde im Format HH (08)
%%MINUTE%%	Aktuelle Minute im Format mm (56)
%%SECOND%%	Aktuelle Sekunde im Format ss (31)
%%IPADDRESS%%	Aktuelle IP-Adresse
%%IPADDRESSES%%	Liste aller aktuellen IP-Adressen mit Komma getrennt als ein Text
%%APPPATH%%	Aktueller Programmpfad der Applikation
%%HOSTNAME%%	Hostname des PC
%%VERSION%%	Programmversion der Applikation
%%TOOLSVERSION%%	Programmversion der Tools-Lib

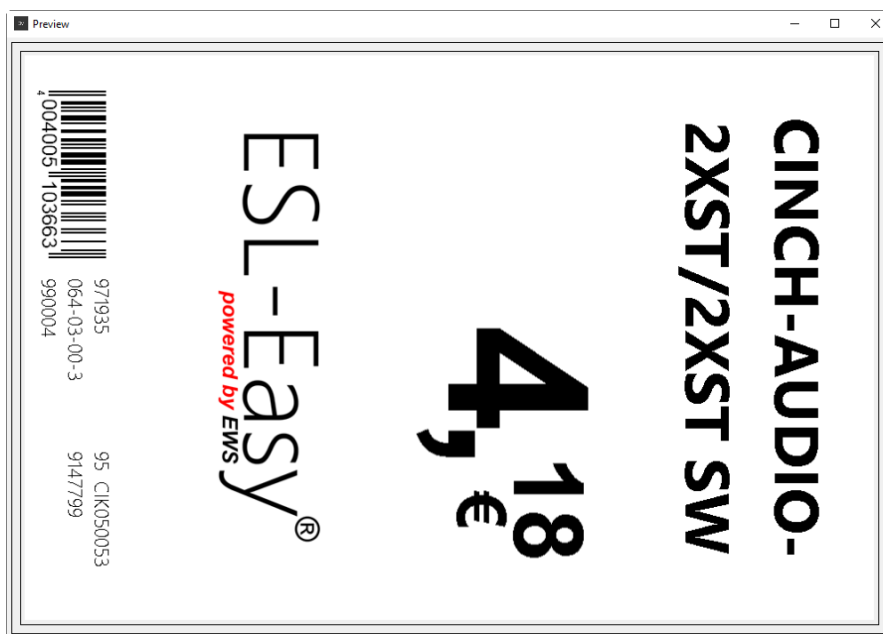
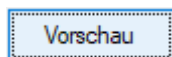
Hilfe

Über die „?“ erhalten Sie eine Kurzhilfe.



Vorschau

Templates werden immer in dem Format (*Hochkant/Quer*) bearbeitet, wie sie später auf den Labels dargestellt werden sollen. Wird ein Template für ein Label erstellt, das später ein Hochkant-Format aufweist, muss die grafische Darstellung des Labels entsprechend rotiert an das Label gesendet werden (*das Label selbst bleibt ja physisch im Quer-Format*). Um dies zu überprüfen, können Sie eine Vorschau anzeigen lassen:



Anzeige aktualisieren

Mit dem Button  kann die Anzeige des Templates aktualisiert werden, wenn dies nicht automatisch durch das Programm erfolgt.

Befehlsreferenz

Nachfolgend werden alle Befehle beschrieben, die zum Erstellen eines Templates eingesetzt werden können.

CANVAS

Gibt die Auflösung des Templates in Pixel für **Höhe** und **Breite** vor. Für elektronische Labels müssen diese Werte mit den physischen Werten der Labels übereinstimmen. Am mobilen **MDE**-Gerät werden nur passende Templates angeboten, die zur Auflösung des angesprochenen Labels passen (*Ausnahme, wenn „Alle“ explizit ausgewählt wird*). Wird ein Template z.B. für die Hochkant-Darstellung erzeugt, das Label selbst ist aber für das Querformat konstruiert, muss das Bild später gedreht an das Label gesendet werden (siehe **ROTATE**).

<i>Parameter</i>	<i>Erklärung</i>
WIDTH	Breite in Punkten
HEIGHT	Höhe in Punkten
ROTATE	Angabe in Grad , wenn die Ausgabe bei der Übertragung zum Label gedreht werden soll. Gültige Werte sind 0, 90, 180 und 270

Beispiel:

```
<CMD="CANVAS"><WIDTH="640"><HEIGHT="960"><ROTATE="90">
```

Diese Werte werden auch über den **Config**-Dialog gesetzt.

BACKGROUNDCOLOR

Gibt die Hintergrundfarbe an. Bei elektronischen Labels ist zu beachten, dass nur Hintergrundfarben zum Einsatz kommen, die vom Label unterstützt werden (wie z.B. *Rot*). Standard ist **WHITE** (*weiß*).

<i>Parameter</i>	<i>Erklärung</i>
COLOR	Breite in Punkten
HEIGHT	Höhe in Punkten
COLOR	Farbangabe als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

	YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
--	---

Beispiel:

<CMD="BACKGROUNDCOLOR"><COLOR="WHITE">

QUALITY

Gibt die Ausgabequalität an. Bei ESL-Labels muss „**ESL**“ gewählt werden, da die Labels keine Farbabstufungen unterstützen und somit das Ergebnis bei den anderen Konstanten zu unbefriedigenden Ergebnissen führen würde.

<i>Parameter</i>	<i>Erklärung</i>
MODE	DEFAULT – Default-Einstellung des Betriebssystems (<i>kann nicht empfohlen werden</i>) ANTIALIAS – Ränder werden weich gezeichnet HIGHQUALITY – Die Ausgabequalität wird auf hoch gesetzt. Dadurch wird mehr Zeit für die Berechnung der grafischen Ausgabe benötigt HIGHSPEED – Die Ausgabequalität wird auf niedrig gesetzt. Dadurch wird weniger Zeit für die Berechnung der grafischen Ausgabe benötigt NONE – Es wird keine Qualitätsanpassung vorgenommen ESL – Die Einstellung für ESL-Labels . Da diese Labels ein ePaper-Display haben, welches keine Farbabstufungen unterstützt, erfolgt die Berechnung auf Pixelebene ohne Kantenglättung

Beispiel:

<CMD="QUALITY"><MODE="ESL">

TEXTQUALITY

Gibt die Ausgabequalität für Text an. Bei ESL-Labels muss „**ESL**“ gewählt werden, da die Labels keine Farbabstufungen unterstützen und somit das Ergebnis bei den anderen Konstanten zu unbefriedigenden Ergebnissen führen würde.

<i>Parameter</i>	<i>Erklärung</i>
MODE	DEFAULT – Default-Einstellung des Betriebssystems (<i>kann nicht empfohlen werden</i>) ANTIALIAS – Ränder werden weich gezeichnet ANTIALIASGRIDFIT – Ränder werden weich gezeichnet anhand der Pixelvorgaben CLEARTYPEGRIDFIT – Ausgabe als ClearType-Schrift mit Kantenglättung SINGLEBITPERPIXEL – Spezieller Berechnungsmodus SINGLEBITPERPIXEGRIDFITL – Spezieller Berechnungsmodus anhand der Pixelvorgaben ESL – Die Einstellung für ESL-Labels . Da diese Labels ein ePaper-Display

	haben, welches keine Farbabstufungen unterstützt, erfolgt die Berechnung auf Pixelebene ohne Kantenglättung
--	---

Beispiel:

```
<CMD="TEXTQUALITY"><MODE="ESL">
```

LINKEDDATAIDS

Gibt die Anzahl der in diesem Template verknüpften DataID's an (z.B. *Artikel*). Dieser Wert wird bei einer Verknüpfung von einem Label mit einer oder mehrerer DataID's geprüft (*wurden bei einem Template mit 2 DataID's auch 2 DataID's für die Verknüpfung abgerufen?*). Wird diese Angabe weggelassen, erfolgt keine Prüfung bei der Verknüpfung von einer DataID mit einem Label (z.B. *auch, wenn in einem Template keine DataID verwendet wird*). Sind unterschiedlich viele DataID's in einem Template erlaubt, so können mehrere Werte mit Komma getrennt angegeben werden.

Parameter	Erklärung
COUNT	Anzahl der für dieses Template benötigten DataID's (z.B. <i>Artikel</i>). Es können mehrere Werte mit Komma getrennt angegeben werden

Beispiele:

```
<CMD="LINKEDDATAIDS"><COUNT="1">
```

```
<CMD="LINKEDDATAIDS"><COUNT="1, 2, 3, 4">
```

OFFSET

Dieser Befehl verschiebt den Nullpunkt für alle folgenden Objekte um die angegebenen **X/Y**-Werte. Die folgenden Objekte werden damit an die angegebene Position (**XY-Angabe im Kommando**) plus der **OFFSET**-Angabe gezeichnet.

Dies ermöglicht es, extrem einfach mehrere Blöcke von Objekten auf einem Label darzustellen (z.B. *mehrere Artikel*), ohne für jedes Objekt in jedem Block die **X/Y**-Koordinaten anpassen zu müssen. Die Quellcodezeilen können einfach kopiert werden. Der zusätzliche **OFFSET**-Befehl vor den kopierten Zeilen verschiebt dann die Koordinaten für die Darstellung.

Parameter	Erklärung
X	Angabe in Pixel, um welchen Wert jeder X-Wert der folgenden Befehle verschoben wird
Y	Angabe in Pixel, um welchen Wert jeder Y-Wert der folgenden Befehle verschoben wird

Beispiele:

```
<CMD="OFFSET"><X="250">
```

```
<CMD="OFFSET"><Y="150">
```

```
<CMD="OFFSET"><Y="-75">
```

```
<CMD="OFFSET"><X="250"><Y="150">
```

LAYER

Mit diesem Befehl wird die Ebene aller folgender Datenfelder gesetzt, ohne dies in jedem Objekt anpassen zu müssen (*es muss nicht mehr **_X** bei einem Datenfeld angegeben werden*).

Werden Quellcodezeilen kopiert, kann für die neuen Zeilen auf eine andere Datenfeldebene (z.B. *auf den zweiten Artikel*) zugegriffen werden, ohne dies in den kopierten Quellcodezeilen anpassen zu müssen.

<i>Parameter</i>	<i>Erklärung</i>
LAVEL	Angabe der Zugriffsebene für die folgenden Datenfelder (<i>es wird _X an den Datenfeldnamen angehängen</i>)

Beispiele:

<CMD="LAYER"><LEVEL="2"> -> Die Angabe {DESCRIPTION} wird zu {DESCRIPTION_2}

<CMD="LAYER"><LEVEL="1"> -> Es wird wieder auf die Angabe im Quellcode zurückgegriffen (*Reset*)

TEXT

Ausgabe von Text. Es handelt sich um ein sehr komplexes Objekt, welches viele Formatierungen erlaubt.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Maximale Breite des Textes (<i>Begrenzungsrahmen</i>)
HEIGHT	Maximale Höhe des Textes (<i>Begrenzungsrahmen</i>)
COLOR	Farbe des Textes als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
BGCOLOR	Hintergrundfarbe als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

	YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
TEXT	Auszugebener Text Ein ~ Zeichen wird in einen Zeilenumbruch übersetzt Innerhalb des Textes kann auf Datenfelder der verknüpften Daten mit {Datenfeld} zugegriffen werden (alternativ mit [DATAVALUE~Datenfeld])
FILE	Der Text wird aus der angegebenen Datei geladen
LTRIM	Führende Leerzeichen werden abgeschnitten
MAXLENGTH	Maximale Anzahl der auszugebenen Zeichen. Ist der Text länger, wird er abgeschnitten und „...“ bzw. MAXLENGHTTEXT angehängen
MAXLENGHTTEXT	Der Text, der bei der Überschreitung der maximalen Textlänge (MAXLENGTH) an den Text angehängen wird (Default „...“)
LEFTLENGTH LEFTTEXT	Es wird nur die angegebene Anzahl von Buchstaben von links vom Text ausgegeben
RIGHTLENGTH	Es wird nur die angegebene Anzahl von Buchstaben von rechts vom Text ausgegeben
SUBSTRING	Es wird nur ein Teil des Textes ausgegeben. Es muss die Startposition und die Anzahl der Zeichen ab der Startposition mit Komma getrennt angegeben werden, z.B. < SUBSTRING =“3,5”> für die Ausgabe von fünf Zeichen beginnend nach der dritten Position
ANGEL	Drehwinkel des Textes
FONT	Name des Fonts (installierte Schriftarten)
FONTSIZE	Font Größe des Textes
BOLD	Fett
ITALIC	Kursiv
REGULAR	Standard
STRIKEOUT	Durchgestrichen
UNDERLINE	Unterstrichen
DATEFORMAT	Angabe des zu verwendenden Datums-Formats. Ohne diese Angabe wird das Default-Datumsformat benutzt (siehe Makros)

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

TIMEFORMAT	Angabe des zu verwendenden Uhrzeit-Formats. Ohne diese Angabe wird das Default-Zeitformat benutzt (<i>siehe Makros</i>)
CENTER	Zentriert den Text im angegebenen Bereich horizontal mittig
RIGHT	Positioniert den Text rechtsbündig im angegebenen Bereich
WORDWRAP	Führt einen automatischen Umbruch der Worte im angegebenen Bereich durch
AUTOSIZE	Passt die Schriftgröße automatisch so an, dass der Text in den angegebenen Bereich passt (<i>siehe MINFONTSIZE, MAXFONTSIZE</i>)
MINFONTSIZE	Wenn AUTOSIZE gesetzt ist, wird die Schriftgröße nicht kleiner als bis zur angegebenen Font-Größe skaliert. Text, der nicht mehr in den vorgegebenen Rahmen passt, wird abgeschnitten
MAXFONTSIZE	Wenn AUTOSIZE gesetzt ist, wird die Schriftgröße nicht größer als bis zur angegebenen Font Größe skaliert
POSITION	Positioniert den Text innerhalb des angegebenen Rahmens aus X,Y-Sicht UPPERLEFT UPPERCENTER UPPERRIGHT CENTERLEFT CENTERCENTER CENTERRIGHT LOWERLEFT LOWERCENTER LOWERRIGHT

Beispiel:

```
<CMD="TEXT"><Y="36"><X="0"><WIDTH="640"><HEIGHT="100"><TEXT="{ARTIKE  
LBEZEICHNUNG 1}"><COLOR="BLACK"><FONT="Segoe  
UI"><FONTSIZE="40"><BOLD="FALSE"><CENTER>
```

FONT

Erzeugt einen Font, der für weitere Befehle genutzt werden kann. Dieser ist nur so lange gültig, bis eine andere Font-Angabe diesen überschreibt. Wird normalerweise nicht benötigt, da der Font bei jedem Befehl mit angegeben wird.

<i>Parameter</i>	<i>Erklärung</i>
NAME	Name des Fonts (<i>installierte Schriftarten</i>)
BOLD	Fett
ITALIC	Kursiv
REGULAR	Standard

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

STRIKEOUT	Durchgestrichen
UNDERLINE	Unterstrichen

Beispiel:

<CMD="FONT"><NAME="Arial"><BOLD>

IMAGEGibt eine Grafik aus. Es werden Dateien im Format **PNG, JPG, BMP, GIF, TIF, TIFF** und **WMF** unterstützt.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite der Grafik
HEIGHT	Höhe der Grafik
WIDTHMAX	Setzt die Breite der Grafik auf die Label-Größe
HEIGHTMAX	Setzt die Höhe der Grafik auf die Label-Größe
BASE64	Image im BASE64-Format
FILE	Image aus einer Datei
NOTTRANSPARENT	Stellt die Grafik nicht transparent dar (z.B. bei einer PNG-Datei)
GREYSCALE	Wandelt eine Grafik in s/w um
ROTATE	Rotiert eine Grafik im angegebenen Winkel
STRETCH	Skaliert die Grafik auf den angegebenen Bereich
CENTER	Zentriert die Grafik innerhalb des angegebenen Bereichs
TOP	Gibt die Grafik am oberen Rand des angegebenen Bereichs aus
BOTTOM	Gibt die Grafik am unteren Rand des angegebenen Bereichs aus
RIGHT	Gibt die Grafik am rechten Rand des angegebenen Bereichs aus
LEFT	Gibt die Grafik am linken Rand des angegebenen Bereichs aus
INVERT	Invertiert eine Grafik
INVERTBW	Invertiert nur den schwarz/weiß Anteil in einer Grafik und lässt die farblichen Anteile (wie z.B. Rot) unverändert

INVERTBWTOLERANCE

Gibt einen Wert zwischen 0 und 255 vor, der abweichend von reinem schwarz (0) und reinem weiß (255) noch als schwarz bzw. weiß interpretiert wird

Beispiel:

```
<CMD="IMAGE"><Y="215"><X="170"><HEIGHT="38"><WIDTH="150"><FILE="ESL-Easy.PNG">
```

Für jede Grafikdatei kann eine Formatierungsdatei mit dem gleichen Namen der Grafikdatei und der Endung „.fmt“ angelegt werden. Ist eine solche Datei vorhanden, werden folgende Formatierungsangaben aus der **FMT**-Datei gelesen und überschreiben die Angaben im Template.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite der Grafik
HEIGHT	Höhe der Grafik
WIDTHMAX	Setzt die Breite der Grafik auf die Label-Breite
HEIGHTMAX	Setzt die Höhe der Grafik auf die Label-Größe
STRETCH	Skaliert die Grafik auf den angegebenen Bereich
NOTTRANSPARENT	Stellt die Grafik nicht transparent dar (z.B. bei einer PNG-Datei)
CENTER	Zentriert die Grafik innerhalb des angegebenen Bereichs
TOP	Gibt die Grafik am oberen Rand des angegebenen Bereichs aus
BOTTOM	Gibt die Grafik am unteren Rand des angegebenen Bereichs aus
RIGHT	Gibt die Grafik am rechten Rand des angegebenen Bereichs aus
LEFT	Gibt die Grafik am linken Rand des angegebenen Bereichs aus

RECTANGLE

Zeichnet ein Rechteck.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite des Rechtecks

HEIGHT	Höhe des Rechtecks
COLOR	Farbe des Rahmens als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
THICKNESS	Rahmenstärke in Punkten Wird diese Angabe weggelassen, wird kein Rahmen gezeichnet.
STYLE	Die Art des Rahmens nach folgenden Konstanten: DASH – Gestrichelte Linie DASHDOT – Punkt – Strich – Linie DASHDOTDOT – Punkt – Punkt – Strich – Linie DOT – Linie aus Punkten SOLID – Durchgezogene Linie
BGCOLOR	Farbe der Füllung des Rechtecks als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0 Wird diese Angabe weggelassen, wird keine Füllung gezeichnet.
EDGE	Grad der Abrundung der Ecken (<i>sinnvolle Werte zwischen 1 und 15</i>)
ANGEL	Drehwinkel

Beispiel:

```
<CMD="RECTANGLE"><Y="127"><X="74"><WIDTH="100"><HEIGHT="100"><COLOR="BLACK"><THICKNESS="1"><STYLE="SOLID"><BGCOLOR="WHITE">
```

LINE

Zeichnet eine Linie. Linien können an den Enden verschiedene Formen, wie z.B. eine Pfeilspitze, zugewiesen bekommen (siehe **STARTCAP**, **ENDCAP**).

Parameter	Erklärung
X	Horizontale Startposition
Y	Vertikale Startposition
TOX	Horizontale Endposition
TOY	Vertikale Endposition
COLOR	Farbe des Rahmens als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
THICKNESS	Linienstärke in Punkten
STYLE	Die Art des Rahmens nach folgenden Konstanten: DASH – Gestrichelte Linie DASHDOT – Punkt – Strich – Linie DASHDOTDOT – Punkt – Punkt – Strich – Linie DOT – Linie aus Punkten SOLID – Durchgezogene Linie
STARTCAP	Legt die Art des Linienbeginns fest: ARROWANCHOR - Gibt ein pfeilförmiges Ankerende an DIAMONDANCHOR - Gibt ein rautenförmiges Ankerende an FLAT -Gibt ein abgeflachtes Linienende an NOANCHOR - Gibt an, dass kein Anker verwendet wird ROUND - Gibt ein rundes Linienende an ROUNDANCHOR - Gibt ein rundes Ankerende an SQUARE - Gibt ein quadratisches Linienende an SQUAREANCHOR - Gibt ein quadratisches Ankerlinienende an TRIANGLE - Gibt ein dreieckiges Linienende an
ENDCAP	Legt die Art des Linienbeginns fest: ARROWANCHOR - Gibt ein pfeilförmiges Ankerende an

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

	DIAMONDANCHOR - Gibt ein rautenförmiges Ankerende an FLAT -Gibt ein abgeflachtes Linienende an NOANCHOR - Gibt an, dass kein Anker verwendet wird ROUND - Gibt ein rundes Linienende an ROUNDANCHOR - Gibt ein rundes Ankerende an SQUARE - Gibt ein quadratisches Linienende an SQUAREANCHOR - Gibt ein quadratisches Ankerlinienende an TRIANGLE - Gibt ein dreieckiges Linienende an
--	--

Beispiel:

```
<CMD="LINE"><Y="230"><X="232"><TOX="50"><TOY="50"><COLOR="BLACK"><WIDTH="50"><THICKNESS="1"><STYLE="SOLID">
```

ELLIPSEZeichnet eine Ellipse bzw. einen Kreis (*bei gleicher Höhe, Breite*).

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite der Ellipse
HEIGHT	Höhe der Ellipse
COLOR	Farbe des Rahmens als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
THICKNESS	Rahmenstärke in Punkten Wird diese Angabe weggelassen, wird kein Rahmen gezeichnet.
STYLE	Die Art des Rahmens nach folgenden Konstanten: DASH – Gestrichelte Linie DASHDOT – Punkt – Strich – Linie DASHDOTDOT – Punkt – Punkt – Strich – Linie DOT – Linie aus Punkten SOLID – Durchgezogene Linie

BGCOLOR	Farbe der Füllung der Ellipse als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0 Wird diese Angabe weggelassen, wird keine Füllung gezeichnet.
ANGEL	Drehwinkel

Beispiel:

```
<CMD="ELLIPSE"><Y="260"><X="316"><WIDTH="110"><HEIGHT="67"><COLOR="BLACK"><STYLE="SOLID"><THICKNESS="2">
```

POLYGON

Zeichnet ein Polygon aus einer Anzahl von angegebenen Punkten. Die Höhe und Breite des Polygons ergibt sich aus den angegebenen Punkten, daher gibt es die Angaben **WIDTH** und **HEIGHT** nicht.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
COLOR	Farbe des Rahmens als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
THICKNESS	Rahmenstärke in Punkten Wird diese Angabe weggelassen, wird kein Rahmen gezeichnet.

STYLE	Die Art des Rahmens nach folgenden Konstanten: DASH – Gestrichelte Linie DASHDOT – Punkt – Strich – Linie DASHDOTDOT – Punkt – Punkt – Strich – Linie DOT – Linie aus Punkten SOLID – Durchgezogene Linie
BGCOLOR	Farbe der Füllung der Ellipse als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0 Wird diese Angabe weggelassen, wird keine Füllung gezeichnet.
POINTS	Angabe der Punkt-Koordinaten im Format X,Y;X,Y;... Beispiel: 25,0;75,0;100,40;75,80;25,80;0,40
ANGEL	Drehwinkel

Beispiel:

```
<CMD="POLYGON"><Y="0"><X="0"><POINTS="25,0;75,0;100,40;75,80;25,80;0,40"><COLOR="BLACK"><STYLE="SOLID"><THICKNESS="2">
```

PIE

Zeichnet ein Stück eines Tortendiagramms. Ein komplettes Tortendiagramm kann aus einer Kombination mehrerer **PIE**-Befehle gezeichnet werden.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite
HEIGHT	Höhe
COLOR	Farbe des Rahmens als Konstante: BLACK BLUE GREEN CYAN

	RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
THICKNESS	Rahmenstärke in Punkten Wird diese Angabe weggelassen, wird kein Rahmen gezeichnet.
STYLE	Die Art des Rahmens nach folgenden Konstanten: DASH – Gestrichelte Linie DASHDOT – Punkt – Strich – Linie DASHDOTDOT – Punkt – Punkt – Strich – Linie DOT – Linie aus Punkten SOLID – Durchgezogene Linie
BGCOLOR	Farbe der Füllung der Ellipse als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0 Wird diese Angabe weggelassen, wird keine Füllung gezeichnet.
STARTANGLE	Startwinkel zwischen X und der ersten Seite des Tortendiagramms im Uhrzeigersinn gemessener Winkel in Grad
SWEEPANGLE	Zwischen dem Startwinkel und der zweiten Seite des Tortendiagramms im Uhrzeigersinn gemessener Winkel in Grad

Beispiel:

```
<CMD="PIE"><Y="0"><X="0"><WIDTH="100"><HEIGHT="100"><STARTANGLE="0">
<SWEEPANGLE="120"><COLOR="BLACK"><BGCOLOR="RED">
```

ARC

Zeichnet einen Bogen, der einen Teil einer Ellipse darstellt.

<i>Parameter</i>	<i>Erklärung</i>
------------------	------------------

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite
HEIGHT	Höhe
COLOR	Farbe des Rahmens als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
THICKNESS	Rahmenstärke in Punkten Wird diese Angabe weggelassen, wird kein Rahmen gezeichnet.
STYLE	Die Art des Rahmens nach folgenden Konstanten: DASH – Gestrichelte Linie DASHDOT – Punkt – Strich – Linie DASHDOTDOT – Punkt – Punkt – Strich – Linie DOT – Linie aus Punkten SOLID – Durchgezogene Linie
BGCOLOR	Farbe der Füllung der Ellipse als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0 Wird diese Angabe weggelassen, wird keine Füllung gezeichnet.
STARTANGLE	Startwinkel zwischen X und dem Anfangspunkt des Bogens im Uhrzeigersinn gemessener Winkel in Grad

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

SWEEPANGLE

Zwischen dem **Startwinkel** und dem **Endpunkt** des Bogens im Uhrzeigersinn gemessener Winkel in **Grad**

Beispiel:

```
<CMD="ARC"><Y="0"><X="0"><WIDTH="100"><HEIGHT="100"><STARTANGLE="0">
<SWEEPANGLE="150"><COLOR="BLACK">
```

BARGRAPH

Zeichnet ein Balken eines Balkendiagramms. Die Füllung des Balkens kann in Prozent angegeben werden.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite
HEIGHT	Höhe
COLOR	Farbe des Rahmens als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
THICKNESS	Rahmenstärke in Punkten Wird diese Angabe weggelassen, wird kein Rahmen gezeichnet.
STYLE	Die Art des Rahmens nach folgenden Konstanten: DASH – Gestrichelte Linie DASHDOT – Punkt – Strich – Linie DASHDOTDOT – Punkt – Punkt – Strich – Linie DOT – Linie aus Punkten SOLID – Durchgezogene Linie
BGCOLOR	Farbe der Füllung der Ellipse als Konstante: BLACK BLUE GREEN CYAN RED

	MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0 Wird diese Angabe weggelassen, wird keine Füllung gezeichnet.
PERCENT	Prozentangabe des Füllgrades des Balkens. Damit kann z.B. ein Verlauf in Prozent eines Verarbeitungszustandes angezeigt werden.
BARPOSITION	Zeichenrichtung des Füllgrades aus PERCENT : LEFT -> Von links nach rechts UPPER -> Von oben nach unten RIGHT -> Von rechts nach links LOWER -> Von unten nach oben (<i>Default</i>)
BARSTYLE	Die Art der Füllung nach folgenden Konstanten: BACKWARDDIAGONAL CROSS DARKDOWNWARDDIAGONAL DARKHORIZONTAL DARKUPWARDDIAGONAL DARKVERTICAL DASHEDDOWNWARDDIAGONAL DASHEDHORIZONTAL DASHEDUPWARDDIAGONAL DASHEDVERTICAL DIAGONALBRICK DIAGONALCROSS DIVOT DOTTEDIAMOND DOTTEDGRID FORWARDDIAGONAL HORIZONTAL HORIZONTALBRICK LARGECHECKERBOARD LARGECONFETTI LARGEGRID LIGHTDOWNWARDDIAGONAL LIGHTHORIZONTAL LIGHTUPWARDDIAGONAL LIGHTVERTICAL MAX MIN NARROWHORIZONTAL NARROWVERTICAL

	OUTLINEDDIAMOND
	PERCENT05
	PERCENT10
	PERCENT20
	PERCENT25
	PERCENT30
	PERCENT40
	PERCENT50
	PERCENT60
	PERCENT70
	PERCENT75
	PERCENT80
	PERCENT90
	PLAID
	SHINGLE
	SMALLCHECKERBOARD
	SMALLCONFETTI
	SMALLGRID
	SOLIDDIAMOND
	SPHERE
	TRELLIS
	VERTICAL
	WAVE
	WEAVE
	WIDEDOWNWARDDIAGONAL
	WIDEUPWARDDIAGONAL
	ZIGZAG

Beispiel:

```
<CMD="BARGRAPH"><Y="10"><X="10"><WIDTH="20"><HEIGHT="100"><COLOR="BLACK"><BARCOLOR="RED"><PERCENT="20"><THICKNESS="1"><BARPOSITION="LOWER">
```

BARCODE

Zeichnet einen Barcode.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite
HEIGHT	Höhe
TEXT	Barcode-Daten

FORECOLOR	Vordergrundfarbe (<i>Barcode-Striche</i>) als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
BACKCOLOR	Hintergrundfarbe als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
TYPE	Barcode-Typ: EAN13 EAN8 CODABAR CODE128 CODE128A CODE128B CODE128C CODE39 CODE39EXTENDED CODE93 INDUSTRIAL2OF5 2OF5I ISBN 2OF5 UPCA UPCE BOOKLAND CODE11 FIM ITF14

	JAN13 LOGMARS MODIFIEDPLESSEY MSI2MOD10 MSIMOD10 MSIMOD11 MSIMOD11MOD10 PHARMACODE POSTNET TELEPEN UCC12 UCC13 UPCSUPPLEMENTAL2DIGIT UPCSUPPLEMENTAL5DIGIT USD8
ROTATE	Drehwinkel in Grad. Gültige Werte sind: 0, 90, 180, 270
FLIP	Spiegelt den Barcode: X – Horizontal Y – Vertikal XY – Horizontal und Vertikal
LABEL	Blendet den Barcode-Text unten am Barcode ein
IGNOREERRORS	Wenn angegeben, werden Fehler ignoriert. Fehler können entstehen, wenn die verlinkten Daten keinen gültigen Barcode ergeben. In diesem Fall wird der Barcode in der Darstellung auf dem Label weggelassen.
ANGEL	Drehwinkel

Beispiel:

```
<CMD="BARCODE"><Y="0"><X="0"><TYPE="EAN13"><TEXT="978020137962"><SIZE="150"><WIDTH="200"><HEIGHT="50">
```

DATAMATRIX

Zeichnet einen Data Matrix-Code.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite
HEIGHT	Höhe
TEXT	Data Matrix Daten

FORECOLOR	Vordergrundfarbe (<i>Data Matrix-Code</i>) als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
BACKCOLOR	Hintergrundfarbe als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
ANGEL	Drehwinkel
MARGINSIZE	Größe des Randes um den Data Matrix-Code
SCHEME	Data Matrix Schema nach ISO: ASCII (<i>Default</i>) ASCIGS1 AUTOBEST AUTOFAST BASE256 C40 EDIIFACT TEXT X12
MODULSIZE	Größe (<i>Qualität</i>) des Data Matrix-Code

Beispiel:

```
<CMD="DATAMATRIX"><Y="10"><X="10"><WIDTH="75"><HEIGHT="75"><TEXT="www.esl-easy.de">
```

QRCODE

Zeichnet einen QR-Code.

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite (<i>und Höhe</i>) des QR-Codes
QUITEZONES	Randbereich um den QR-Code: ZERO – Kein Randbereich (<i>Default</i>) TWO – Randgröße 1 FOUR – Randgröße 2
TEXT	Der Text, der als QR-Code dargestellt werden soll
ERRORCORRECTIONLEVEL	H, L, M, Q (<i>Fehlerkorrektur</i>)
ANGEL	Drehwinkel

Beispiel :

```
<CMD="QRCODE"><Y="0"><X="0"><WIDTH="75"><TEXT="www.esl-
easy.de"><QUITEZONES="ZERO"><WIDTH="75">
```

QRCODE_VCARD

Zeichnet einen QR-Code als **vCard** (*Visitenkarte im internationalen Standard*). Dieser QR-Code wird von Smartphones als Visitenkarte erkannt und als Kontakt im Gerät gespeichert.

Nicht benötigte Tags für die **vCard**-Information können weggelassen werden (z.B. **TEL_HOME**).

<i>Parameter</i>	<i>Erklärung</i>
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite (<i>und Höhe</i>) des QR-Codes
QUITEZONES	Randbereich um den QR-Code: ZERO – Kein Randbereich (<i>Default</i>) TWO – Randgröße 1 FOUR – Randgröße 2
TEXT	Der Text, der als QR-Code dargestellt werden soll
ERRORCORRECTIONLEVEL	H, L, M, Q (<i>Fehlerkorrektur</i>)
VCARD	TRUE Der QR-Code wird als vCard interpretiert

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

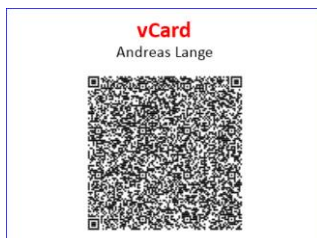
NAME	Name
FIRSTNAME	Vorname
LASTNAME	Nachname
TITLE	Titel
TEL_CEL	Mobilfunknummer
TEL_WORK	Telefonnummer Arbeit
TEL_HOME	Telefonnummer Privat
TEL_FAX	Fax-Nummer
EMAIL_HOME	eMail-Adresse Privat
EMAIL_WORK	eMail-Adresse Arbeit
URL	Homepage
STREET	Straße
CITY	Stadt
ZIP	Postleitzahl
COUNTRY	Land
ORG	Organisation
VCARD_TEMPLATE	Filename der Template-Datei, die für den QR-Code als vCard verwendet werden soll (<i>mit Pfadangabe</i>). Wird dieses Tag nicht angegeben, wird das Default-Template vCard.txt verwendet (<i>Bestandteil in ESL-Easy</i>): <u>Inhalt vCard.txt:</u> <pre> BEGIN:VCARD VERSION:2.1 FN:<NAME> N:<FIRSTNAME>;<LASTNAME> TITLE:<TITLE> TEL;CELL:<TEL_CELL> TEL;WORK;VOICE:<TEL_WORK> TEL;HOME;VOICE:<TEL_HOME> EMAIL;HOME;INTERNET:<EMAIL_HOME> EMAIL;WORK;INTERNET:<EMAIL_WORK> URL:<URL> ADR;;;<STREET>;<CITY>;;<ZIP>;<COUNTRY> ORG:<ORG> END:VCARD </pre>

ANGEL**Drehwinkel**Beispiel:

```
<CMD="QRCODE_VCARD"><Y="71"><X="88"><WIDTH="327"><QUITEZONES="ZERO">
<ERRORCORRECTIONLEVEL="H"><VCARD><NAME="{NAME}"><FIRSTNAME="{FIRSTNAME}">
<LASTNAME="{LASTNAME}"><TITLE="{TITLE}"><TEL_CELL="{TEL_CELL}">
<TEL_WORK="{TEL_WORK}"><TEL_HOME="{TEL_HOME}"><EMAIL_HOME="{EMAIL_HOME}">
<EMAIL_WORK="{EMAIL_WORK}"><URL="{URL}"><STREET="{STREET}"><CITY="{CITY}">
<ZIP="{ZIP}"><COUNTRY="{COUNTRY}"><ORG="{ORG}"><TEL_FAX="{TEL_FAX}">
<HEIGHT>
```

Daten:

Data 1	Data 2
<pre><STREET="Muehlenstrasse 20"> <TEL_WORK="+495904964426"> <CITY="Lengerich"> <_DESCRIPTION> <ORG="EWS GmbH & Co KG"> <TEL_HOME> <COUNTRY="Germany"> <TEL_CELL="+4915901600051"> <_DATAID="Andreas Lange"> <FIRSTNAME="Andreas"> <ZIP="49838"> <EMAIL_HOME="andreas@a-lange.de"> <TEL_FAX="+493094631134"> <NAME="Andreas Lange"> <URL="www.ews-gmbh.org"> <LASTNAME="Lange"> <TITLE="ESL-Team Germany"> <EMAIL_WORK="a.lange@ews-gmbh.org"></pre>	

Ergebnis:**PRICE**

Ausgabe eines Preises mit der Möglichkeit, die Nachkommastellen in einer anderen Größe versetzt und ggf. unterstrichen darzustellen. Zusätzlich kann ein Währungssymbol eingeblendet werden. Das Preisfeld kann weiterhin mit einer Grafik überlagert werden, um z.B. den Preis insgesamt durchzustreichen. Die Positionen der einzelnen Bestandteile des Preises (*Vorkomma*, *Nachkomma*, *Währung*) können in ihrer Größe und Position angepasst werden.

Parameter	Erklärung
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite des Preisfeldes

HEIGHT	Höhe des Preisfeldes
BACKCOLOR	Hintergrundfarbe des Preisfeldes als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
TEXTCOLOR	Hintergrundfarbe des Preisfeldes als Konstante: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. 255,0,0
KOFFSETX	Abstand des Kommas zum Hauptpreis. Kann negativ oder positiv sein
NSCALE	Skalierungsfaktor der Nachkommastellen
NOFFSETX	Horizontaler Abstand der Nachkommastellen zum Hauptpreis. Kann negativ oder positiv sein
NOFFSETY	Vertikaler Abstand der Nachkommastellen zum Hauptpreis. Kann negativ oder positiv sein
WSCALE	Skalierungsfaktor des Währungssymbols
WOFFSETX	Horizontaler Abstand des Währungssymbols zum Hauptpreis. Kann negativ oder positiv sein
WOFFSETY	Vertikaler Abstand des Währungssymbols zum Hauptpreis. Kann negativ oder positiv sein
IMAGE	Ein Bild (<i>Dateiname</i>), welches dem Preisfeld überlagert wird. Damit kann z.B. eine Grafik zum Durchstreichen des Preises ausgegeben werden

IMAGEBASE64	Ein Bild im BASE64 -Format, welches dem Preisfeld überlagert wird. Damit kann z.B. eine Grafik zum Durchstreichen des Preises ausgegeben werden
IMAGETRANSPARENT	Stellt das Bild mit transparentem Hintergrund dar
ROTATE	Drehwinkel des Preisfeldes in Grad
RIGHT	Preis rechtsbündig ausrichten
FONT	Name des Fonts (<i>installierte Schriftarten</i>)
FONTSIZE	Font-Größe des Textes
BOLD	Fett
ITALIC	Kursiv
REGULAR	Standard
STRIKEOUT	Durchgestrichen
UNDERLINE	Unterstrichen

Beispiel:

```
<CMD="PRICE"><X="0"><Y="0"><WIDTH="200"><HEIGHT="100"><TEXT="99.95">
<TEXTCOLOR="BLACK"><FONT="Calibri"><FONTSIZE="50"><NSCALE="50"><NOFF
SETY="-40"><WAEHRUNG="€"><WSCALE="70"><WOFFSETX="20"><WOFFSETY="-
20">
```

NFC (*Hanshow*)

Kodiert Informationen für den in den ESL-Labels verbauten **NFC**-Chip. Daten im **öffentlichen** Bereich können mit allen gängigen NFC-Lesegeräten (z.B. *Smartphone des Kunden*) ausgelesen werden. Die Daten werden im **NDEF**-Format abgelegt, so dass ein Smartphone anhand der Kodierung erkennen kann, was es mit den gelesenen Daten anfangen soll (bei **URI_STANDARD** z.B. *die angegebene Webseite zu öffnen*).

<i>Parameter</i>	<i>Erklärung</i>
URI_STANDARD	Es wird eine URI im öffentlichen Bereich des NFC -Chip abgelegt, die z.B. von einem Smartphone gelesen und ausgewertet werden kann. Es muss sich um eine der Well-Formed-Kodierungen im NDEF -Standard handeln (siehe NDEF-Dokumentation im Internet) Beispiel: http://www.esl-easy-de
TEXT_STANDARD	Es wird ein freier Text im öffentlichen Bereich des NFC -Chip abgelegt, der z.B. von einem Smartphone gelesen und ausgewertet werden kann
URI_PRIVATE	Es wird eine URI im privaten Bereich des NFC -Chip abgelegt, die z.B. von einem Smartphone gelesen und ausgewertet werden kann. Es muss sich

	um eine der Well-Formed-Kodierungen im NDEF -Standard handeln (<i>siehe NDEF-Dokumentation im Internet</i>). Für das Lesen und Auswerten des privaten Bereichs des NFC-Chip ist spezielle HANSHOW-Software notwendig
TEXT_PRIVATE	Es wird ein freier Text im privaten Bereich des NFC -Chip abgelegt, der z.B. von einem Smartphone gelesen und ausgewertet werden kann. Für das Lesen und Auswerten des privaten Bereichs des NFC-Chip ist spezielle HANSHOW-Software notwendig
NFC_KEY	Ein 16 Zeichen langer Text, der als Schlüssel für das Lesen des privaten Bereichs des NFC -Chip verwendet wird. Bei einem kürzeren Text wird dieser vorne mit Leerzeichen aufgefüllt.
TAG_PERMISSION	Bestimmt die Zugriffsberechtigung für den öffentlichen Bereich des NFC -Chip. READONLY – Nur lesen WRITE – Nur schreiben READWRITE – Lesen und schreiben Zum Setzen der Berechtigung muss NFC_KEY angegeben werden
APP_PERMISSION	Bestimmt die Zugriffsberechtigung für den privaten Bereich des NFC -Chip. READONLY – Nur lesen WRITE – Nur schreiben READWRITE – Lesen und schreiben Zum Setzen der Berechtigung muss NFC_KEY angegeben werden

Beispiel:

```
<CMD="NFC"><URI_STANDARD="{URI_STANDARD}"><URI_PRIVATE="{URI_PRIVATE}"><NFC_KEY="GeheimerKey"><TAG_PERMISSION="READONLY"><APP_PERMISSION="READWRITE">
```

NFC (Overtime)

Kodiert Informationen für den in den ESL-Labels verbauten **NFC**-Chip.

<i>Parameter</i>	<i>Erklärung</i>
NFC_URL	Eine URL oder ein Text . Bei einer URL wird diese von einem Smartphone nach dem Lesen des NFC-Chips automatisch geöffnet Beispiel: http://www.esl-easy-de

Beispiel:

```
<CMD="NFC"><NFC_URL="http://www.esl-easy.de">
```

LOCATION

Mit diesem Befehl wird die Sichtbarkeit des Templates in der Template-Auswahl beschränkt. Es kann eine **Einschluss**- und eine **Ausschluss**-Liste angegeben werden. Entsprechend wird das Template nur angezeigt, wenn die **Location** in den Listen vorkommt.

Parameter	Erklärung
INCLUSION	Hier können Location -Nummern mit Komma getrennt angegeben werden. Das Template wird angezeigt, wenn die aktuelle Location in der INCLUSION -Liste vorkommt
EXCLUSION	Hier können Location -Nummern mit Komma getrennt angegeben werden. Das Template wird <u>nicht</u> angezeigt, wenn die aktuelle Location in der EXCLUSION -Liste vorkommt

Beispiel:

```
<CMD="LOCATION"><INCLUSION="1"><EXCLUSION="">
```

CHECKDATA

Mit **CHECKDATA** ist es möglich, die Nutzdaten (*interne bzw. externe Daten vom Universal-Gateway*) zu prüfen, ob sie zum aktuellen Template passen. Damit kann verhindert werden, dass ein falsches Template vom Anwender gewählt wird, was nicht zu den Nutzdaten passt. Dies kann vorkommen, wenn es mehrere Templates für einen ESL-Label-Typ gibt. Die Prüfung kann auf jedes bereitgestellte Datenfeld erfolgen. Es können auch Prüfungen von mehreren Werten im Datenfeld mit **UND** bzw. **ODER** erfolgen. Ist die Prüfung negativ (*es stimmen also erwartete Werte nicht überein*), wird dem Anwender bei einer Label-Verknüpfung eine Meldung angezeigt. Je nach gewählter **ACTION** kann der Anwender dann wählen, ob er fortfahren möchte oder der Vorgang abgebrochen wird.

Parameter	Erklärung
FIELD	Datenfeld aus den Nutzdaten, welches zur Prüfung herangezogen wird
VALUE	Der hier angegebene Wert wird mit dem Inhalt vom Feld verglichen
VALUESAND	Es können mehrere Werte mit Komma getrennt angegeben werden. Jeder Wert wird mit dem Inhalt vom Feld verglichen. Es müssen alle Werte mit dem Feldinhalt übereinstimmen
VALUESOR	Es können mehrere Werte mit Komma getrennt angegeben werden. Jeder Wert wird mit dem Inhalt vom Feld verglichen. Es muss mindestens ein Wert mit dem Feldinhalt übereinstimmen
COMPARISION	Die Art des Vergleichs zwischen dem Feldinhalt und dem Vergleichswert aus VALUE . EQUAL – Der Vergleich muss korrekt sein (<i>Feld = Wert</i>) UNEQUAL – Der Vergleich darf nicht korrekt sein (<i>Feld <> Wert</i>)

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

	CONTAINS – Der Wert muss im Feld enthalten sein (<i>InString-Vergleich</i>)
MESSAGE	Der Meldungstext, der dem Anwender angezeigt wird, wenn die Prüfung negativ ist
ACTION	YESNO – Der Anwender kann auswählen, ob er fortfahren möchte OK – Der Anwender kann die Meldung nur bestätigen, es wird mit der Verknüpfung fortgefahren CANCEL - Der Anwender kann die Meldung nur bestätigen, die Verknüpfung wird abgebrochen

Beispiel:

```
<CMD="CHECKDATA"><FIELD="WGR"><VALUE="10"><COMPARISON="UNEQUAL"><MESSAGE="Artikel und Template passen nicht zur Produktgruppe WEIN. Soll die Verknüpfung trotzdem erfolgen?"><ACTION="YESNO">
```

PARAM

Es können Label-Spezifische Parameter im Template definiert werden, die der Anwender bei der Label-Verknüpfung individuell für das aktuelle Label setzen kann. Damit ist es möglich, z.B. einen Pfeil auf einem Label einzublenden, der nur auf dem aktuellen Label erscheinen soll, auch wenn es weitere Labels gibt, die mit den gleichen Daten verknüpft sind. Solche Daten können nicht aus den externen Daten geliefert werden, da sie individuell bezogen auf das aktuelle Label gesetzt werden müssen. Die Label-Parameter werden in ESL-Easy direkt zur Label-ID gespeichert.

Im Template kann auf die Label-Parameter gleich den Nutzdaten zugegriffen werden (*{Parametername}*). Damit stehen die gleichen Möglichkeiten zum Vergleich und der Ausgabe zur Verfügung, wie für die Nutzdaten.

<i>Parameter</i>	<i>Erklärung</i>
NAME	Name des Parameters
VALUES	Werte mit Semikolon getrennt, die der Anwender für diesen Parameter als Vorauswahl bekommt. Bleibt VALUES leer, kann der Anwender einen beliebigen Text für den Parameter erfassen
GROUP	Ein Parameter kann einer Gruppe zugeordnet werden. Wenn GROUP angegeben wurde, werden die Label-Parameter auf dem mobilen Gerät gruppiert mit der Gruppenüberschrift angezeigt
LOCATIONINCLUSION	Hier kann festgelegt werden, in welchen Locations der Parameter angezeigt werden soll. Mehrere Locations können mit Komma getrennt angegeben werden
LOCATIONEXCLUSION	Hier kann festgelegt werden, in welchen Locations der Parameter <u>nicht</u> (<i>Ausschlussliste</i>) angezeigt werden soll. Mehrere Locations können mit Komma getrennt angegeben werden

Beispiel:

```
<CMD="PARAM"><NAME="PFEIL"><VALUES="OHNE;OBEN;RECHTS;UNTEN;LINKS">
```

ATEXT (*Attribute Text*)

Mit diesem Kommando ist es möglich, mit einem Text-Befehl einen Text unterschiedlich zu formatieren (z.B. *unterschiedliche Schriften, Schriftgrößen, Farben und Stiele*). Dazu können in den eigentlichen Text Steuerbefehle eingefügt werden.

Für die allgemeine Formatierung des Textes können die Parameter von „**TEXT**“ genutzt werden (z.B. **LTRIM**, **MAXLENGTH**, usw., siehe **TEXT**).

Parameter	Erklärung
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite
HEIGHT	Höhe
TEXT	Text mit Steuerbefehlen (<i>siehe nächste Tabelle</i>)
ANGLE	Drehwinkel
ROTATIONPOINT	Punkt, um den rotiert wird UPPERLEFT UPPERCENTER UPPERRIGHT CENTERLEFT CENTERCENTER CENTERRIGHT LOWERLEFT LOWERCENTER LOWERRIGHT
LEFT CENTER RIGHT	Textausrichtung
JUSTIFICATION	Blocksatz
BGCOLOR	Hintergrundfarbe
BGDISTANCE	Rahmenbreite der Hintergrundfarbe

Steuerbefehle innerhalb des Textes werden mit **zwei Unterstrichen** eingeleitet und abgeschlossen (__Formatierungsbefehl:Wert__).

Steuerbefehl in TEXT	Erklärung
__COLOR:Farbe__	Textfarbe: BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B z.B. __COLOR:RED__ oder __COLOR:255,0,0__
__BASE:Pixel__	Abweichung der vertikalen Position des folgenden Textes in Pixel an __BASE:-5__ area. an area.
__STYLE:Stil__	BOLD ITALIC REGULAR STRIKEOUT UNDERLINE Es können mehrere Angaben kombiniert werden (z.B. ITALICBOLD)
__FONT:Fontname,Fontgröße,Fontstyle__	Fontname, Fontgröße, Fontstyle Beispiel: __FONT:Sanserif,30,REGULAR__
__SIZE:FontSize__	Schriftgröße (es wird nur die Fontgröße geändert, nicht der Font selbst)

Beispiel:

```
<CMD="ATEXT"><Y="13"><X="12"><WIDTH="378"><HEIGHT="279"><TEXT=" __STY
LE:REGULAR__ With the new command
__STYLE:BOLD__ COLOR:RED__ ATEXT__ STYLE:REGULAR__ COLOR:BLACK__
different __STYLE:ITALIC__ formatting__ STYLE:ITALICBOLD__ of
__STYLE:UNDERLINE__ text areas__ STYLE:REGULAR__ can be combined
within a text line. Previously it was only possible to display the
text of one size and one font for each
__FONT:Sanserif,30,REGULAR__ TEXT__ FONT:Arial,20,REGULAR__ command.
With __FONT:Times New
Roman,30,BOLDITALIC__ ATEXT__ FONT:Arial,20,REGULAR__ you can specify
formatting commands within the text, which then use different
__FONT:SegoeUI,25,REGULAR__ fonts, __FONT:SegoeUI,20,UNDERLINE__ sizes
and __COLOR:GREEN__ colors__ FONT:ARIAL,
20,REGULAR__ COLOR:BLACK__ allow. It is also possible to use
wordwrap and autosize to automatically fit the text
```

```
__STYLE:STRIKEOUT__euf __STYLE:REGULAR__to an
area."><COLOR="BLACK"><FONT="Calibri"><FONTSIZE="22"><WORDWRAP><AUTO
SIZE><JUSTIFICATION>
```

Ergebnis:

With the new command **ATEXT** different *formatting of text areas* can be combined within a text line. Previously it was only possible to display the text of one size and one font for each **TEXT** command. With **ATEXT** you can specify formatting commands within the text, which then use different **fonts**, **sizes** and **colors** allow. It is also possible to use wordwrap and autosize to automatically fit the text euf to an area.

400px

COCKPITCHART

Stellt eine Cockpit-Grafik dar.

Parameter	Erklärung
X	Horizontale Position
Y	Vertikale Position
WIDTH	Breite
HEIGHT	Höhe
VALUE	Angezeigter Wert
MAXVALUE	Höchster möglicher Wert
TEXT	Angezeigter Text im Chart
FONT	Font des Textes
FONTSIZE	Größe des Fonts
EDGE	Breite des Randes des Charts
COLOREDGE COLORVALUE COLORBACK COLORVALUEMARKER	BLACK BLUE GREEN CYAN RED MAGENTA YELLOW WHITE oder als RGB-Wert R,G,B

ESL-Easy Template-Editor und Befehlsreferenz

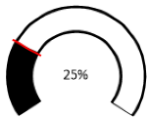
Datum: 03.09.2025

	z.B. 255,0,0
TEXTAUTOSIZE	TRUE Wenn angegeben, wird der Text automatisch skaliert
VALUEMARKER	TRUE Wenn angegeben, wird eine Markierung am angegebenen VALUE angezeigt
VALUEMARKERWIDTH VALUEMARKERHEIGHT	Höhe und Breite des Markers in Pixel
LINECAPBACK LINECAPEDGE LINECAPVALUE LINECAPVALUEMARKER	Legt die Art des Linienbeginn und Linienende fest: ARROWANCHOR - Gibt ein pfeilförmiges Ankerende an DIAMONDANCHOR - Gibt ein rautenförmiges Ankerende an FLAT -Gibt ein abgeflachtes Linienende an NOANCHOR - Gibt an, dass kein Anker verwendet wird ROUND - Gibt ein rundes Linienende an ROUNDANCHOR - Gibt ein rundes Ankerende an SQUARE - Gibt ein quadratisches Linienende an SQUAREANCHOR - Gibt ein quadratisches Ankerlinienende an TRIANGLE - Gibt ein dreieckiges Linienende an
BARWIDTH	Breite des Chart-Balkens
ANGLE	Drehwinkel für die Darstellung
ROTATIONPOINT	Punkt, um den rotiert wird UPPERLEFT UPPERCENTER UPPERRIGHT CENTERLEFT CENTERCENTER CENTERRIGHT LOWERLEFT LOWERCENTER LOWERRIGHT
OPENCIRCLEPERCENT	Offener Bereich des Charts in Prozent (<i>unten</i>)

Beispiel:

```
<CMD="COCKPITCHART"><Y="29"><X="37"><WIDTH="100"><HEIGHT="100"><VALUE="25"><MAXVALUE="100"><TEXT="25%"><FONT="Calibri"><FONTSIZE="22"><EDGE="2"><TEXTAUTOSIZE="TRUE"><VALUEMARKER="TRUE"><COLORVALUEMARKER="255,0,0">
```

Ergebnis:



END

Bricht die Verarbeitung des Templates an dieser Stelle ab. Alle folgenden Befehlszeilen werden nicht mehr verarbeitet.

Dynamische Ausrichtung von Objekten

Objekte können in Abhängigkeit zu anderen Objekten positioniert bzw. angepasst werden. Dazu kann jedem Objekt im Quellcode ein Objekt-Name zugewiesen werden.

```
<NAME="Object1">
```

In jedem anderen Objekt im Quellcode (*nachfolgend zu diesem Objekt – ein Objekt muss im Quellcode bereits gezeichnet worden sein, um auf seine Daten zugreifen zu können*) kann nun auf Parameter des Objektes mit dem angegebenen Namen zugegriffen werden. Damit lassen sich Objekte dynamisch in der Größe anpassen oder seine Position anhand der Größe des Referenzobjektes verschieben.

Achtung: Das Ergebnis dynamischer Objektanpassungen wird technisch bedingt nicht im Template-Editor grafisch dargestellt. Erst die Vorschau zeigt das endgültige Ergebnis an.

Folgende Befehle stehen für die dynamische Berechnung von Objekten zur Verfügung:

Parameter	Erklärung
ADDXFROM	Addiert alle Attribute zur X-Koordinate hinzu für eine berechnete Position X
SUBXFROM	Subtrahiert alle Attribute von der X-Koordinate für eine berechnete Position X
ADDYFROM	Addiert alle Attribute zur Y-Koordinate hinzu für eine berechnete Position Y
SUBYFROM	Subtrahiert alle Attribute von der Y-Koordinate für eine berechnete Position Y
ADDWIDTHFROM	Addiert alle Attribute zur WIDTH-Koordinate hinzu für eine berechnete Position WIDTH
SUBWIDTHFROM	Subtrahiert alle Attribute von der WIDTH-Koordinate für eine berechnete Position WIDTH
ADDHEIGHTFROM	Addiert alle Attribute zur HEIGHT-Koordinate hinzu für eine berechnete Position HEIGHT

SUBHEIGHTFROM	Subtrahiert alle Attribute von der HEIGHT-Koordinate für eine berechnete Position HEIGHT
ADDTOXFROM	Addiert alle Attribute zur TOX-Koordinate hinzu für eine berechnete Position TOX (<i>Linien</i>)
SUBTOXFROM	Subtrahiert alle Attribute von der TOX-Koordinate für eine berechnete Position TOX (<i>Linien</i>)
ADDTOYFROM	Addiert alle Attribute zur TOY-Koordinate hinzu für eine berechnete Position TOY (<i>Linien</i>)
SUBTOYFROM	Subtrahiert alle Attribute von der TOY-Koordinate für eine berechnete Position TOY (<i>Linien</i>)

Aufbau der dynamischen Befehle

<Befehl, @Objektname, Parameter, Parameter, @Objektname, Parameter, Parameter, ...>

Beispiel:

<ADDYFROM="@OBJECT1, Y, HEIGHT">

Addiert zum angegebenen Y-Wert die Werte **Y** und **HEIGHT** vom Objekt „**OBJECT1**“ hinzu. Ist also als <Y="10"> angegeben und im OBJECT1 die Werte <Y="20"><HEIGHT="10">, dann würde sich als Ergebnis für Y = 40 ergeben (10 + 20 + 10).

Es kann auf Werte aus unterschiedlichen Objekten zugegriffen werden. Dazu wird in den Parametern zuerst der Objektname angegeben, dann folgen die Parameter, auf die zugegriffen werden soll. Danach wiederholt sich diese Angabe mit dem nächsten Objekt.

Beispiel:

<ADDYFROM="@OBJECT1, Y, HEIGHT, @OBJECT2, Y, HEIGHT">

Addiert zu **Y** den **Y**-Wert und den **HEIGHT**-Wert vom **OBJECT1** und vom **OBJECT2** hinzu.

Anstelle von Parametern können auch fixe Zahlen angegeben werden.

Beispiel:

<ADDXFROM="@OBJECT1, Y, WIDTH, 20"> - Addiert noch **20 Punkte** zum Ergebnis dazu.

Wird nur der Objektname angegeben, wird auf das zum Befehl passende Attribut zugegriffen.

Beispiel:

<ADDXFROM="PRICE"> -> greift auf das Objekt mit dem Namen **PRICE** und auf das Attribut **X** zu

Wird nur eine Zahl angegeben, so wird diese Zahl entsprechend addiert bzw. subtrahiert.

Beispiel:

<ADDXFROM="-100"> - Subtrahiert **100 Punkte** von **X** (entspricht <SUBXFROM="100">)

Bei **IMAGE**, **BARCODE**, **QRCODE**, **QRCODE_VCARD** und **PRICE** müssen im Quellcode **WIDTH** und **HEIGHT** größer **0** angegeben werden, sonst können diese Objekte nicht gezeichnet werden. Bei einer Abhängigkeit muss **WIDTH** bzw. **HEIGHT** wieder abgezogen werden, wenn sich die Abhängigkeit direkt auf ein anderes Objekt bezieht.

Beispiel:

```
<CMD="RECTANGLE"><NAME="REC3"><Y="200"><X="0"><WIDTH="400"><HEIGHT="100"><ADDYFROM="@REC1,Y,HEIGHT,-200,@REC2,HEIGHT"><SUBHEIGHTFROM="@REC1,HEIGHT,@REC2,HEIGHT"><ADDHEIGHTFROM="200"><COLOR="BLACK"><THICKNESS="1"><STYLE="SOLID"><BGCOLOR="BLACK">
```

Variablen

Im Template können Variablen mit Inhalten vorgegeben werden, auf die dann innerhalb einer **WHEN**-Bedingung zugegriffen werden kann.

SETVARIABLE

Setzt eine Variable auf einen Wert.

Parameter	Erklärung
NAME	Name der Variable
VALUE	Inhalt der Variable

Beispiel:

```
<CMD="SETVARIABLE"><NAME="FILIALE"><VALUE="0815">
```

CLEARVARIABLE

Löscht eine Variable.

Parameter	Erklärung
NAME	Name der Variable

Beispiel:

```
<CMD="CLEARVARIABLE"><NAME="FILIALE">
```

CLEARALLVARIABLE

Löscht alle Variablen.

Beispiel:

```
<CMD="CLEARALLVARIABLE">
```

Variable abfragen (Beispiel)

```
<CMD="TEXT"><WHEN="[ISEQUAL~(FILIALE)~0815]"><X="334"><WIDTH="153"><HEIGHT="62"><TEXT="FILIALE stimmt mit 0815 überein"><COLOR="BLACK"><FONT="Calibri"><FONTSIZE="22"><Y="519">
```

Makros

Makros werden in der Befehlszeile durch die entsprechende Kodierung ersetzt. Es wird die Standardformatierung des Betriebssystems für die Ausgabe verwendet. Es kann eine eigene Formatierung für das **Datum** und die **Uhrzeit** in der Befehlszeile angegeben werden, die die Standardformatierung überschreibt (<**DATEFORMAT**=""> und <**TIMEFORMAT**="">):

Beispiel:

<**DATEFORMAT**="dd.MM.yyyy"> für 31.12.2020

<**TIMEFORMAT**="HH:mm:ss"> für 12:09:23

<i>Funktion</i>	<i>Kodierung</i>
%%TODAY%%	Aktuelles Datum
%%TODAYMINUSONE%%	Aktuelles Datum – 1 Tag
%%TODAYMINUSTWO%%	Aktuelles Datum – 2 Tage
%%TODAYMINUSTHREE%%	Aktuelles Datum – 3 Tage
%%TODAYMINUSFOUR%%	Aktuelles Datum – 4 Tage
%%TODAYMINUSFIVE%%	Aktuelles Datum – 5 Tage
%%TODAYPLUSONE%%	Aktuelles Datum + 1 Tag
%%TODAYPLUSTWO%%	Aktuelles Datum + 2 Tage
%%TODAYPLUSTHREE%%	Aktuelles Datum + 3 Tage
%%TODAYPLUSFOUR%%	Aktuelles Datum + 4 Tage
%%TODAYPLUSFIVE%%	Aktuelles Datum + 5 Tage
%%WEEKDAY%%	Aktueller Wochentag als Text (<i>in der eingestellten Landessprache des PC</i>)
%%WEEKDAYPLUSONE%%	Aktueller Wochentag + 1 als Text (<i>in der eingestellten Landessprache des PC</i>)
%%WEEKDAYPLUSTWO%%	Aktueller Wochentag + 2 als Text (<i>in der eingestellten Landessprache des PC</i>)
%%WEEKDAYPLUSTHREE%%	Aktueller Wochentag + 3 als Text (<i>in der eingestellten Landessprache des PC</i>)
%%WEEKDAYPLUSFOUR%%	Aktueller Wochentag + 4 als Text (<i>in der eingestellten Landessprache des PC</i>)
%%WEEKDAYPLUSFIVE%%	Aktueller Wochentag + 5 als Text (<i>in der eingestellten Landessprache des PC</i>)

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

%%WEEKDAYPLUSSIX%%	Aktueller Wochentag + 6 als Text (<i>in der eingestellten Landessprache des PC</i>)
%%TOMORROW%%	Morgen
%%YESTERDAY	Gestern
%%FIRSTDAYOFTHEWEEK%%	Erster Tag der aktuellen Woche
%%WEEK%%	Kalenderwoche des aktuellen Datums
%%DAY%%	Aktueller Tag im Format dd (z.B. 04)
%%MONTH%%	Aktueller Monat im Format MM (z.B. 01)
%%SHORTYEAR%%	Aktuelles Jahr im Format yy (z.B. 20)
%%LONGYEAR%%	Aktuelles Jahr im Format yyyy (z.B. 2020)
%%YEAR%%	Aktuelles Jahr im Format yyyy (z.B. 2020, gleich %%YEAR%%)
%%TIME%%	Aktuelle Uhrzeit im Format HH:mm:ss (z.B. 08:55:25)
%%HOUR%%	Aktuelle Stunde im Format HH (z.B. 08)
%%MINUTE%%	Aktuelle Minute im Format mm (z.B. 56)
%%SECOND%%	Aktuelle Sekunde im Format ss (z.B. 31)
%%IPADDRESS%%	Aktuelle IP-Adresse
%%IPADDRESSES%%	Liste aller aktuellen IP-Adressen mit Komma getrennt als ein Text
%%APPPATH%%	Aktueller Programmpfad der Applikation
%%HOSTNAME%%	Hostname des PC
%%VERSION%%	Programmversion der Applikation (<i>für interne Verwendung</i>)
%%TOOLSVERSION%%	Verwendete DLL-Version der BSL_Tools (<i>für interne Verwendung</i>)

Makrofelder

In den folgenden Feldern gibt es zusätzliche Makro-Auswertungen, die mit eckigen Klammern angegeben werden:

Command	Tag
NFC	PAYLOAD

TEXT	FILE, TEXT
IMAGE	FILE
BARGRAPH	PERCENT
BARCODE	TEXT
QRCODE	TEXT
PRICE	TEXT, IMAGE

DATAVALUE

Mit **[DATAVALUE~Datenfeld]** kann auf ein Datenfeld zugegriffen werden. Dies ist eine alternative Methode gegenüber der einfachen Schreibweise **{Datenfeld}**. Wird innerhalb von Macrofeldern auf Daten zugegriffen, wird diese Schreibweise benötigt.

Beispiele:

[DATAVALUE~NAME] -> Gibt den Inhalt des Datenfeldes **NAME** aus.

```
<TEXT=" [SPLITREPEAT~[DATAVALUE~LIST]~,~__SPLITVALUE__~-] ">
```

Datenfeld: <LIST="Jan, Feb, Mar, Apr">

Ergebnis: Jan-Feb-Mar-Apr

SPLITREPEAT

[SPLITREPEAT~Text~Trennzeichen~Teilstring~Ersatzzeichen]

Trennt einen Text anhand des angegebenen Trennzeichens auf und gibt die einzelnen Elemente neu formatiert aus.

Teilstring – Dieser Text wird für jeden aufgetrennten Teilstring ausgegeben. Um auf den originalen Text zuzugreifen, muss **__SPLITVALUE__** angegeben werden. Bei freiem Text wird dieser für jeden Teilstring aus dem Trennvorgang ausgegeben.

Beispiele:

```
<TEXT=" [SPLITREPEAT~1,2,1~,~__SPLITVALUE__~-] ">
```

Ausgabe: 1-2-3

Der Text **1,2,3** wird aufgeteilt in Teilstrings anhand des **Kommas**. Danach wird jeder Teilstring mit einem **Minus**-Zeichen getrennt neu zusammengebaut und ausgegeben.

```
<TEXT=" [SPLITREPEAT~1,2,1~,~X~---] ">
```

Ausgabe: X---X---X

Der Text **1,2,3** wird aufgeteilt in Teilstrings anhand des Kommas. Danach wird für jeden Teilstring ein **X** eingesetzt und mit **drei Minus**-Zeichen getrennt neu zusammengebaut und ausgegeben.

REGEX

Mit **[REGEX~Text~Pattern~]** kann ein regulärer Ausdruck ausgewertet werden. Das Ergebnis wird als Text eingesetzt.

Beispiel:

```
<TEXT="[REGEX~Eins Zwei Drei~\b[A-Z]~]">
```

Ausgabe: EZD

Der Text wird nach **Großbuchstaben** durchsucht und diese als neuer Text zurückgegeben.

CALC

Mit **[CALC~Rechenformel~Formatierung]** kann eine Berechnung durch Angabe einer Rechenformel erfolgen. Dabei können feste Zahlenwerte oder Werte aus den Daten verwendet werden.

Beispiel:

```
<TEXT="Grundpreis [CALC~{PRICE}/{QUANTITY}~0.00]€ / {UNIT}">
```

Ausgabe z.B.: **Grundpreis 59.60 / kg**

Die Berechnungsformel wird nach **[CALC~]** angegeben. Es können komplexe Berechnungsausdrücke mit Klammerebenen genutzt werden, um eine Berechnung durchzuführen. Anstelle fester Zahlenwerte können die Werte auch aus den Datenfeldern kommen (**{Datenfeld}**).

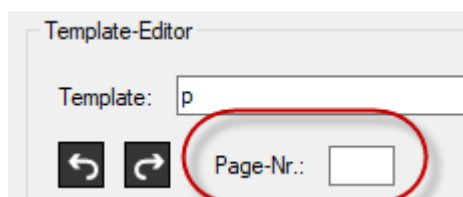
Nach der Angabe der Berechnungsformel kann eine Formatierung des berechneten Wertes erfolgen. Siehe dazu **Formatierung von Datenfeldern**.

Beispiel: **[CALC~...~0.00|0|.] -> 59.60**

Der berechnete Wert wird mit zwei Nachkommastellen ausgegeben und das Dezimaltrennzeichen durch einen Punkt ersetzt (*falls das Dezimaltrennzeichen z.B. ein Komma ist*).

PAGE

Eine Besonderheit der **ESL-Labels** besteht darin, dass mehrere Ansichten im Label abgelegt werden können. Es wird dann auf dem Display immer die Ansicht angezeigt, die aktuell aktiv gesetzt ist. Um mehrere Ansichten in einem Template generieren zu können, kann für jede Ansicht ein eigener Quellcodebereich definiert werden. Im grafischen Editor wird aktuell die Page angezeigt, die unter



angegeben wird. Wird keine Pagenummer vorgegeben bzw. im Template nur eine Ansicht für das Label definiert (*keine PAGE-Befehl vorhanden*), wird alles der ersten Page zugewiesen und angezeigt.

Command	Tag
PAGENO	Aktuelle Page-Nummer für das ESL-Label. Ab dieser Zeile werden alle folgenden Zeilen der angegebenen Page zugewiesen

Der ESL-Easy-Server kennt Befehle zum Umschalten der Ansicht auf den ESL-Labels (*siehe Befehlsreferenz ESL-Easy Server*).

Beispiel:

```
<CMD="PAGE"><PAGENO="1">
```

```
<CMD="TEXT"><Y="0"><X="0"><WIDTH="554"><HEIGHT="416"><TEXT="[SPLITRE
PEAT~1,2,1~,~__SPLITVALUE__~-
]"><COLOR="BLACK"><FONT="Calibri"><FONTSIZE="22"><BOLD="FALSE"><AUTO
SIZE><WORDWRAP>
```

Der **rot** markierte Quellcode gehört zur Page 1.

```
<CMD="PAGE"><PAGENO="2">
```

```
<CMD="TEXT"><Y="0"><X="0"><WIDTH="554"><HEIGHT="416"><TEXT="[SPLITRE
PEAT~2,1,2~,~__SPLITVALUE__~-
]"><COLOR="BLACK"><FONT="Calibri"><FONTSIZE="22"><BOLD="FALSE"><AUTO
SIZE><WORDWRAP>
```

```
<CMD="ELLIPSE"><Y="0"><X="0"><WIDTH="100"><HEIGHT="100"><COLOR="BLAC
K"><STYLE="SOLID"><THICKNESS="2"><BGCOLOR="255,0,0">
```

```
<CMD="POLYGON"><COLOR="BLACK"><BGCOLOR="255,0,0"><POINTS="10,10;100,
100;150,100;150,200">
```

Der **blau** markierte Quellcode gehört zur Page 2.

WHEN

Für jedes Kommando in einer Template-Datei kann eine **WHEN**-Bedingung angegeben werden. Ist die **WHEN**-Bedingungen erfüllt, wird die Zeile ausgeführt. In einem **WHEN**-Parameter können mehrere Bedingungen kombiniert werden.

Bei einer **IF**-Bedingung bestimmt die **WHEN**-Bedingung, ob der folgende Quellcodeabschnitt (*bis zum ELSE bzw. ENDIF*) ausgeführt wird.

Kombination von WHEN-Bedingungen

WHEN-Parameter	Beschreibung
----------------	--------------

WHEN, WHENAND	Alle Bedingungen müssen erfüllt sein (UND-Verknüpfung), damit das Kommando ausgeführt wird
WHENOR	Mindestens eines der angegebenen Bedingungen muss erfüllt sein (ODER-Verknüpfung), damit das Kommando ausgeführt wird

Beispiele:

```
<CMD="TEXT"><WHEN=" [ IEMPTY~PREIS ] ">...
```

Jede **WHEN**-Angabe erfolgt im Format [*Befehl~Parameter 1~Parameter 2*], wobei je nach Befehl eine unterschiedliche Anzahl von Parametern existieren kann. Mehrere **WHEN**-Bedingungen werden hintereinander aufgeführt [*Befehl~Parameter 1~Parameter 2*] [*Befehl~Parameter 1~Parameter 2*]...

Beispiel:

```
<WHEN=" [ IEMPTY~PREIS ] [ ISTAG~SONDERPREIS ] ">
```

In den Parametern können **Makros** (siehe **Makros**) und Datenfelder (siehe **Datenfelder**) verwendet werden.

Liste der WHEN-Befehle

WHEN-Befehl	Beschreibung
ISTAG	Abfrage, ob das angegebene Datenfeld (<i>Tag</i>) in den verbundenen Daten vorhanden ist Parameter 1: Name des Tags Das Datenfeld wird ohne {} angegeben. Beispiel: <WHEN="[ISTAG~ITEMNO]">
ISNOTTAG	Abfrage, ob das angegebene Datenfeld (<i>Tag</i>) in den verbundenen Daten nicht vorhanden ist Parameter 1: Name des Tags Das Datenfeld wird ohne {} angegeben. Beispiel: <WHEN="[ISNOTTAG~ITEMNO]">
ISEQUAL	Abfrage, ob Parameter 1 und Parameter 2 gleich sind (<i>String-Vergleich</i>) String(Parameter 1) = String(Parameter 2)
ISNOTEQUAL	Abfrage, ob Parameter 1 und Parameter 2 ungleich sind (<i>String-Vergleich</i>) String(Parameter 1) <> String(Parameter 2)
ISINTEQUAL	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob beide gleich sind

	Int(Parameter 1) = Int(Parameter 2)
ISNOTINTEQUAL	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob beide ungleich sind Int(Parameter 1) <> Int(Parameter 2)
ISFLOATEQUAL	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob beide gleich sind Float(Parameter 1) = Float(Parameter 2)
ISNOTFLOATEQUAL	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob beide ungleich sind Float(Parameter 1) <> Float(Parameter 2)
ISLESSTHAN	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob Parameter 1 kleiner Parameter 2 ist Int(Parameter 1) < Int(Parameter 2)
ISFLOATLESSTHAN	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob Parameter 1 kleiner Parameter 2 ist Float(Parameter 1) < Float(Parameter 2)
ISLESSEQUAL	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob Parameter 1 kleiner gleich Parameter 2 ist Int(Parameter 1) <= Int(Parameter 2)
ISFLOATLESSEQUAL	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob Parameter 1 kleiner gleich Parameter 2 ist Float(Parameter 1) <= Float(Parameter 2)
ISGREATERTHAN	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob Parameter 1 größer Parameter 2 ist Int(Parameter 1) > Int(Parameter 2)
ISFLOATGREATERTHAN	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob Parameter 1 größer Parameter 2 ist Float(Parameter 1) > Float(Parameter 2)
ISGREATEREQUAL	Die Parameter werden als Int-Zahl betrachtet und ausgewertet, ob Parameter 1 größer gleich Parameter 2 ist Int(Parameter 1) >= Int(Parameter 2)
ISFLOATGREATEREQUAL	Die Parameter werden als Float-Zahl betrachtet und ausgewertet, ob Parameter 1 größer gleich Parameter 2 ist

	Float(Parameter 1) >= Float(Parameter 2)
ISLENGTHGREATERTHAN	Es wird geprüft, ob die String-Länge von Parameter 1(<i>Anzahl Zeichen</i>) größer der Angabe in Parameter 2 ist String(Parameter 1).Length > Int(Parameter 2)
ISLENGTHGREATEREQUAL	Es wird geprüft, ob die String-Länge von Parameter 1(<i>Anzahl Zeichen</i>) größer oder gleich der Angabe in Parameter 2 ist String(Parameter 1).Length >= Int(Parameter 2)
ISLENGTHLESSTHAN	Es wird geprüft, ob die String-Länge von Parameter 1(<i>Anzahl Zeichen</i>) kleiner der Angabe in Parameter 2 ist String(Parameter 1).Length < Int(Parameter 2)
ISLENGTHLESSEQUAL	Es wird geprüft, ob die String-Länge von Parameter 1(<i>Anzahl Zeichen</i>) kleiner oder gleich der Angabe in Parameter 2 ist String(Parameter 1).Length <= Int(Parameter 2)
STARTSWITH	Es wird geprüft, ob Parameter 1 mit der Angabe in Parameter 2 beginnt
NOTSTARTSWITH	Es wird geprüft, ob Parameter 1 mit der Angabe in Parameter 2 nicht beginnt
ENDSWITH	Es wird geprüft, ob Parameter 1 mit der Angabe in Parameter 2 endet
NOTENDSWITH	Es wird geprüft, ob Parameter 1 mit der Angabe in Parameter 2 nicht endet
CONTAINS	Es wird geprüft, ob Parameter 2 in Parameter 1 vorkommt (<i>enthält</i>)
CONTAINSNOT	Es wird geprüft, ob Parameter 2 in Parameter 1 nicht vorkommt (<i>enthält</i>)
ISEMPTY	Es wird geprüft, ob das Datenfeld (<i>Parameter 1</i>) leer ist. Das Datenfeld wird ohne {} angegeben. Beispiel: <WHEN="[ISEMPTY~ITEMNO]">
ISNOTEMPTY	Es wird geprüft, ob das Datenfeld (<i>Parameter 1</i>) nicht leer ist. Das Datenfeld wird ohne {} angegeben. Beispiel: <WHEN="[ISNOTEMPTY~ITEMNO]">
ISVALUEEMPTY	Es wird geprüft, ob Parameter 1 leer ist
ISNOTVALUEEMPTY	Es wird geprüft, ob Parameter 1 nicht leer ist

ISFILE	Es wird geprüft, ob das in Parameter 1 angegebene File vorhanden ist
ISNOTFILE	Es wird geprüft, ob das in Parameter 1 angegebene File nicht vorhanden ist
ISDATEEQUAL	Es wird geprüft, ob das Datum in Parameter 1 gleich dem Datum in Parameter 2 ist. Beide Parameter müssen ein gültiges Datum darstellen. Es können Macros verwendet werden, um mit dem aktuellen Datum zu arbeiten. Beispiel: <WHEN="[ISDATEEQUAL~%%TODAY%%~{HOLIDAY}]">
ISDATENOTEQUAL	Es wird geprüft, ob das Datum in Parameter 1 ungleich dem Datum in Parameter 2 ist. Beide Parameter müssen ein gültiges Datum darstellen. Es können Macros verwendet werden, um mit dem aktuellen Datum zu arbeiten. Beispiel: <WHEN="[ISDATENOTEQUAL~%%TODAY%%~{HOLIDAY}]">
ISDATEGREATERTHAN	Es wird geprüft, ob das Datum in Parameter 1 größer (<i>später</i>) dem Datum in Parameter 2 ist. Beide Parameter müssen ein gültiges Datum darstellen. Es können Macros verwendet werden, um mit dem aktuellen Datum zu arbeiten. Beispiel: <WHEN="[ISDATEGREATERTHAN~%%TODAY%%~2023-01-01]">
ISDATELESSTHAN	Es wird geprüft, ob das Datum in Parameter 1 kleiner (<i>früher</i>) dem Datum in Parameter 2 ist. Beide Parameter müssen ein gültiges Datum darstellen. Es können Macros verwendet werden, um mit dem aktuellen Datum zu arbeiten. Beispiel: <WHEN="[ISDATELESSTHAN~%%TODAY%%~2023-12-31]">
ISDATEGREATEREQUAL	Es wird geprüft, ob das Datum in Parameter 1 größer (<i>später</i>) oder gleich dem Datum in Parameter 2 ist. Beide Parameter müssen ein gültiges Datum darstellen. Es können Macros verwendet werden, um mit dem aktuellen Datum zu arbeiten. Beispiel: <WHEN="[ISDATEGREATEREQUAL~%%TODAY%%~2023-01-01]">
ISDATELESSEQUAL	Es wird geprüft, ob das Datum in Parameter 1 kleiner (<i>früher</i>) oder gleich dem Datum in Parameter 2 ist. Beide Parameter müssen ein gültiges Datum darstellen. Es können Macros verwendet werden, um mit dem aktuellen Datum zu arbeiten.

	Beispiel: <code><WHEN="[ISDATELESSEQUAL~%%TODAY%%~2023-12-31]"></code>
ISINLIST	Es wird geprüft, ob der Parameter 1 in der Liste weiterer Parameter enthalten ist. Beispiel: <code><WHEN="[ISINLIST~Andreas~Peter~Andreas~Marc]"></code> Ergebnis ist WAHR (<i>Andreas ist in der Liste enthalten</i>)
ISNOTINLIST	Es wird geprüft, ob der Parameter 1 nicht in der Liste der weiteren Parameter enthalten ist. Beispiel: <code><WHEN="[ISNOTINLIST~Eric~Peter~Andreas~Marc]"></code> Ergebnis ist WAHR (<i>Eric ist in der Liste nicht enthalten</i>)
CONTAINSINLIST	Es wird geprüft, ob einer der Parameter in Parameter 1 enthalten ist. Beispiel: <code><WHEN="[CONTAINSINLIST~Andreas~Pe~An~Ma]"></code> Ergebnis ist WAHR (<i>An ist in Andreas enthalten</i>)
CONTAINSNOTINLIST	Es wird geprüft, ob einer der Parameter nicht in Parameter 1 enthalten ist. Beispiel: <code><WHEN="[CONTAINSNOTINLIST~Andreas~Pe~Ab~Ma]"></code> Ergebnis ist WAHR (<i>Pe, Ab und Ma sind nicht in Andreas enthalten</i>)

IF

IF-Bedingungen erlauben es, ganze Bereiche in einem Template nur dann auszuführen, wenn die Bedingung erfüllt ist. **IF** wird immer in Verbindung mit **WHEN** benutzt. Eine **IF**-Bedingung hat einen **ELSE**-Bereich, der alternativ ausgeführt wird, wenn die Bedingung nicht erfüllt ist. Ein **IF**-Block wird mit **ENDIF** abgeschlossen.

Es sind verschachtelte **IF**-Bedingungen möglich.

Beispiel:

```
<CMD="IF"><WHEN=" [ISTAG~NAME] ">
  <CMD="TEXT">...
  ...
<CMD="ELSE">
  <CMD="TEXT">...
```

...
<CMD="ENDIF">

Datenfelder

In einer Template-Zeile kann auf verbundene Datenfelder mit der Angabe **{Datenfeld}** zugegriffen werden.

Beispiel:

Template:

```
<CMD="TEXT"><WHEN=" [ ISNOTEMPTY~PREIS ] "><Y="0"><X="0"><WIDTH="50"><HEIGHT="25"><TEXT="{ PREIS }"><COLOR="BLACK"><FONT="Calibri"><FONTSIZE="22">
```

Daten: <PREIS="1.99">

In der Template-Zeile wird **{PREIS}** mit dem Inhalt aus den Daten (*hier 1.99*) ersetzt.

Formatierung von Datenfeldern

Datenfelder können eine spezielle Formatierung erhalten. So ist es z.B. möglich, Zahlen mit zwei Nachkommastellen darzustellen, obwohl die Daten z.B. nur **<PREIS="1">** für 1.00 Euro enthalten. Weiterhin kann ein Teilungsfaktor angegeben werden, um z.B. Preise ohne Dezimaltrennzeichen trotzdem korrekt darzustellen.

Um ein Datenfeld zu formatieren, wird das Format **{@@Datenfeld|Formatangabe|Teilungsfaktor}** verwendet.

Formatangabe	Erklärung
0	Diese Stelle wird immer ausgegeben und ggf. mit 0 ausgegeben, wenn in dem Datenfeld diese Stelle nicht vorhanden ist. Daten: <INHALT="1"> Formatangabe: {@@INHALT 0.0} Ergebnis: 1,0 Daten: <INHALT="1.6789"> Formatangabe: {@@INHALT 0.00} Ergebnis: 1,68 Daten: <INHALT="16789"> Formatangabe: {@@INHALT 0.00 100} Ergebnis: 167,89
#	Diese Stelle wird nur ausgegeben, wenn sie in dem Datenfeld vorhanden ist.

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

	Daten: <INHALT="1"> Formatangabe: {@@INHALT 0.#} Ergebnis: 1 Daten: <INHALT="1.5"> Formatangabe: {@@INHALT 0.#} Ergebnis: 1,5
L <i>gefolgt von der Anzahl Stellen</i>	Vom Datenfeld werden von Links die Anzahl der Zeichen ausgegeben, der Rest wird weggelassen. Daten: <INHALT="1.12345"> Formatangabe: {@@INHALT L4} Ergebnis: 1.12
R <i>gefolgt von der Anzahl Stellen</i>	Vom Datenfeld werden von rechts die Anzahl der Zeichen ausgegeben, der Rest wird weggelassen. Daten: <INHALT="1.12345"> Formatangabe: {@@INHALT R4} Ergebnis: 2345
S <i>gefolgt von der Startposition , Anzahl Zeichen</i>	Gibt einen Ausschnitt (Substring) des Datenfeldes aus Der erste Parameter bestimmt die Startposition beginnend mit 0 für das erste Zeichen Der zweite Parameter gibt die Anzahl von Zeichen an, die ab der Startposition ausgegeben werden sollen. Daten: <INHALT="1.12345"> Formatangabe: {@@INHALT S2,3} Ergebnis: 123
CEILING	Der Wert wird auf die nächste Ganzzahl aufgerundet 1.3 = 2 1.9 = 2
FLOOR	Der Wert wird auf die nächste Ganzzahl abgerundet 1.3 = 1 1.9 = 1

Beispiele:

<TEXT="{@@PREIS|0.00|100}">

Inhalt Feld **PREIS**: <PREIS="10000">

Ergebnis: 100,00

ESL-Easy Template-Editor und Befehlsreferenz

Datum: 03.09.2025

<TEXT="{@@PREIS|0.00}">

Inhalt Feld **PREIS**: <PREIS="1.89654">

Ergebnis: 1,90

<TEXT="FLOOR: {@@PREIS|FLOOR}">

Inhalt Feld **PREIS**: <PREIS="1.89654">

Ergebnis: 1