

Csv2Tpl 1.3 Vereinbarungen

Der Aufbau erfolgt nach folgendem Muster

Daten aus Dateistart übernehmen, für jede Csv-Zeile das Template ausfüllen und Dateieinde anhängen. Dateistart und Dateieinde können auch leer sein.

Die CSV-Datei trennt die Spalten durch „;“ oder Tabulator

Platzhalter werden mit % umschlossen

Platzhalter	Bemerkung
CNT0 bis CNTx	Zähler, die pro csv-Zeile um eins erhöht werden. Die Initialisierung erfolgt über das Textfeld auf der Oberfläche. Die Werte werden durch Komma getrennt eingegeben. Nicht initialisierte Zähler werden nicht ersetzt.
CNT0+y bis CNTx+y	Der Zähler CNTx wird um y erhöht. Dadurch ist z.B.: CNT0 im nächsten Durchlauf um y+1 höher.
T0 – Tx	Inhalt der csv-Zeile aufgeteilt in die Spalten 0 bis x
L0 – Lx	Inhalt der csv-Zeile aufgeteilt in die Spalten 0 bis x, ist der Inhalt leer, so wird der Inhalt der vorherigen gefüllten übernommen. Gleichzeitig wird eine neue Datei (altername_neu.txt) mit gefüllten Feldern geschrieben.

Austauschreihenfolge

- 1.Zählerwerte
- 2.erweiterte Zählwerte
- 3.Funktionen von innen nach aussen bzw. rechts nach links.
- 4.Textteile

Funktionalitäten

Funktionsblöcke werden mit % umschlossen. Darin enthaltene Funktionsblöcke werden geklammert.

z.B.: %SUB(YEAR(%T9%),YEAR(%T7%%T8%))%

wird nach dem Textaustausch %SUB(YEAR(.02.2007)-YEAR(78y))%

und ergibt 1929

oder %DATE(%T0%SUB(YEAR(%T9%),YEAR(%T4%%T5%)))%

wird nach dem Textaustausch %DATE(12.01.SUB(YEAR(.02.2007),YEAR(78y)))%

ergibt nach der 1. Funktionsauwertung %DATE(12.01.1929)%

und als Ergebnis 12 JAN 1929

Funktion	Bemerkung
TIME(text)	Der „text“ wird im Format HH:MM angezeigt Eingangsformate= 10.30 bzw 10:30 werden zu 10:30 Es erfolgt lediglich ein Tausch . gegen :
DATE(text)	Der „text“ wird im Format TT MMM JJJJ angezeigt Eingangsformat 01.04.2000 wird zu 01 APR 2000 Die Folge .MM. wird in MMM umgewandelt MMM entspricht den Gedcomvorgaben
YEAR(text)	Der „text“ wird im Format JJJJ angezeigt Eingangsformat 01.04.2000 wird zu 2000 78y 3m 5d wird zu 78
MONTH(text)	Der „text“ wird im Format mm angezeigt Eingangsformat 01.04.2000 wird zu 04 78y 3m 5d wird zu 3
DAY(text)	Der „text“ wird im Format dd angezeigt Eingangsformat 01.04.2000 wird zu 01 78y 3m 5d wird zu 5
SUB(A, B)	Berechnet a-b und gibt das Ergebnis als String zurück
ADD(A, B)	Berechnet a+b und gibt das Ergebnis als String zurück
COMP(a,b,Text 1, Text 2)	Wenn a == b ist übernehme Text 1, ansonsten Text 2
FORMAT(form,wert)	Formatierte Ausgabe Zahlenausgabe form entspricht der Formatierung aus Java, wobei %1\$ schon vorgegeben wird. Wert enthält die zu formatierende Zahl.
EMPTY(Text a, Text b, Text c)	Wenn der Text a leer ist übernehme den Text b , ansonsten Text c
NOTEMPTY(Text a, Text b, Text c)	Wenn der Text a nicht leer ist übernehme den Text b , ansonsten Text c
REPLACE(Quelle,Suchtext, neuer Text)	Ersetze in der Quelle alle Vorkommen von Suchtext durch „neuer Text“ REPLACE (%T5%, karl,hugo) ergibt wenn %T5% karl, meier enthält hugo, meier Der Suchtext kann auch reguläre Ausdrücke enthalten. Für die Gruppenbildung müssen eckige Klammern „[]“ verwendet werden.

SPLIT(csvText,index,trennzeichen)

CsvText enthält durch Komma getrennte angaben. Index zeigt auf den Teil, der genutzt werden soll .z.B.:

SPLIT(%T5%,1,,) ergibt wenn %T5% karl, meier enthält karl

SPLIT(%T5%,1,) ergibt wenn %T5% karl, meier enthält karl,