**Was wollen wir Testen?**

Die in diesem Dokument eingefügten Marker, sollen normalisiert werden. Normalisierung bedeutet, dass es u.U. vorkommen kann, dass Marker in mehrere Text-Elemente aufgeteilt werden. Zum Parsen eines Dokuments, müssen alle Felder aber in einem Text-Element zusammengefasst werden. Dieser Vorgang wird als Normalisierung bezeichnet.

Dabei muss es unerheblich sein, in *welcher* Formatierung der F*ließtext* und die Felder sind. Auch ein Feld, welches wie folgt formatiert ist, muss erkannt und u.U. zu einem Text-Element zusammengefasst werden:

Dabei wird die Formatierung des ersten Zeichens „**$**“als Formatierung für das restliche Feld übernommen!

Es dürfen durchaus auch gerne mal ein paar mehr Felder kommen:

Aber auch in einem Fließtext muss man mal das eine oder andere Felder einbauen, sonst spiegelt das ja nicht wirklich die Realität wieder. Aber was wäre das Leben ohne die Möglichkeit Entscheidungen treffen zu können!? Deswegen . Läuft, oder!? :-P

**Es stehen folgende Felder zur Verfügung:**

1. Es soll das Feld mit der ID 156 gedruckt werden
2. $(C:156:<>:2} Es überprüft werden, ob das Feld mit der ID 156 einen Wert hat. Sollte es einen Wert haben, dann wird die Action 2 (nächsten Platzhalter löschen) ausgeführt.
3. Hier handelt es sich um ein Textfeld, welches den Inhalt „FOO BAR“ ausdruckt.

Sollten Felder syntaktisch nicht gültig sein, wie z.B. ${FUBU oder {FA} oder $C156 ${C:12}, dann werden diese von der Normalisierung ausgenommen, blieben aber trotzdem im Dokument erhalten.

Darüber hinaus gibt sogenannten Blockfelder. Mit Hilfe dieser Felder können Text oder auch andere Felder entsprechende der Zeilen einer Ergebnismenge wiederholt werden.

Mit einem Block „0“ wird der Beginn und mit „1“ entsprechend das Ende markiert. Das Feld mit der ID 156 befindet sich zwischen den beiden Blockelementen und wird entsprechende der Einträge einer Ergebnisse, welche das Feld 156 enthalten, wiederholt.

**Grundsätzlich müssen bei der Deklarierung von Blöcke auf folgendes geachtet werden:**

1. Positionen der Start- und Endmarkierung eines Blocks

Der Start und das Ende eines Blockes müssen Geschwisterknoten sein. Befinden sich diese in unterschiedlichen Absätzen, so bezieht sich die Prüfung auf die Absätze. Schlägt die Prüfung fehl, dann werden diese Blockmarkierungen nicht weiter berücksichtigt.

Eine Verschachtelung von Blöcken ist nicht möglich!

**Textblock:**

**Absatzblock:**

1. Wiederholung der Knoten

Ist die unter 1) beschriebene Prüfung erfolgreich, dann werden allen Knoten, welche sich auf der selber Achse der Start- und Endmarkierung befinden, entsprechend der Ergebnismenge wiederholt.

1. Löschen der Markierungen

Grundsätzlich werden sowohl End- und Startmarkierung aus dem fertigen Dokument entfernt. Sollte sich innerhalb des Absatzes keine weiteren Inhalte befinden, dann wird der Absatz ebenfalls aus dem Dokument entfernt.

Achtung: Ein Leerzeichen wird als Inhalt betrachtet!

1. Verhalten von Blöcken innerhalb von Tabellen

Grundsätzlich verhalten sich Blöcke innerhalb von Zellen analog zu dem unter 1) beschriebenem Verhalten, sofern sich der Start- sowie der Endmarker in derselben Zelle befinden.

* 1. **Verhalten von Blöcken in unterschiedlichen Zellen**

Befinden sich Start- und Endmarker in verschiedenen Zellen, dann werden die Zellen, welche sich zwischen den Markierungen befinden, entsprechend den Zeilen der dazugehörigen Ergebnismenge wiederholt. Voraussetzung für dieses Verhalten ist, dass die markierten Zellen sich in einer Reihe befinden.

Dieses Verhalten kann u.U. zu einer konsistenten Tabellenstruktur führen. Um dies vermeiden, werden die Zellen in den verbleibenden Reihen angepasst.

**Beispiel:**

**Tabellenstruktur**

|  |  |  |  |
| --- | --- | --- | --- |
| **Marker** | **Kundenummer** | **Kundenname** | **Marker** |
|  |  |  |  |
| Inhalt1 | Inhalte 2 | Inhalt 3 | Inhalt 4 |

**Ergebnismenge**

|  |  |
| --- | --- |
| **156** | **175** |
| 12345 | Meyer |
| 67890 | Schmidt |

**Ergebnisstabelle**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Marker** | **Kundennummer** | **Kundenname** | **Marker** | | |
|  | 12345 | Meyer | 67890 | Schmidt |  |
| Inhalt1 | Inhalt2 | Inhalte 3 | Inhalte 4 | | |

Wie der Ergebnistabelle zu entnehmen ist, wurden die Einträge aus der Ergebnismenge der Zeile hinzugefügt und die verbleibenden Zeilen so angepasst, dass keine invalide Tabellenstruktur entsteht.

Zu beachten ist, dass das Löschen der Marker zwar zu einer Löschung des Absatzes führt, aber die Zelle als solches erhalten bleibt.

Diese Vorgehensweise kann zu unerwarteten Ergebnissen führen, da in den meisten Fällen die Anzahl der Zeilen in der Ergebnismenge nicht bekannt ist und damit auch die Anzahl der Zellen in der Ergebnistabelle, wird dieses Vorgehen nicht empfohlen.

* 1. **Verhalten von Blöcken in Zeilen**

Befinden sich Start- und Endmarker in verschiedenen Zeilen, dann werden die dazwischen liegenden Zeilen, entsprechend den Zeilen in der Ergebnismenge wiederholt. Zu beachten ist, dass es hier ein geändertes Verhalten bei der Löschung der Start- und Endmarker gibt.

Befindet sich den Zellen der Zeile, in welchem sich die Marker befinden keine Einträge, dann wird die komplette Zeile gelöscht. Sollten es Einträge geben, dann wird nur der Marker gelöscht. Der Zelle und Reihe bleiben erhalten.

Achtung: Leerzeichen bzw. Whitespaces werden auch als Inhalt betrachtet.

* + 1. **Beispiel**

|  |  |
| --- | --- |
| **Kundenummer** | **Kundenname** |
|  |  |
|  |  |
|  |  |

**Ergebnismenge**

|  |  |
| --- | --- |
| **156** | **175** |
| 12345 | Meyer |
| 67890 | Schmidt |

**Ergebnistabelle**

|  |  |
| --- | --- |
| **Kundenummer** | **Kundenname** |
| 12345 | Meyer |
| 6789 | Schmidt |

* + 1. **Beispiel**

|  |  |
| --- | --- |
| **Kundenummer** | **Kundenname** |
|  | Foo |
|  |  |
|  | Bar |

**Ergebnismenge**

|  |  |
| --- | --- |
| **156** | **175** |
| 12345 | Meyer |
| 67890 | Schmidt |

**Ergebnistabelle**

|  |  |
| --- | --- |
| **Kundenummer** | **Kundenname** |
|  | Foo |
| 12345 | Meyer |
| 6789 | Schmidt |
|  | Bar |

Der Test gilt als erfolgreich, wenn die folgenden Testtabellen validiert wurden

**Textfelder**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Anzahl** | **Id** | **Anzahl** | **Id** | **Anzahl** |
| 156 | 6 | 1752 | 1 | 1863 | 1 |
| 1121 | 1 | 1757 | 2 | 1860 | 1 |
| 1744 | 1 | 1760 | 1 | 1862 | 1 |
| 1745 | 1 | 1761 | 1 | 1749 | 1 |
| 1747 | 1 | 1762 | 2 | 1750 | 1 |
| 1748 | 1 | 1858 | 2 | 157 | 4 |
|  | **11** |  | **9** |  | **9** |

**Bedingungsfelder**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Anzahl** | **Exprssion** | **Action** |
| 1760 | 5 | <> | 0 |
| 1744 | 1 | = | 0 |
| 1752 | 1 | = | 0 |
| 1858 | 1 | = | 0 |
| 1757 | 1 | = | 0 |
| 1860 | 1 | = | 0 |
| 1862 | 1 | = | 0 |
| 1760 | 2 | = | 0 |
| 1761 | 1 | = | 0 |
|  | **14** |  |  |

**Textfelder**

|  |  |
| --- | --- |
| Anzahl | Text |
| 1 | FOO BAR |
| 1 | Hurray |
| 2 |  |

Blockfelder

|  |  |
| --- | --- |
| Anzahl | Typ |
| 6 | Start |
| 6 | End |
| 12 |  |

Insgesamt müssen es 57 Felder sein.

Dann mal viel Spaß beim Testen.  
Denken immer daran: Das was Du nicht findest, dann werden die DV’s auf alle Fälle finden