

Inhaltsverzeichnis

Tasko XML Formater.....	I
Erstellen eines leeren Reports.....	I
Reportname.....	I
Document-Tag.....	I
Report-Tag.....	2
Header-Tag.....	2
Footer-Tag.....	2
ReportAreas, Datenabfragen und Gruppierung.....	2
ReportArea-Tag.....	2
Alle Ergebnisse einer Datenmenge einbinden, allResults-Tag.....	3
SQL-Datenabfrage, sql-Tag.....	3
SortTable-Tag.....	3
Sortierung nach Gruppen, group-Tag.....	3
Tabellen.....	3
Tabellen, tableStart-Tag.....	3
Kopfzeile einer Tabelle, tr-Tag.....	4
Detail in Tabellen.....	4
Iterierbare Tags.....	4
Detail-Tags.....	4
Field-Tags.....	4
Valide Namen für Field-Tags.....	5
Text-Tags.....	5
QRCode-Tag.....	5
Anhang.....	5
Komplettes Beispiel.....	5

Tasko XML Formater

Ein neues Reportsystem auf Basis unserer JSON-Results, welche wir in Tasko zur Kommunikation zwischen Endgerät und Server verwenden.

Erstellen eines leeren Reports

Reportname

Ein Tag unter dem der Name des Reportes eingetragen wird. Unter diesem Namen finden Sie den Report in der Auswahl in der Reportkonfiguration. Dieses Tag wird nur zur Installation verwendet.

<reportName>Name-des-Reportes</reportName>

Document-Tag

Zwischen den Tags <document> und </document> wird der Aufbau des Reportes Beschrieben. So wird der einfachste Leere Report folgendermaßen beschrieben:

**<reportName>Versorgungsunterbrechung-Statistik-A3</reportName>
<document>
</document>**

Die Ausrichtung und Papiergröße wird eingestellt unter:

Report-Tag

Hier werden die Standard für den Report vorgegeben. Es muss alle Reportinformationen einschließen und ist notwendig. Parameter können übergeben werden.

Optionen:

orientation: Hoch- oder Querformat. Parameter: portrait oder landscape.

size: Stellt die Papiergröße ein. Parameter: A3 oder A4.

Beispiel:

```
<report orientation="landscape" size="A3">
```

```
</report>
```

Unser Report sieht nun so aus

```
<reportName>Versorgungsunterbrechung-Statistik-A3</reportName>
```

```
<document>
```

```
  <report orientation="landscape" size="A3">
```

```
  </report>
```

```
</document>
```

Nun wird nach dem Formatieren der Report mit leben gefüllt:

Header-Tag

Beschreibt, ob ein Standard Kopf eingebunden wird. Wenn kein Inhalt angegeben wird, wird der Standard Tasko Kopf verwendet

```
<header></header>
```

Footer-Tag

Beschreibt, ob eine Fußzeile eingebunden wird. Wenn Es leer ist, Wird die Standard-Fußzeile eingebunden

```
<footer></footer>
```

ReportAreas, Datenabfragen und Gruppierung

ReportArea-Tag

Notwendig. In diesem Bereich, werden die Daten eingebunden.

```
<ReportArea> </ReportArea>
```

Es können mehrere ReportAreas nebeneinander in einem Report existieren. Alle folgenden Tags befinden sich immer innerhalb einer ReportArea.

Wird eine ReportArea als Unterbericht verwendet, gibt es die Möglichkeit, key-value Paare zu übergeben, die in der SQL-Abfrage als Parameter verwendet werden:

<ReportArea keys = „key1,key2“ values = „value1,value2“> </ReportArea>

Dabei bezieht sich dann der erste key im keys-String auf den ersten Wert im values-String. In der dann folgenden SQL-Abfrage (siehe unten) kann man die values dann mit dem key ansteuern, indem man den key in Geschweiften Klammern {key1} in die sql-Abfrage schreibt.

Als value Werte kann man jeden Wert aus der vorangegangenen sql-Abfrage wählen, wie man das im field-tag (siehe unten) auch machen würde. Anbieten würde sich zum Beispiel eine ReportArea als Unterbericht in einem group-tag aufzurufen und dann den Wert als Parameter zu übergeben, über den im group-tag gruppiert wird.

Alle Ergebnisse einer Datenmenge einbinden, allResults-Tag

<allResults></allResults>

In diesem Bereich werden die Daten angezeigt, die aus der Datenbank gezogen werden. Mit dem Quellcode zwischen den beiden Tags wird angegeben, in welcher Art die Daten angezeigt werden.

SQL-Datenabfrage, sql-Tag

Dieses Tag befindet sich innerhalb des allResults-Bereiches.

<sql> ![CDATA[sqlERG]] </sql>

Mit dem sql-Tag wird die Datenabfrage für den allResults-Bereich eingebunden. Wenn die Datenabfrage nicht selbst geschrieben wird, kann man eine der vorgefertigten Abfragen benutzen:

sqlERG: Alle Ergebnisse von Aufgaben, die die Position 01 in der Berichtskonfiguration haben. Nicht abgefragt werden Datenpunkte die in der Konfiguration der Aufgaben unter Punkte eingestellt werden. Eignet sich im Vergleich zu sqlERGPkt besser für Messwerte o.Ä. mit einfachem Ergebnis.

SqlERGPkt: Fragt alle Ergebnisse inklusive Datenpunkten von den unter Position 01 eingebundenen Aufgaben ab. Eignet sich im Vergleich zu sqlERG besser für Wartungen oder Störungsmeldungen, die kompliziertere Ergebnisse haben.

sqlERG_notDone: Fragt nur Ergebnisse ab, die noch nicht erledigt sind.

sqlERG_Done: Nur Ergebnisse, die als erledigt markiert sind.

SqlTarget: dynamische Aufgaben.

sqlTarget_Done: erledigte dynamische Aufgaben.

sqlTarget_notDone: nicht erledigte dynamische Aufgaben.

SortTable-Tag

Hat trotz des Namens nichts mit Tabellen im Bericht zu tun, sondern nur mit der Datensatztable. Das Tag dient dem Sortieren von Datensätzen nach dem angegebenen **key**. Wird nach der Datenabfrage innerhalb des allResults-Tags angegeben.

<sortTable key="agb_name,erg_datum" order="asc"/>

Optionen:

key: Erwarteter Parameter: Name einer Spalte des verwendeten Datensatzes. Im angegeben Beispiel wird zuerst nach dem Namen der Aufgabe(agb_name) und dann nach dem Datum des

Ergebnisses sortiert.

order: Mögliche Parameter sind asc (aufsteigend sortiert) und desc (absteigend sortiert).

Sortierung nach Gruppen, group-Tag

Mehrere Datenzeilen des Datensatzes können, wenn sie den gleichen Wert in einer Spalte haben zu einer Gruppe zusammengeführt werden.

<group changeEvent =“agb_name“></group>

Der Inhalt innerhalb der beiden Tags wird über die angegebene Datengruppe iteriert. Im Beispiel wird also jede Datenzeile mit dem gleichen Aufgabennamen zwischen den beiden Tags eingebunden. Dann wird der Inhalt des Tags erneut für die nächste Gruppe von Aufgabennamen eingebunden.

ReportAreas (siehe oben) können in einer Gruppe eingebunden werden, um Unterberichte zu generieren.

Optionen:

changeEvent: Erwarteter Parameter: Name einer Spalte des Datensatzes.

StartOnNewPage: „true“, „false“ lässt jede Gruppe auf einer neuen Seite beginnen

Tabellen

Tabellen, tableStart-Tag

Der Tag tableStart erstellt eine Tabelle, die dann zwischen den beiden Tags

<tableStart format= „border=‘1‘ cellspacing=‘0‘ cellpadding=‘2‘ size=‘1‘ style = ‘margin:auto;width:100%;’ “ />

und **<tableEnd/>**

mit Daten gefüttert wird.

Optionen:

border: Erwarteter Parameter: ganze Zahl. Stellt die Dicke der Trennlinien der Tabelle ein.

Kopfzeile einer Tabelle, tr-Tag

<tr></tr>

Innerhalb der beiden Tags werden einfache Textfelder mit den Spaltenüberschriften angegeben. In diesen Textfeldern bietet es sich an die Option **col** zu verwenden, um die Spaltenbreite anzugeben.

Es können auch field-tags eingebunden werden, wobei allerdings der Wert der ersten Zeile der Abfrage genommen werden, da über den tr-tag nicht iteriert wird. Das ist also nur sinnvoll, wenn ein Wert über die Zeilen der gleiche ist.

Detail in Tabellen

<detail></detail>

In Tabellen wird der Inhalt, der mit einem Detail Tag zusammengefasst wird als eine Zeile ausgegeben. Zwischen den beiden Tags wird der Inhalt des Details angegeben, wobei dieser Inhalt über jede Zeile des verwendeten Datensatzes iteriert wird.

Iterierbare Tags

Detail-Tags

Iteration über eine Zeile des für dieses Detail eingebundenen Datensatzes

<detail></detail>

Unter einem Detail-Tag können einfache Reportelemente wie Textfelder eingebunden werden. Dabei werden diese eingebundenen Elemente über jede Zeile der eingebundenen Gruppe von Daten iteriert. Eingebunden werden kann zum Beispiel ein einfaches Textfeld oder ein field-Tag, das ein Textfeld aus einem Datenpunkt des Datensatzes erstellt.

Field-Tags

Ein Field-Tag gibt die Möglichkeit einen Datenpunkt aus dem Datensatz einzubinden. Befindet das Feld sich in einem Gruppen oder- oder Detail-Tag entsprechend iteriert. Zwischen die Tags kann außerdem immer Text geschrieben werden, der dann entsprechend angezeigt wird. Der Text wird vor dem Inhalt, der durch das name-Attribut angegeben wird angezeigt.

<field name="" " fontSize="" " textAlign="" "> </field>

Optionen:

name: durch diesen Wert wird der Inhalt des Feldes bestimmt. Als Parameter kann der Name eines Datenpunktes angegeben werden.

fontSize: Erfordert eine ganze Zahl. Standardeinstellung bei vielen Textverarbeitungsprogrammen ist 11.

textAlign: Mögliche Werte sind left, center, right.

backgroundColor: Optional. Stellt die Hintergrundfarbe dieses Feldes ein. Erwarteter Parameter: Farbe im gängigen HTML-Format.

picWidth: Optional (nur sinnvoll für Bilder). Erwarteter Parameter: ganze Zahl mit der Ergänzung px für Pixel (Bsp.: 120px).

picHeight: Optional (nur sinnvoll für Bilder). Erwarteter Parameter: ganze Zahl mit der Ergänzung px für Pixel (Bsp.: 120px).

pattern: Ist der Wert dieses Feldes ein Datum, so kann das Datum mit dieser Option formatiert werden:

d	Tag
M	Monat(Kürzel)
Y	Jahr(yyyy)
i	Minuten
H	Stunden
m	Monat(Zahl)
s	Sekunden
y	Jahr(yy)

Valide name-Werte für Field-Tags

Im Field-Tag wird durch ein Kennwort unter der Option **name** der Inhalt eines Feldes der

Datentabelle geholt. Möglich ist es die Daten eines Ergebnisses zu holen:

erg_value, erg_datum, erg_id

Oder die Daten der dazugehörigen Aufgabe:

agb_name, agb_id, agb_nextaufgabe (Datum, zu dem die Aufgabe das nächste Mal ansteht),

Oder die Daten der dazugehörenden Gruppe:

grp_name

Oder aber die Daten der einzelnen Datenpunkte eines Ergebnisses, die in der Aufgabenkonfiguration als Punkte angegeben werden. Dann muss der in der Konfiguration angegebene Name angegeben werden.

Text-Tags

Ein Text-Tag bindet einen einfachen Text ein. Wird er in einem Gruppen- oder Detail-Tag eingebunden, so wird der Text entsprechend iteriert.

<text fontSize=" " cols = " " textAlign = " " > <![CDATA[Beispieltext]]> </text>

Optionen:

fontSize: erwarteter Parameter: ganze Zahl. Gibt die Textgröße an.

Cols: Optional. Erwarteter Parameter: ganze Zahl. Wird in Tabellen verwendet um die Anzahl der Spalten anzugeben, über die das Textfeld geht.

TextAlign: Mögliche Werte sind left, center, right.

Anstelle des Beispieltextes wird der Text angegeben, der im Textfeld erscheinen soll.

QRCode-Tag

Bindet einen QRCode ein:

<QRCode></QRCode>

Mit diesem Qr-Code kann die entsprechende Aufgabe geöffnet werden.

Anhang

Komplettes Beispiel

<reportName>Stoerungsstatistik</reportName>

<document>

<report orientation="landscape">

<header></header>

<footer></footer>

<ReportArea>

<allResults>

<sql><![CDATA[ergSQL]]>

```

</sql>
<sortTable key="agb_name,erg_datum" order="asc"/>
    <text fontSize="7" cols="5"><![CDATA[<br><br>]]></text>
    <tableStart format="border='1' cellspacing='0' cellpadding='2' size='1'
    style='margin:auto;width:100%;'" />
    <tr>
        <text fontSize="8" cols="1" textAlign="center"><![
        CDATA[ID]]> </text>

        <text fontSize="8" cols="1" textAlign="center"><![CDATA[Datum]]></text>
        <text fontSize="8" cols="1" textAlign="center"><![CDATA[Start der
Störung]]></text>
        <text fontSize="8" cols="1" textAlign="center"><![CDATA[Ende der
Störung]]></text>
        <text fontSize="8" cols="1" textAlign="center"><![CDATA[Anzahl unterbrochener
Letztverbraucher]]></text>
        <text fontSize="8" cols="1" textAlign="center"><![CDATA[unterbrochene
Gesamtleistung[kW]]></text>
        <text fontSize="8" cols="1"
textAlign="center"><![CDATA[Störungsursache]]></text>
    </tr>
    <detail>
        <field name='erg_id' fontSize="8" textAlign="center" ></field>
        <field name='erg_datum' pattern="d-M" fontSize="8" textAlign="center" ></field>
        <field name='Versorgungsunterbrechung Uhrzeit " Anfang "' fontSize="8"
textAlign="center" ></field>
        <field name='Versorgungsunterbrechung Uhrzeit " Ende "' fontSize="8"
textAlign="center" > </field>
        <field name='Anzahl unterbrochener Letztverbraucher:' fontSize="8"
textAlign="center"> </field>
        <field name='Unterbrochene Gesamtleistung kw:' fontSize="8"
textAlign="center"> </field>
        <field name='Störungsursache:' fontSize="8" textAlign="center"> </field>
    </detail>

    <tableEnd/>
    <text fontSize="12" cols="1"><![CDATA[<br>]]></text>
</allResults>
</ReportArea>
</report>
</document>

```