

## UserAttributes im UPK

Die Module im UPK teilen sich eine Grundfunktionalität – hauptsächlich die Speicherung und Abrufung von Füllständen. Diese werden in den Standard-UserAttributes (siehe 1.1) festgelegt und können in jedem Modul verwendet werden.

Die Module, die als Trigger fungieren, können zusätzlich zu den Standard-UserAttributes und ihren eigenen die Trigger-UserAttributes (siehe 1.2) nutzen. Dort wird festgelegt, wann der Trigger die Funktion ausführen soll.

Alle als „default“ gekennzeichneten Werte der UserAttributes brauchen nicht gesetzt werden, sie sind es standardmäßig schon.

## Changelog

### V0.7.6

- Modultyp **dumptrigger** hinzugefügt (2.3)
- neu: **acceptedFillTypes** (dumptrigger, 2.3.1)
- neu: **revenuePerLiter** (dumptrigger, 2.3.2)
- neu: **revenuesPerLiter** (dumptrigger, 2.3.3)
- neu: **statName** (dumptrigger, 2.3.4)
- Fix für initialFillLevels (Standard-UserAttributes, 1.1.7)
- Beispielmод UPK\_DumpTriggerTest hinzugefügt

### V0.7.5

- Fix für convertFillTypes (Standard-UserAttributes, 1.1.3)
- Beispielmод UPK\_TipTriggerTest2 hinzugefügt

### V0.7.4

- neu: Fülltypen-Behandlung für money, void, sun, rain und temperature
- Beispielmод UPK\_ProcessorTest3 hinzugefügt

### V0.7.3

- neu: **Speichern und Laden von Füllständen**
- neu: **convertFillTypes** (Standard-UserAttributes, 1.1.3)
- Modultyp **emptytrigger** hinzugefügt (2.4)
- neu: **emptyFillTypes** (emptytrigger, 2.4.1)
- neu: **emptyLitersPerSecond** (emptytrigger, 2.4.2)
- neu: **revenuePerLiter** (emptytrigger, 2.4.3)
- neu: **revenuesPerLiter** (emptytrigger, 2.4.4)
- neu: **statName** (emptytrigger, 2.4.5)
- neu: **revenuePerLiter** (tiptrigger, 2.8.4)
- neu: **revenuesPerLiter** (tiptrigger, 2.8.5)
- neu: **statName** (tiptrigger, 2.8.6)
- Beispielmод UPK\_EmptyTriggerTest hinzugefügt

## V0.7.2

- Modultyp **filltrigger** hinzugefügt (2.6)
- neu: **fillType** (filltrigger, 2.6.1)
- neu: **fillLitersPerSecond** (filltrigger, 2.6.2)
- neu: **createFillType** (filltrigger, 2.6.3)
- neu: **pricePerLiter** (filltrigger, 2.6.4)
- neu: **statName** (filltrigger, 2.6.5)
- Beispielmod UPK\_FillTriggerTest hinzugefügt

## V0.7.1

- Modultyp **tiptrigger** hinzugefügt (2.8)
- neu: **acceptedFillTypes** (tiptrigger, 2.8.1)
- neu: **showNotAcceptedWarning** (tiptrigger, 2.8.2)
- neu: **showCapacityReachedWarning** (tiptrigger, 2.8.3)
- Beispielmod UPK\_TipTriggerTest hinzugefügt

## V0.7.0

- neue Art Füllstände zu verwalten

(Versionsnummern übersprungen)

## V0.1.3

- Modultyp **unspecified** hinzugefügt (2.9)
- geändert: **enableChildrenIfProcessing** (processor, 2.7.19)
- neu: **addIfProcessing** (processor, 2.7.16)
- neu: **emptyFillTypesIfProcessing** (processor, 2.7.17)
- neu: **enableChildrenIfNotProcessing** (processor, 2.7.20)
- neu: **disableChildrenIfProcessing** (processor, 2.7.21)
- neu: **disableChildrenIfNotProcessing** (processor, 2.7.17)
- umbenannt: von „equal“ zu „uniform“ in **outcomeVariationType** (procesor, 2.7.11)

# 1. Allgemeine UserAttributes

## 1.1. Standard-UserAttributes

### 1.1.1. type (string)

Gibt an, um welches Modul es sich handelt, d.h. welche Funktion es haben soll.

| Ausprägung     | Beschreibung |
|----------------|--------------|
| base           | siehe 2.1    |
| displaytrigger | siehe 2.2    |
| dumptrigger    | siehe 2.3    |
| emptytrigger   | siehe 2.4    |
| entitytrigger  | siehe 2.5    |
| filltrigger    | siehe 2.6    |
| processor      | siehe 2.7    |
| tiptrigger     | siehe 2.8    |
| unspecified    | siehe 2.9    |

### 1.1.2. adjustToTerrainHeight (boolean)

Dieses UserAttribute ist auch für alle Nodes oder TriggerGroups im Mod anwendbar, die kein UPK-Modul sind.

| Ausprägung      | Beschreibung  |
|-----------------|---|
| true            | Verschiebt den Node-Ursprung auf Höhe des Terrains. |
| false (default) | Verändert die Lage der Node nicht.                  |

### 1.1.3. convertFillTypes (string)

| Ausprägung                 | Beschreibung   |
|----------------------------|--|
| (ohne) (default)           | Keine Fülltypen werden umgewandelt   |
| (Auflistung von Fülltypen) | <p>Auflistung von Fülltypen, die jeweils in den zuerst aufgeführten umgewandelt werden sollen. Mehrere Auflistungen können durch Komma getrennt werden.</p> <p>Am besten in dem Modul verwenden, wo der Ziel-Fülltyp gespeichert wird (siehe store 1.1.4).</p> <p>Hinweis: Die umzuwandelnden Fülltypen müssen trotzdem bei Triggern als akzeptiert aufgeführt werden.</p> <p>Bsp: Jede Strohart soll als Weizenstroh gespeichert werden:<br/>         „wheat_windrow oat_windrow rice_windrow“</p> <p>Zudem soll Gras als Heu gespeichert werden:<br/>         „wheat_windrow oat_windrow rice_windrow, dryGrass_windrow grass_windrow“</p> |

#### 1.1.4. store (string)

Regelt die Speicherung und Verwaltung von Füllständen.

| Ausprägung                 | Beschreibung  |
|----------------------------|---|
| (ohne) (default)           | Das Modul greift auf den selben Speicher zu wie das übergeordnete Modul und nutzt dessen Einstellungen.<br>Nicht kombinierbar mit capacity oder capacities.<br>Die Basis (siehe 2.1) speichert standardmäßig jeden Fülltyp separat ab.  |
| (Auflistung von Fülltypen) | Die Speicherung erfolgt separat für jeden aufgelisteten Fülltyp.<br>Bei nicht aufgelisteten Fülltypen wird an das übergeordnete Modul weiterverwiesen.<br>Kombinierbar mit capacities (siehe 1.1.6)<br>Bsp: „wheat“, „barley water“, „manure wheat barley“  |
| single                     | Speichert nur einen einzigen Fülltyp ab. Sobald ein Füllstand gespeichert ist, ist auch dessen Fülltyp fixiert. Weitere Füllstände anderer Fülltypen können dann nicht mehr gespeichert werden.<br>Der Fülltyp kann sich erst ändern, wenn der Füllstand des vorherigen wieder leer ist.<br>Kombinierbar mit capacities (siehe 1.1.6) |
| fifo                       | „first in – first out“: Füllstände, die zuerst in dem Modul gespeichert werden, werden als erste wieder ausgegeben.<br>Funktion ähnlich zu einem Rohr: Wenn man Weizen, Gerste und Raps in ein Rohr kippt, kommt auf der anderen Seite Weizen, Gerste und Raps in dieser Reihenfolge wieder raus.                                     |
| filo                       | „first in – last out“: Füllstände, die zuerst in dem Modul gespeichert werden, werden als letzte wieder ausgegeben.<br>Funktion ähnlich zu einem Fass: Wenn man Weizen, Gerste und Raps in ein Fass kippt, muss man Raps, Gerste und Weizen in dieser Reihenfolge wieder entnehmen, um es zu leeren.                                  |

#### 1.1.5. capacity (float)

Legt den maximalen Standard-Füllstand fest.

Bei separater und einzelner Füllstandsspeicherung gilt dies für jeden Fülltyp, für fifo und filo nur insgesamt.

| Ausprägung       | Beschreibung                                      |
|------------------|---|
| (ohne) (default) | Die Füllmenge ist unendlich.                      |
| (Zahl)           | Maximaler Füllstands-Wert<br>Bsp: „30000“, „6500“ |

### 1.1.6. capacities (string)

| Ausprägung                                 | Beschreibung  |
|--|---|
| (ohne) (default)                           | Keine Ausnahmen für capacity (siehe 1.1.5)  |
| (Auflistung von Füllständen und Fülltypen) | Legt die Füllstände für verschiedene Fülltypen fest. Für alle hier nicht aufgelisteten Fülltypen, die gespeichert werden, gilt capacity (siehe 1.1.5).<br>Nicht in Kombination mit store="fifo" oder store="filo".<br>Bsp: „6000 wheat“, „5000 water 1000 manure“ |

### 1.1.7. initialFillLevels (string)

| Ausprägung                                 | Beschreibung   |
|--|--|
| (ohne) (default)                           | Alle Füllmengen sind beim Kauf des Mods/ zu Beginn des Spiels 0.   |
| (Auflistung von Füllständen und Fülltypen) | Setzt bestimmte Fülltypen einmalig beim Kauf/ beim Beginn des Spiels auf die angegebenen Werte.<br>Bsp: „500 wheat“, „300 water 60 manure“ |

### 1.1.8. isEnabled (boolean)

Legt fest, ob das Modul beim Kauf des Mods/ zu Beginn des Spiels aktiviert oder deaktiviert ist.

| Ausprägung     | Beschreibung              |
|----------------|---------------------------|
| true (default) | das Modul ist aktiviert   |
| false          | das Modul ist deaktiviert |

## 1.2. Trigger-UserAttributes

### 1.2.1. allowWalker (boolean)

| Ausprägung     | Beschreibung                                 |
|----------------|--|
| true (default) | der Trigger funktioniert für Fußgänger       |
| false          | der Trigger funktioniert nicht für Fußgänger |

### 1.2.2. allowMotorized (boolean)

| Ausprägung     | Beschreibung   |
|----------------|--|
| true (default) | der Trigger funktioniert für Fahrzeuge mit Motor       |
| false          | der Trigger funktioniert nicht für Fahrzeuge mit Motor |

### 1.2.3. allowCombine (boolean)

| Ausprägung     | Beschreibung                                      |
|----------------|---|
| true (default) | der Trigger funktioniert für Erntemaschinen       |
| false          | der Trigger funktioniert nicht für Erntemaschinen |

### 1.2.4. allowFillable (boolean)

Als Überkategorie für alle speziellen Anhänger.

| Ausprägung     | Beschreibung   |
|----------------|--|
| true (default) | der Trigger funktioniert für alle Anhänger   |
| false          | der Trigger funktioniert nur für diejenigen Anhänger, die mit dem entsprechenden UserAttribute auf true gesetzt sind |

Bsp um nur Ballenwickler und Ladewagen zu akzeptieren:

allowFillable=false  
allowBaler=true  
allowForageWagon=true

### 1.2.5. allowTipper (boolean)

| Ausprägung      | Beschreibung                              |
|-----------------|---|
| true            | der Trigger funktioniert für Kipper       |
| false (default) | der Trigger funktioniert nicht für Kipper |

### 1.2.6. allowShovel (boolean)

| Ausprägung      | Beschreibung                                 |
|-----------------|--|
| true            | der Trigger funktioniert für Schaufeln       |
| false (default) | der Trigger funktioniert nicht für Schaufeln |

**1.2.7. allowWaterTrailer (boolean)**

| Ausprägung      | Beschreibung                                   |
|-----------------|--|
| true            | der Trigger funktioniert für Wassertanks       |
| false (default) | der Trigger funktioniert nicht für Wassertanks |

**1.2.8. allowFuelTrailer (boolean)**

| Ausprägung      | Beschreibung                                       |
|-----------------|--|
| true            | der Trigger funktioniert für Treibstofftanks       |
| false (default) | der Trigger funktioniert nicht für Treibstofftanks |

**1.2.9. allowLiquidManureTrailer (boolean)**

| Ausprägung      | Beschreibung                                  |
|-----------------|---|
| true            | der Trigger funktioniert für Gülletanks       |
| false (default) | der Trigger funktioniert nicht für Gülletanks |

**1.2.10. allowMilkTrailer (boolean)**

| Ausprägung      | Beschreibung                                  |
|-----------------|---|
| true            | der Trigger funktioniert für Milchtanks       |
| false (default) | der Trigger funktioniert nicht für Milchtanks |

**1.2.11. allowSowingMachine (boolean)**

| Ausprägung      | Beschreibung                                   |
|-----------------|--|
| true            | der Trigger funktioniert für Sämaschinen       |
| false (default) | der Trigger funktioniert nicht für Sämaschinen |

**1.2.12. allowSprayer (boolean)**

| Ausprägung      | Beschreibung                                |
|-----------------|---|
| true            | der Trigger funktioniert für Spritzen       |
| false (default) | der Trigger funktioniert nicht für Spritzen |

**1.2.13. allowForageWagon (boolean)**

| Ausprägung      | Beschreibung                                 |
|-----------------|--|
| true            | der Trigger funktioniert für Ladewagen       |
| false (default) | der Trigger funktioniert nicht für Ladewagen |

**1.2.14. allowBaler (boolean)**

| <b>Ausprägung</b> | <b>Beschreibung</b>                                       |
|-------------------|---|
| true              | der Trigger funktioniert für Ballenpressen/-wickler       |
| false (default)   | der Trigger funktioniert nicht für Ballenpressen/-wickler |

**1.2.15. allowTrafficVehicle (boolean)**

| <b>Ausprägung</b> | <b>Beschreibung</b>  |
|-------------------|--|
| true              | der Trigger funktioniert für Wagen des automatischen Straßenverkehrs       |
| false (default)   | der Trigger funktioniert nicht für Wagen des automatischen Straßenverkehrs |



## 2. spezielle UserAttributes

### 2.1. base

Dieser Modultyp wird für das oberste Modul im Mod automatisch vergeben und kann nicht manuell verwendet werden. D.h. alle unten aufgeführten UserAttributes gelten nur für das oberste Modul im Mod.

#### 2.1.1. UPKversion (string)

| Ausprägung       | Beschreibung   |
|------------------|--|
| (ohne) (default) | keine Überprüfung der Versionsnummer   |
| (Zahl.Zahl.Zahl) | Mindest zu verwendende UPK-Version. Falls diese vom Nutzer unterschritten wird, wird ein Fehler angezeigt und der Mod nicht geladen.<br>Bsp: „0.1.1“ |

#### 2.1.2. modname (string)

| Ausprägung       | Beschreibung  |
|------------------|---|
| (ohne) (default) |   |
| (Name des Mods)  | Gibt den Namen des Mods bzw. des verwendeten Mod-Ordners an. Für verschiedene Dinge wichtig, z.Bsp. für Extraeintragungen in der ModDesc.xml. Groß- und Kleinschreibung beachten. |

## 2.2. displaytrigger

Für dieses Modul gelten auch die Trigger-UserAttributes (siehe 1.2).

Die Default-Werte der Trigger-UserAttributes sind folgende:

| UserAttribute  | Default-Wert |
|----------------|--------------|
| allowWalker    | true         |
| allowMotorized | true         |
| allowFillable  | true         |
| allowCombine   | true         |
| (andere)       | false        |

### 2.2.1. displayFillTypes (string)

| Ausprägung                   | Beschreibung  |
|------------------------------|---|
| (ohne) (default)             | Keine Füllstände werden angezeigt   |
| (Auflistung von Füllständen) | Eine Liste mit anzuzeigenden Füllsorten, durch Leerzeichen getrennt.<br>Bsp: „wheat“, „wheat barley“, „wheat barley water manure“ |

### 2.2.2. onlyFilled (boolean)

| Ausprägung     | Beschreibung  |
|----------------|---|
| true (default) | zeigt nur die angegebenen Füllsorten (displayFillTypes) an, wenn deren Füllstand größer als 0 Liter ist |
| false          | zeigt immer jeden Füllstand an, auch 0 Liter  |

### 2.2.3. showFillLevel (boolean)

| Ausprägung     | Beschreibung                     |
|----------------|----------------------------------|
| true (default) | zeigt den Füllstand in Litern an |
| false          | zeigt ihn nicht an               |

### 2.2.4. showPercentage (boolean)

| Ausprägung     | Beschreibung   |
|----------------|--|
| true (default) | zeigt den Füllstand in Relation zur Kapazität (capacity) in Prozent an |
| false          | zeigt ihn nicht an   |

## 2.3. dumptrigger

Dieser „Trigger“ ist ein flaches Objekt am Boden und fängt die Entladung von Schaufeln und Erntemaschinen auf.

Es ist kein Trigger im eigentlichen Sinn (da nichts in ihn reinfahren muss), weswegen nur die Standard-UserAttributes hinzukommen, nicht die für normale Trigger.

### 2.3.1. acceptedFillTypes (string)

| Ausprägung                   | Beschreibung   |
|------------------------------|--|
| (ohne) (default)             | Nichts wird akzeptiert.  |
| (Auflistung von Füllständen) | Eine Liste mit zu akzeptierenden Fülltypen.<br>Bsp: „wheat“, „wheat barley“, „wheat barley rape maize“ |

### 2.3.2. revenuePerLiter (float)

revenuePerLiter gilt für alle erlaubten Fruchttypen. Davon kann man mit revenuesPerLiter Ausnahmen festlegen bzw. gleich alle mit Preisen auflisten. Zuerst schaut das Skript ob in revenuesPerLiter ein Wert für den Fruchttyp festgelegt wurde, sonst nimmt es den Wert von revenuePerLiter (per default auf 0€ pro Liter gesetzt).

ZBsp.: acceptedFillTypes="wheat barley maize rape", revenuePerLiter="0.1", revenuesPerLiter="1.2 wheat 1.3 barley" und convertFillTypes="void wheat rape". Für Weizen bekommt man 1,2€ pro Liter und es wird gelöscht. Für Barley bekommt man 1,3€ pro Liter und es wird gespeichert. Für Mais bekommt man 0,1€ und es wird gespeichert. Für Raps bekommt man auch 0,1€ und es wird gelöscht.

| Ausprägung  | Beschreibung  |
|-------------|---|
| 0 (default) | Das Abladen ist bringt kein Geld ein.   |
| (Zahl)      | Das Abladen bringt X € pro Liter ein. Negative Werte werden zum Kontostand hinzugefügt.<br>Mit statName kombinierbar. |

### 2.3.3. revenuesPerLiter (string)

| Ausprägung                            | Beschreibung   |
|---------------------------------------|--|
| (ohne) (default)                      | Das Abladen ist bringt kein Geld ein bzw. es gilt pricePerLiter.   |
| (Auflistung von Kosten und Fülltypen) | Das Abladen bringt X € pro Liter ein. Negative Werte werden dem Kontostand hinzugefügt.<br>Mit statName kombinierbar.<br>Für nicht hier aufgeführte, aber im Trigger akzeptierte Fülltypen, gilt pricePerLiter.<br>Bsp: „1.2 wheat 1.3 barley“ |

#### 2.3.4. statName (string)

Falls das Abladen Geld einbringt (oder kostet), legt statName fest, auf welchen Punkt in der Spiel-Statistik der Betrag gebucht werden soll.

| <b>Ausprägung</b>     | <b>Beschreibung</b>      |
|-----------------------|--------------------------|
| „other“ (default)     | Sonstiges                |
| „newVehiclesCost“     | Neue Fahrzeuge           |
| „newAnimalsCost“      | Tierkosten               |
| „constructionCost“    | Baukosten                |
| „vehicleRunningCost“  | Betriebskosten Fahrzeuge |
| „propertyMaintenance“ | Betriebskosten Gebäude   |
| „wagePayment“         | Lohnzahlungen            |
| „harvestIncome“       | Einnahmen Ernte          |
| „missionIncome“       | Einnahmen Missionen      |
| „loanInterest“        | Kreditzinsen             |

## 2.4. emptytrigger

Dieser Trigger entleert Fahrzeuge und Anhänger ohne Animationen oder Geräusche derselben auszulösen wie die Abkip-Animation (siehe tiptrigger) oder Tankgeräusche (siehe gasstationtrigger). Dieser Trigger ist quasi das Gegenteil des filltriggers.

Die Default-Werte der Trigger-UserAttributes sind folgende:

| UserAttribute            | Default-Wert                       |
|--------------------------|------------------------------------|
| allowTipper              | true                               |
| allowShovel              | true                               |
| allowSowingMachine       | true, wenn filltype="seeds"        |
| allowWaterTrailer        | true, wenn filltype="water"        |
| allowMilkTrailer         | true, wenn filltype="milk"         |
| allowLiquidManureTrailer | true, wenn filltype="liquidManure" |
| allowSprayer             | true, wenn filltype="fertilizer"   |
| allowFuelTrailer         | true, wenn filltype="fuel"         |
| allowMotorized           | false                              |
| (andere)                 | false                              |

### 2.4.1. emptyFillTypes (string)

| Ausprägung                 | Beschreibung   |
|----------------------------|--|
| (ohne) (default)           | Nichts wird abgefüllt.   |
| (Auflistung von Fülltypen) | Auflistung von Fülltypen, die geleert werden sollen.<br>Bsp: „wheat“, „barley maize“, „water fuel wheat“ |

### 2.4.2. emptyLitersPerSecond (float)

| Ausprägung     | Beschreibung   |
|----------------|--|
| 1500 (default) | Die Entleerungsgeschwindigkeit ist 1500 Liter pro Sekunde.     |
| (Zahl)         | Legt die Entleerungsgeschwindigkeit in Liter pro Sekunde fest. |

### 2.4.3. revenuePerLiter (float)

revenuePerLiter gilt für alle erlaubten Fruchttypen. Davon kann man mit revenuesPerLiter Ausnahmen festlegen bzw. gleich alle mit Preisen auflisten. Zuerst schaut das Skript ob in revenuesPerLiter ein Wert für den Fruchttyp festgelegt wurde, sonst nimmt es den Wert von revenuePerLiter (per default auf 0€ pro Liter gesetzt).

ZBsp.: acceptedFillTypes="wheat barley maize rape", revenuePerLiter="0.1", revenuesPerLiter="1.2 wheat 1.3 barley" und convertFillTypes="void wheat rape".  
Für Weizen bekommt man 1,2€ pro Liter und es wird gelöscht. Für Barley bekommt man 1,3€ pro Liter und es wird gespeichert. Für Mais bekommt man 0,1€ und es wird gespeichert. Für Raps bekommt man auch 0,1€ und es wird gelöscht.

| Ausprägung  | Beschreibung   |
|-------------|--|
| 0 (default) | Die Entleerung ist ohne Gewinn.  |
| (Zahl)      | Die Entleerung bringt X € pro Liter ein. Negative Werte werden zum Kontostand hinzugefügt.<br>Mit statName kombinierbar. |

#### 2.4.4. revenuesPerLiter (string)

| Ausprägung                            | Beschreibung  |
|---------------------------------------|---|
| (ohne) (default)                      | Die Entleerung ist kostenlos bzw. es gilt pricePerLiter.  |
| (Auflistung von Kosten und Fülltypen) | Die Entleerung kostet X € pro Liter. Negative Werte werden dem Kontostand hinzugefügt.<br>Mit statName kombinierbar.<br>Für nicht hier aufgeführte, aber im Trigger akzeptierte Fülltypen, gilt pricePerLiter.<br>Bsp: „1.2 wheat 1.3 barley“ |

#### 2.4.5. statName (string)

Falls die Abfüllung Geld kostet (oder einbringt), legt statName fest, auf welchen Punkt in der Spiel-Statistik der Betrag gebucht werden soll.

| Ausprägung            | Beschreibung             |
|-----------------------|--------------------------|
| „other“ (default)     | Sonstiges                |
| „newVehiclesCost“     | Neue Fahrzeuge           |
| „newAnimalsCost“      | Tierkosten               |
| „constructionCost“    | Baukosten                |
| „vehicleRunningCost“  | Betriebskosten Fahrzeuge |
| „propertyMaintenance“ | Betriebskosten Gebäude   |
| „wagePayment“         | Lohnzahlungen            |
| „harvestIncome“       | Einnahmen Ernte          |
| „missionIncome“       | Einnahmen Missionen      |
| „loanInterest“        | Kreditzinsen             |

## 2.5. entitytrigger

Aktiviert und deaktiviert die untergeordneten Module.

Zum Beispiel um eine Animation, wie eine Schranke, abzuspielen, wenn sich ein Fahrzeug davor befindet.

Die Default-Werte der Trigger-UserAttributes sind folgende:

| UserAttribute  | Default-Wert |
|----------------|--------------|
| allowWalker    | true         |
| allowMotorized | true         |
| allowFillable  | true         |
| allowCombine   | true         |
| (andere)       | false        |

### 2.5.1. enableOnEmpty (boolean)

| Ausprägung      | Beschreibung  |
|-----------------|---|
| true            | aktiviert die untergeordneten Module wenn sich nichts im Trigger befindet |
| false (default) | aktiviert die untergeordneten Module wenn sich etwas im Trigger befindet  |

## 2.6. filltrigger

Dieser Trigger befüllt Fahrzeuge und Anhänger ohne Animationen oder Geräusche derselben auszulösen wie die Tankgeräusche (siehe gasstationtrigger). Dieser Trigger ist quasi das Gegenteil des emptytriggers.

Die Default-Werte der Trigger-UserAttributes sind folgende:

| UserAttribute            | Default-Wert                       |
|--------------------------|------------------------------------|
| allowTipper              | true                               |
| allowShovel              | true                               |
| allowSowingMachine       | true, wenn filltype="seeds"        |
| allowWaterTrailer        | true, wenn filltype="water"        |
| allowMilkTrailer         | true, wenn filltype="milk"         |
| allowLiquidManureTrailer | true, wenn filltype="liquidManure" |
| allowSprayer             | true, wenn filltype="fertilizer"   |
| allowFuelTrailer         | true, wenn filltype="fuel"         |
| allowMotorized           | false                              |
| (andere)                 | false                              |

### 2.6.1. fillType (string)

| Ausprägung       | Beschreibung   |
|------------------|--|
| (ohne) (default) | Nichts wird abgefüllt.   |
| (Fülltyp)        | Der Fülltyp, der abgefüllt werden soll.<br>Bsp: „wheat“, „barley“, „water“ |

### 2.6.2. fillLitersPerSecond (float)

| Ausprägung     | Beschreibung  |
|----------------|---|
| 1500 (default) | Die Befüllgeschwindigkeit ist 1500 Liter pro Sekunde.     |
| (Zahl)         | Legt die Befüllgeschwindigkeit in Liter pro Sekunde fest. |

### 2.6.3. createFillType (boolean)

| Ausprägung      | Beschreibung  |
|-----------------|---|
| true            | Legt fest, dass die Befüllung unabhängig vom Füllstand stattfindet. |
| false (default) | Legt fest, dass die abgefüllte Füllmenge vorhanden sein muss.       |

### 2.6.4. pricePerLiter (float)

| Ausprägung  | Beschreibung  |
|-------------|---|
| 0 (default) | Die Befüllung ist kostenlos.  |
| (Zahl)      | Die Befüllung kostet X € pro Liter. Negative Werte werden zum Kontostand hinzugefügt.<br>Mit statName kombinierbar. |



### 2.6.5. statName (string)

Falls die Abfüllung Geld kostet (oder einbringt), legt statName fest, auf welchen Punkt in der Spiel-Statistik der Betrag gebucht werden soll.

| Ausprägung            | Beschreibung             |
|-----------------------|--------------------------|
| „other“ (default)     | Sonstiges                |
| „newVehiclesCost“     | Neue Fahrzeuge           |
| „newAnimalsCost“      | Tierkosten               |
| „constructionCost“    | Baukosten                |
| „vehicleRunningCost“  | Betriebskosten Fahrzeuge |
| „propertyMaintenance“ | Betriebskosten Gebäude   |
| „wagePayment“         | Lohnzahlungen            |
| „harvestIncome“       | Einnahmen Ernte          |
| „missionIncome“       | Einnahmen Missionen      |
| „loanInterest“        | Kreditzinsen             |

## 2.7. processor

Das Kernstück des UPK. Es erzeugt Güter bzw. Früchte, wenn gewünscht verbraucht es dafür andere.

Der Referenzwert ist 1 Liter des product (siehe), zu dem alle anderen Mengenangaben im Verhältnis stehen.

### 2.7.1. product (string)

Legt den Fülltyp der Produktion fest.

Bsp: „wheat“, „barley“

### 2.7.2. recipe (string)

| Ausprägung                                 | Beschreibung   |
|--|--|
| (ohne) (default)                           | Keine Füllstände werden bei der Produktion verbraucht.   |
| (Auflistung von Füllständen und Fülltypen) | Legt die verbrauchte Mengen an Fülltypen fest, die für 1 Liter des Produkts verbraucht werden sollen.<br>Beeinflusst die produzierte Menge in einem Produktionszyklus.<br>Bsp: „2 water“, „5 manure 100 barley 30 water“ |

### 2.7.3. byproducts (string)

Gibt an, welche Mengen von anderen Fruchttypen noch produziert werden sollen.

| Ausprägung                                 | Beschreibung  |
|--|---|
| (ohne) (default)                           | Es wird neben dem Produkt nichts weiter produziert.   |
| (Auflistung von Füllständen und Fülltypen) | Zusätzlich und in Relation zu einem Liter des Produkts hergestellte Füllmengen.<br>Bsp: „2 water“, „5 manure 100 barley 30 water“ |

### 2.7.4. onlyWholeProducts (boolean)

Ob das angegebene Produkt nur als Ganzes produziert werden soll. Die byproducts sind davon nicht betroffen.

| Ausprägung      | Beschreibung   |
|-----------------|--|
| true            | Das Produkt wird nur in ganzzahliger Stückmenge produziert.<br>Anfallende Reste werden beim nächsten Produktionszyklus dazugerechnet.<br>Z. Bsp. für alle Dinge wie Tiere, Flaschen, Ballen etc. |
| false (default) | Das Produkt wird auch mit Nachkommasetellen dem Füllstand dazugerechnet.<br>Z. Bsp. für alle Volumenangaben wie Weizen, Wasser, Dünger etc.  |

### 2.7.5. productionPrerequisite (string)

Legt die Voraussetzungen fest, wann 1 Liter des Produkts produziert werden kann.

| Ausprägung                                 | Beschreibung  |
|--|---|
| (ohne) (default)                           | Diese Bedingung wird nicht angewandt.   |
| (Auflistung von Füllständen und Fülltypen) | Zusätzlich und in Relation zu einem Liter des Produkts benötigte Füllmengen. Diese Füllstände werden nicht verbraucht. Beeinflusst die produzierte Menge in einem Produktionszyklus. Bsp: „2 water“, „5 manure 100 barley 30 water“ |

### 2.7.6. productsPerDay (float)

| Ausprägung  | Beschreibung  |
|-------------|---|
| 0 (default) | nichts wird beim Tageswechsel produziert.   |
| (Zahl)      | Legt die Menge des Produkt fest, die beim Tageswechsel produziert werden soll. Beeinflusst, ob ein Produktionszyklus ausgeführt wird. Nicht kombinierbar mit productsPerHour, productsPerMinute und productsPerSecond |

### 2.7.7. productsPerHour (float)

| Ausprägung  | Beschreibung   |
|-------------|--|
| 0 (default) | nichts wird beim Stundenwechsel produziert.  |
| (Zahl)      | Legt die Menge des Produkt fest, die beim Stundenwechsel produziert werden soll. Beeinflusst, ob ein Produktionszyklus ausgeführt wird. Nicht kombinierbar mit productsPerDay, productsPerMinute und productsPerSecond |

### 2.7.8. productsPerMinute (float)

| Ausprägung  | Beschreibung   |
|-------------|--|
| 0 (default) | nichts wird beim Minutenwechsel produziert.  |
| (Zahl)      | Legt die Menge des Produkt fest, die beim Minutenwechsel produziert werden soll. Beeinflusst, ob ein Produktionszyklus ausgeführt wird. Nicht kombinierbar mit productsPerDay, productsPerHour und productsPerSecond |

### 2.7.9. productsPerSecond (float)

| Ausprägung  | Beschreibung   |
|-------------|--|
| 0 (default) | nichts wird beim Sekundenwechsel produziert.   |
| (Zahl)      | Legt die Menge des Produkt fest, die jede Sekunde (Echtzeit) produziert werden soll.<br>Beeinflusst, ob ein Produktionszyklus ausgeführt wird.<br>Nicht kombinierbar mit productsPerDay, productsPerHour und productsPerMinute |

### 2.7.10. outcomeVariation (float)

| Ausprägung                   | Beschreibung   |
|------------------------------|--|
| 0 (default)                  | Die produzierte Füllmenge schwankt nicht zufällig.   |
| (Kommazahl zwischen 0 und 1) | Legt den Anteil fest, um den die Produktion schwanken soll.<br>Beeinflusst die produzierte Menge in einem Produktionszyklus.<br>Z.Bsp. „0.05“ entspricht 5% Schwankung nach oben und unten |

### 2.7.11. outcomeVariationType (string)

Legt das Zufallsverfahren für die Produktionsschwankung fest.  
Nur in Kombination mit outcomeVariation.

| Ausprägung        | Beschreibung                                  |
|-------------------|---|
| uniform (default) | Die Produktionsschwankung ist gleichverteilt. |
| normal            | Die Produktionsschwankung ist normalverteilt. |

### 2.7.12. productionHours (string)

Zu welchen Uhrzeiten produziert werden soll.  
Nicht in Kombination mit productionPerDay.

| Ausprägung             | Beschreibung  |
|------------------------|---|
| 0-23 (default)         | Zu jeder Uhrzeit wird produziert  |
| (Zahl-Zahl, Zahl-Zahl) | Legt die Uhrzeiten in Stunden fest, in denen produziert werden soll.<br>Beeinflusst, ob ein Produktionszyklus ausgeführt wird.<br>Bsp: „7-18“, „7-12, 14-16, 18-21“ |

### 2.7.13. productionInterval (float)

| Ausprägung  | Beschreibung   |
|-------------|--|
| 1 (default) | In jedem Zyklus wird produziert.   |
| (Zahl)      | Legt die Produktionszyklen fest, bei denen produziert werden soll. Zu lesen als „jeder x-ter Zyklus“.<br>Beeinflusst, ob ein Produktionszyklus ausgeführt wird.<br>Bsp: „2“, „5“ |

#### 2.7.14. productionProbability (float)

| Ausprägung                   | Beschreibung   |
|------------------------------|--|
| 1 (default)                  | Es wird immer produziert   |
| (Kommazahl zwischen 0 und 1) | Legt die Wahrscheinlichkeit fest, ob produziert werden soll. Beeinflusst, ob ein Produktionszyklus ausgeführt wird. Bsp: „0.5“, „0.2“ heißt, dass in 50% bzw. 20% der Fälle produziert wird. |

#### 2.7.15. addIfProcessing (boolean)

| Ausprägung                                 | Beschreibung   |
|--|--|
| (ohne) (default)                           | Es wird nichts hinzugefügt   |
| (Auflistung von Füllständen und Fülltypen) | Fügt bestimmte Füllstände hinzu, wenn bei einem Produktionszyklus etwas produziert wurde. Im Unterschied zu product oder byproducts ist diese hinzugefügte Füllmenge von der Menge der produzierten Güter unabhängig. Bsp: „1 counter“, „5 manure 100 barley 30 water“ |

#### 2.7.16. addIfNotProcessing (boolean)

| Ausprägung                                 | Beschreibung   |
|--|--|
| (ohne) (default)                           | Es wird nichts hinzugefügt   |
| (Auflistung von Füllständen und Fülltypen) | Fügt bestimmte Füllstände hinzu, wenn bei einem Produktionszyklus nichts produziert wurde (z.B. weil productionPrerequisite (siehe 2.7.5) nicht erfüllt wurde). Bsp: „1 counter“, „5 manure 100 barley 30 water“ |

#### 2.7.17. emptyFillTypesIfProcessing (boolean)

| Ausprägung                 | Beschreibung   |
|----------------------------|--|
| (ohne) (default)           | Es wird nichts hinzugefügt   |
| (Auflistung von Fülltypen) | Leert bestimmte Füllstände, dh. setzt sie auf 0, wenn etwas produziert wird. Bspw. um einen Zähler der unproduktiven Zyklen zurückzusetzen. Bsp: „counter“, „wheat barley“ |

### 2.7.18. emptyFillTypesIfNotProcessing (boolean)

| Ausprägung                 | Beschreibung  |
|----------------------------|---|
| (ohne) (default)           | Es wird nichts hinzugefügt  |
| (Auflistung von Fülltypen) | Leert bestimmte Füllstände, dh. setzt sie auf 0, wenn nichts produziert wird.<br>Bspw. um einen Zähler der produktiven Zyklen zurückzusetzen.<br>Bsp: „counter“, „wheat barley“ |

### 2.7.19. enableChildrenIfProcessing (boolean)

| Ausprägung      | Beschreibung   |
|-----------------|--|
| true            | Die untergeordneten Module werden aktiviert, wenn etwas produziert wird, um bspw. eine Animation abzuspielen.<br>Nicht in Kombination mit disableChildrenIfProcessing. |
| false (default) | Die untergeordneten Module werden weder aktiviert noch deaktiviert.  |

### 2.7.20. enableChildrenIfNotProcessing (boolean)

| Ausprägung      | Beschreibung   |
|-----------------|--|
| true            | Die untergeordneten Module werden aktiviert, wenn in einem Produktionszyklus nichts produziert wird.<br>Nicht in Kombination mit disableChildrenIfNotProcessing. |
| false (default) | Die untergeordneten Module werden weder aktiviert noch deaktiviert.  |

### 2.7.21. disableChildrenIfProcessing (boolean)

| Ausprägung      | Beschreibung  |
|-----------------|---|
| true            | Die untergeordneten Module werden deaktiviert, wenn etwas produziert wird, um bspw. eine Animation abzuspielen.<br>Nicht in Kombination mit enableChildrenIfProcessing. |
| false (default) | Die untergeordneten Module werden weder aktiviert noch deaktiviert.   |

### 2.7.22. disableChildrenIfNotProcessing (boolean)

| Ausprägung      | Beschreibung  |
|-----------------|---|
| true            | Die untergeordneten Module werden deaktiviert, wenn in einem Produktionszyklus nichts produziert wird.<br>Nicht in Kombination mit enableChildrenIfNotProcessing. |
| false (default) | Die untergeordneten Module werden weder aktiviert noch deaktiviert.   |

## 2.8. tiptrigger

Dieser Trigger ermöglicht es Kipper zu entladen und deren Kipp-Animation abzuspielen.

Die Default-Werte der Trigger-UserAttributes sind folgende:

| UserAttribute | Default-Wert |
|---------------|--------------|
| allowTipper   | true         |
| (andere)      | false        |

### 2.8.1. acceptedFillTypes (string)

| Ausprägung                 | Beschreibung   |
|----------------------------|--|
| (ohne) (default)           | Es wird nichts akzeptiert                                      |
| (Auflistung von Fülltypen) | Akzeptiert bestimmte Füllsorten<br>Bsp: „rape“, „wheat barley“ |

### 2.8.2. showNotAcceptedWarning (boolean)

| Ausprägung     | Beschreibung   |
|----------------|--|
| true (default) | Es wird eine Warnung angezeigt, wenn der Fülltyp des Kippers vom tiptrigger nicht akzeptiert wird.<br>Der angezeigte Text kann im l10n-Abschnitt der modDesc.xml mit „notAcceptedHere“ angepasst werden. |
| false          | Es wird keine Warnung angezeigt.   |

### 2.8.3. showCapacityReachedWarning (boolean)

| Ausprägung     | Beschreibung  |
|----------------|---|
| true (default) | Es wird eine Warnung angezeigt, wenn die maximale Füllmenge der Füllsorte bereits erreicht ist.<br>Der angezeigte Text kann im l10n-Abschnitt der modDesc.xml mit „capacityReached“ angepasst werden. |
| false          | Es wird keine Warnung angezeigt.  |

### 2.8.4. revenuePerLiter (float)

revenuePerLiter gilt für alle erlaubten Fruchttypen. Davon kann man mit revenuesPerLiter Ausnahmen festlegen bzw. gleich alle mit Preisen auflisten. Zuerst schaut das Skript ob in revenuesPerLiter ein Wert für den Fruchttyp festgelegt wurde, sonst nimmt es den Wert von revenuePerLiter (per default auf 0€ pro Liter gesetzt).

ZBsp.: acceptedFillTypes="wheat barley maize rape", revenuePerLiter="0.1", revenuesPerLiter="1.2 wheat 1.3 barley" und convertFillTypes="void wheat rape".  
Für Weizen bekommt man 1,2€ pro Liter und es wird gelöscht. Für Barley bekommt

man 1,3€ pro Liter und es wird gespeichert. Für Mais bekommt man 0,1€ und es wird gespeichert. Für Raps bekommt man auch 0,1€ und es wird gelöscht.

| Ausprägung  | Beschreibung   |
|-------------|--|
| 0 (default) | Die Entleerung ist kostenlos.  |
| (Zahl)      | Die Entleerung kostet X € pro Liter. Negative Werte werden zum Kontostand hinzugefügt.<br>Mit statName kombinierbar. |

### 2.8.5. revenuesPerLiter (string)

| Ausprägung                            | Beschreibung  |
|---------------------------------------|---|
| (ohne) (default)                      | Die Entleerung ist kostenlos bzw. es gilt pricePerLiter.  |
| (Auflistung von Kosten und Fülltypen) | Die Entleerung kostet X € pro Liter. Negative Werte werden dem Kontostand hinzugefügt.<br>Mit statName kombinierbar.<br>Für nicht hier aufgeführte, aber im Trigger akzeptierte Fülltypen, gilt pricePerLiter.<br>Bsp: „1.2 wheat 1.3 barley“ |

### 2.8.6. statName (string)

Falls die Abfüllung Geld kostet (oder einbringt), legt statName fest, auf welchen Punkt in der Spiel-Statistik der Betrag gebucht werden soll.

| Ausprägung            | Beschreibung             |
|-----------------------|--------------------------|
| „other“ (default)     | Sonstiges                |
| „newVehiclesCost“     | Neue Fahrzeuge           |
| „newAnimalsCost“      | Tierkosten               |
| „constructionCost“    | Baukosten                |
| „vehicleRunningCost“  | Betriebskosten Fahrzeuge |
| „propertyMaintenance“ | Betriebskosten Gebäude   |
| „wagePayment“         | Lohnzahlungen            |
| „harvestIncome“       | Einnahmen Ernte          |
| „missionIncome“       | Einnahmen Missionen      |
| „loanInterest“        | Kreditzinsen             |



spezielle UserAttributes - unspecified

## **2.9. unspecified**

Dieses Modul unterstützt nur die Standard-Attribute (siehe 1.1). Hauptsächlich zur Speicherung von Füllständen und Verbindung von Modulen gedacht.