Advanced Weighing Station

XML Configuration Reference / XML-Konfigurationsreferenz

Version: 1.0.0.0 Author: SirJoki80 FS25 modDesc: 113

Diese Dokumentation basiert auf den tatsächlich in V1.0.0.0 registrierten und ausgewerteten XML-Pfaden. Sie beschreibt die Modder-Schnittstelle der Placeable-XML; automatisch erzeugte Savegame-Daten sind kein Bestandteil der manuellen Konfiguration. / This documentation is based on the XML paths actually registered and evaluated by V1.0.0.0. It documents the placeable XML modding interface; automatically generated savegame data is not part of manual configuration.

Schnellübersicht / Quick reference

XML-Bereich / XML area	Zweck / Purpose
<advancedWeighingStation>	Root-Konfiguration für optionale AWS-Funktionen / Root configuration for optional AWS features
<realTrafficLight(s)>	Physische Rot-/Gelb-/Grün-Signale / Physical red-yellow-green signals
<beaconLight(s)>	Zustandsabhängige Warnleuchten / State-dependent warning beacons
<barrierAnimation(s)>	Schrankenlogik und Automatik / Barrier logic and automation
<animations>	Placeable-Animationen für Schranken / Placeable animations for barriers
<driveOverScale>	Achs-/Überfahrwaage / Drive-over axle scale

Inhalt / Contents

Direkt zu den einzelnen XML-Bereichen. Seitenzahlen werden automatisch erzeugt; zusätzlich stehen dieselben Bereiche als echte PDF-Lesezeichen in der Seitenleiste zur Verfügung.

Deutsch 4

1. Grundintegration 4
2. Root-Tag 4
3. Separater Schalttrigger 4
4. Physische Ampeln 5
   - Einfache Ampel mit Rot/Grün 5
   - Dreifarbige Ampel 5
   - Mehrere Ampeln 5
5. Beacon-Lichter 6
   - Ein vorhandener Node 6
   - Externe Beacon-XML 6
   - activeWhen 6
   - Mehrere Beacon-Lichter 7
6. Schrankenanimation 7
   - Einfache Variante 7
   - Eigenes Verhalten 7
   - openWhen 7
   - closeWhen 7
   - Anfangszustand 8
   - Geschwindigkeit 8
   - Manuelle Notfallsteuerung und Reset 8
   - Mehrere Schranken 8
7. Placeable-Animationen für Schranken 8
8. Achs-/Überfahrwaage (Drive-over Scale) 9
   - Kurzform 9
   - Empfohlene strukturierte Form 9
9. Komplettbeispiel 11
10. Interne Savegame-Tags 12
11. Wichtige Hinweise 12

English 13

1. Basic integration 13

2. Root tag . . . 13

3. Separate power trigger . . . 13

4. Physical traffic lights . . . 13

Multiple traffic lights . . . 14

5. Beacon lights . . . 14

activeWhen . . . 14

Multiple beacons . . . 15

6. Barrier animation . . . 16

Simple form . . . 16

Custom behavior . . . 16

openWhen . . . 16

closeWhen . . . 16

Start state and speed . . . 16

Manual emergency control . . . 17

Multiple barriers . . . 17

7. Placeable animations for barriers . . . 17

8. Drive-over axle scale . . . 17

Short form . . . 18

Recommended structured form . . . 18

9. Internal savegame tags . . . 18

10. Key notes . . . 18

Deutsch

1. Grundintegration

Die Mod erweitert kompatible Placeables automatisch, wenn sie die Basegame-Specialization `weighingStation` verwenden. Für die normalen AWS-Funktionen ist deshalb nicht zwingend ein eigener `<advancedWeighingStation>`-Block nötig. Der Block wird erst benötigt, wenn zusätzliche Hardware oder Sonderverhalten konfiguriert werden sollen.

```
<placeable ...>
    ...
    <advancedWeighingStation/>
</placeable>
```

> Alle folgenden Beispiele befinden sich innerhalb von `<placeable> ... </placeable>`. Node-Referenzen sollten bevorzugt über `i3dMappings` aufgelöst werden.

2. Root-Tag

Der Root-Tag kann mehrere optionale Attribute direkt aufnehmen. Zusätzlich können darunter Kind-Tags für Ampeln, Beacon-Lichter, Schranken und die Drive-over-Konfiguration verwendet werden.

Attribut	Bedeutung
trafficLightMode	Ampelmodus: 2 = Rot/Grün, 3 = Rot/Gelb/Grün. Standard: 2.
lampCount	Alias für `trafficLightMode`. Ebenfalls 2 oder 3.
driveOverScale	`true` aktiviert den Achs-/Überfahrwaagenmodus. Standard: false.
toggleTrigger	Optionaler separater Trigger-Node zum Ein-/Ausschalten der Waage zu Fuß.
driveOverScaleScanWidth	Breite des Scanbereichs in Metern. Legacy-/Kurzform.
driveOverScaleScanHeight	Höhe des Scanbereichs in Metern. Legacy-/Kurzform.
driveOverScaleScanLength	Länge des Scanbereichs in Metern. Legacy-/Kurzform.

3. Separater Schalttrigger

Standardmäßig kann die Waage am normalen Waagentrigger zu Fuß ein- und ausgeschaltet werden. Für ein separates Bedienfeld kann ein eigener Trigger angegeben werden.

```
<advancedWeighingStation toggleTrigger="awsPowerTrigger"/>

<i3dMappings>
    <i3dMapping id="awsPowerTrigger" node="0>0|4|0"/>
</i3dMappings>
```

Der Wert ist ein Node-Index bzw. ein Mapping-Name. Ist kein separater Trigger definiert, bleibt die normale Interaktion am Waagentrigger erhalten.

4. Physische Ampeln

AWS kann echte Mesh-/Light-Nodes einer Placeable-Ampel schalten. Ampeln sind optional und funktionieren unabhängig von Schranken.

Einfache Ampel mit Rot/Grün

```
<advancedWeighingStation trafficLightMode="2">
    <realTrafficLight
        redNode="trafficRed"
        greenNode="trafficGreen"/>
</advancedWeighingStation>
```

Dreifarbige Ampel

```
<advancedWeighingStation trafficLightMode="3">
    <realTrafficLight
        redNode="trafficRed"
        yellowNode="trafficYellow"
        greenNode="trafficGreen"
        hideInactive="true"/>
</advancedWeighingStation>
```

Attribut	Bedeutung
redNode	Node für das rote Signal. Optional, wenn diese Farbe physisch nicht vorhanden ist.
yellowNode	Node für Gelb. Sobald ein yellowNode vorhanden ist, wird die Ampel als 3-Lampen-Ampel behandelt.
greenNode	Node für das grüne Signal.
trafficLightMode	2 = Rot/Grün, 3 = Rot/Gelb/Grün. Überschreibt für diese Ampel den Root-Wert.
lampCount	Alias für trafficLightMode.
hideInactive	Standard `true`. Inaktive Signal-Nodes werden unsichtbar geschaltet.

Mehrere Ampeln

```
<advancedWeighingStation trafficLightMode="3">
    <realTrafficLights>
        <realTrafficLight
            redNode="entryRed"
            yellowNode="entryYellow"
            greenNode="entryGreen"/>

        <realTrafficLight
            redNode="exitRed"
            yellowNode="exitYellow"
            greenNode="exitGreen"/>
    </realTrafficLights>
</advancedWeighingStation>
```

Alternativ werden auch mehrere direkt aufeinanderfolgende `<realTrafficLight>`-Tags unterstützt. Für gut lesbare XML empfiehlt sich bei mehreren Ampeln der Wrapper `<realTrafficLights>`.

5. Beacon-Lichter

Beacon-Lichter können abhängig vom Zustand der Waage ein- und ausgeblendet werden. Es kann entweder ein vorhandener Node verwendet oder optional eine externe Beacon-XML an einem Spawn-/Link-Node geladen werden.

Ein vorhandener Node

```
<advancedWeighingStation>
    <beaconLight
        node="warningBeacon"
        activeWhen="measuring"
        hideInactive="true"/>
</advancedWeighingStation>
```

Externe Beacon-XML

```
<advancedWeighingStation>
    <beaconLight
        node="beaconLink"
        filename="$data/shared/assets/beaconLights/lizard/beaconLight01.xml"
        activeWhen="moving,measuring"/>
</advancedWeighingStation>
```

Attribut	Bedeutung
node	Erforderlicher Link-/Spawn-Node bzw. vorhandener Beacon-Node.
filename	Optionaler Pfad zu einer externen Beacon-Light-XML. `$data`-Pfade werden unterstützt.
xmlFilename	Alias für `filename`.
activeWhen	Legt fest, in welchen AWS-Zuständen das Licht aktiv ist. Standard: `measuring`.
hideInactive	Standard true. Blendet den Node im inaktiven Zustand aus.

activeWhen

```
measuring = während der eigentlichen Messung
waiting   = Waage wartet auf einen gültigen/stabilen Messzustand
moving    = Fahrzeug bewegt sich auf/in der Waage
done      = Messung abgeschlossen
active    = waiting, moving, measuring, done oder Live-Loading
notIdle   = jeder Zustand außer IDLE/EMPTY
all       = immer aktiv
never     = nie aktiv
```

Mehrere Werte dürfen mit Komma, Semikolon, | oder + kombiniert werden. Beispiel:

```
<beaconLight node="warningBeacon" activeWhen="moving,measuring,done"/>
```

Akzeptierte Synonyme sind u. a. `measure`, `wait`, `move`, `finished`, `always`, `off`. Unbekannte Werte fallen auf `measuring` zurück.

Mehrere Beacon-Lichter

```
<advancedWeighingStation>
    <beaconLights>
        <beaconLight node="beacon01" activeWhen="moving"/>
        <beaconLight node="beacon02" activeWhen="measuring,done"/>
    </beaconLights>
</advancedWeighingStation>
```

6. Schrankenanimation

Eine vorhandene Placeable-Animation kann optional mit der Waage verbunden werden. Ampel und Schranke funktionieren unabhängig voneinander und können gleichzeitig verwendet werden.

Einfache Variante

```
<advancedWeighingStation barrierAnimation="toggleGate01"/>
```

- Schranke startet geschlossen.
- Erfolgreiche Wiegung: Animation läuft vorwärts.
- Waagentrigger leer: Animation läuft rückwärts.

Eigenes Verhalten

```
<advancedWeighingStation>
    <barrierAnimation
        name="toggleGate01"
        openWhen="measurementDone"
        closeWhen="triggerEmpty"
        startState="closed"/>
</advancedWeighingStation>
```

openWhen

```
measurementDone = nach erfolgreicher Wiegung
vehicleStopped  = sobald das Fahrzeug stabil steht
vehicleEntered  = beim Befahren der Waage
always          = dauerhaft geöffnet
never           = nicht automatisch öffnen
```

Zusätzlich akzeptiert der Parser Kurzformen wie `stopped`, `entered`, `triggerEntered`, `open` und `off`. Ungültige Werte fallen auf `measurementDone` zurück.

closeWhen

```
triggerEmpty    = sobald die Waage vollständig leer ist
afterDelay      = nach einer eingestellten Zeit
nextMeasurement = beim Start der nächsten Wiegung
vehicleMoving   = sobald sich das Fahrzeug wieder bewegt
never           = nicht automatisch schließen
```

```
<barrierAnimation
    name="toggleGate01"
    openWhen="measurementDone"
    closeWhen="afterDelay"
    closeDelay="5"/>
```

Synonyme wie `delay`, `measurementStart`, `moving` und `off` werden ebenfalls normalisiert.

Anfangszustand

```
startState="closed"
startState="open"
```

Für geöffnet werden zusätzlich `opened`, `true` und `1` akzeptiert. Alles andere wird als geschlossen behandelt. Bei `openWhen="always"` erzwingt AWS automatisch Startzustand offen und `closeWhen="never"`.

Geschwindigkeit

```
<barrierAnimation
    name="toggleGate01"
    openSpeed="1"
    closeSpeed="0.75"/>
```

1 entspricht normaler Animationsgeschwindigkeit. Werte werden intern auf mindestens 0.01 begrenzt.

Manuelle Notfallsteuerung und Reset

Konfigurierte Schranken können über die AWS-Notfallsteuerung manuell umgeschaltet werden. Das ältere Attribut `manualToggle` ist nur noch aus Kompatibilitätsgründen registriert und wird in V1.0.0.0 bewusst ignoriert; die Notfallsteuerung ist für konfigurierte Schranken immer verfügbar.

```
manualOverrideReset="triggerEmpty"
manualOverrideReset="nextMeasurement"
manualOverrideReset="manual"
```

- triggerEmpty: manueller Override endet, sobald die Waage leer ist.
- nextMeasurement: Override endet beim Start der nächsten Messung bzw. bei erneutem Befahren.
- manual: Override bleibt bestehen, bis er wieder manuell geändert wird.

Mehrere Schranken

```
<advancedWeighingStation>
    <barrierAnimations>
        <barrierAnimation
            name="entryGate"
            openWhen="vehicleEntered"
            closeWhen="vehicleStopped"
            startState="open"/>

        <barrierAnimation
            name="exitGate"
            openWhen="measurementDone"
            closeWhen="triggerEmpty"
            startState="closed"/>
    </barrierAnimations>
</advancedWeighingStation>
```

AWS unterstützt auch mehrere direkte `<barrierAnimation>`-Tags. Der Wrapper `<barrierAnimations>` ist für mehrere Schranken übersichtlicher. Derselbe Animationsname darf nicht doppelt referenziert werden; Duplikate werden ignoriert.

7. Placeable-Animationen für Schranken

Die Schrankenlogik benötigt eine passende Animation unter `<animations>`. AWS liest diese Animation selbst aus und interpoliert Rotation, Position und/oder Skalierung.

```xml
<animations>
    <animation name="toggleGate01">
        <part
            node="gate01"
            startTime="0"
            endTime="2"
            startRot="0 0 0"
            endRot="0 90 0"/>
    </animation>
</animations>
```

Attribut	Bedeutung
animation#name	Name der Animation. Muss exakt dem Namen aus barrierAnimation entsprechen.
part#node	Node, der animiert wird.
part#startTime	Startzeit des Parts in Sekunden. Standard 0.
part#endTime	Endzeit des Parts in Sekunden.
part#duration	Alternative zu endTime. Endzeit = startTime + duration.
part#startRot / endRot	Start-/Endrotation als drei Werte.
part#startTrans / endTrans	Start-/Endposition als drei Werte.
part#startScale / endScale	Start-/Endskalierung als drei Werte.

Fehlt bei Rotation/Translation/Scale eine Seite, verwendet AWS den aktuellen Zustand bzw. kopiert den vorhandenen Gegenwert. Laufende Schrankenanimationen können sauber in Gegenrichtung fortgesetzt werden.

8. Achs-/Überfahrwaage (Drive-over Scale)

Der Drive-over-Modus ist für Waagen gedacht, bei denen das Fahrzeug während der Überfahrt achsweise erfasst wird. Er kann als Root-Attribut oder als eigener Kind-Tag aktiviert werden.

Kurzform

```xml
<advancedWeighingStation
    driveOverScale="true"
    driveOverScaleScanWidth="10"
    driveOverScaleScanHeight="16"
    driveOverScaleScanLength="16"/>
```

Empfohlene strukturierte Form

```xml
<advancedWeighingStation>
    <driveOverScale
        enabled="true"
        scanWidth="10"
        scanHeight="16"
        scanLength="16"/>
</advancedWeighingStation>
```

Attribut	Bedeutung
enabled	Aktiviert den Drive-over-/Achswaagenmodus. Standard false; ist der Kind-Tag vorhanden und enabled fehlt, wird true verwendet.
scanWidth	Gesamtbreite des Scanbereichs in Metern.
scanHeight	Gesamthöhe des Scanbereichs in Metern.
scanLength	Gesamtlänge des Scanbereichs in Metern.

Standard-Scanbereich der V1.0.0.0: 10 m Breite × 16 m Höhe × 16 m Länge. Die Werte werden absolut ausgewertet und intern auf mindestens 0.10 m Gesamtgröße pro Achse begrenzt. Der Kind-Tag überschreibt die entsprechenden Root-Kurzattribute, wenn beide vorhanden sind.

9. Komplettbeispiel

```
<placeable ...>
    ...

    <advancedWeighingStation trafficLightMode="3" toggleTrigger="awsPowerTrigger">
        <realTrafficLights>
            <realTrafficLight
                redNode="entryRed"
                yellowNode="entryYellow"
                greenNode="entryGreen"/>
            <realTrafficLight
                redNode="exitRed"
                yellowNode="exitYellow"
                greenNode="exitGreen"/>
        </realTrafficLights>

        <beaconLights>
            <beaconLight
                node="warningBeacon"
                activeWhen="moving,measuring"/>
        </beaconLights>

        <barrierAnimations>
            <barrierAnimation
                name="entryGate"
                openWhen="vehicleEntered"
                closeWhen="vehicleStopped"
                startState="open"/>
            <barrierAnimation
                name="exitGate"
                openWhen="measurementDone"
                closeWhen="triggerEmpty"
                startState="closed"
                openSpeed="1"
                closeSpeed="0.8"/>
        </barrierAnimations>
    </advancedWeighingStation>

    <animations>
        <animation name="entryGate">
            <part node="entryGateNode" startTime="0" endTime="2"
                  startRot="0 0 0" endRot="0 90 0"/>
        </animation>
        <animation name="exitGate">
            <part node="exitGateNode" startTime="0" endTime="2"
                  startRot="0 0 0" endRot="0 -90 0"/>
        </animation>
    </animations>

    <i3dMappings>
        <i3dMapping id="awsPowerTrigger" node="0>0|4|0"/>
        <i3dMapping id="entryRed" node="0>0|5|0"/>
        <i3dMapping id="entryYellow" node="0>0|5|1"/>
        <i3dMapping id="entryGreen" node="0>0|5|2"/>
        <i3dMapping id="exitRed" node="0>0|6|0"/>
        <i3dMapping id="exitYellow" node="0>0|6|1"/>
        <i3dMapping id="exitGreen" node="0>0|6|2"/>
        <i3dMapping id="warningBeacon" node="0>0|7"/>
        <i3dMapping id="entryGateNode" node="0>0|8"/>
        <i3dMapping id="exitGateNode" node="0>0|9"/>
    </i3dMappings>
</placeable>
```

10. Interne Savegame-Tags

Die Specialization registriert zusätzlich Savegame-Pfade für Statistik- und Messdaten. Diese werden von AWS selbst geschrieben und gelesen und sind keine Modder-Konfiguration. Dazu gehören u. a. Benutzerstatistiken und der letzte Messdatensatz mit Werten wie Spieler-ID, Hof-ID, Brutto, Tara, Netto, Füllvolumen, Eintragstyp und Zeitstempel. Diese Bereiche sollten nicht in die normale Placeable-XML kopiert werden.

11. Wichtige Hinweise

- Animationsnamen müssen exakt übereinstimmen.
- Node-Namen möglichst über i3dMappings einbinden.
- Ampeln, Beacon-Lichter und Schranken sind vollständig optional und können unabhängig voneinander kombiniert werden.
- Ohne Zusatz-Tags funktioniert die normale Advanced Weighing Station weiterhin auf kompatiblen Basegame-/Mod-Waagen.
- Unbekannte Enum-Werte werden in vielen Fällen auf einen sicheren Standard zurückgesetzt statt einen Fehler zu erzeugen.
- Das Attribut manualToggle ist Legacy-Kompatibilität und steuert in V1.0.0.0 die Notfallbedienung nicht mehr.

English

1. Basic integration

The mod automatically extends compatible placeables that use the basegame `weighingStation` specialization. Therefore, the regular AWS features do not require a dedicated `<advancedWeighingStation>` block. The block is only needed for optional hardware and special behavior.

```
<placeable ...>
    ...
    <advancedWeighingStation/>
</placeable>
```

All examples below are placed inside `<placeable> ... </placeable>`. Node references should preferably be resolved through `i3dMappings`.

2. Root tag

Attribute	Meaning
trafficLightMode	Traffic light mode: 2 = red/green, 3 = red/yellow/green. Default: 2.
lampCount	Alias for `trafficLightMode`.
driveOverScale	`true` enables axle drive-over mode. Default: false.
toggleTrigger	Optional separate trigger node used to switch the scale on/off while on foot.
driveOverScaleScanWidth	Drive-over scan width in metres. Short/legacy form.
driveOverScaleScanHeight	Drive-over scan height in metres. Short/legacy form.
driveOverScaleScanLength	Drive-over scan length in metres. Short/legacy form.

3. Separate power trigger

```
<advancedWeighingStation toggleTrigger="awsPowerTrigger"/>

<i3dMappings>
    <i3dMapping id="awsPowerTrigger" node="0>0|4|0"/>
</i3dMappings>
```

Without a separate trigger, normal interaction at the weighing trigger remains available.

4. Physical traffic lights

```
<advancedWeighingStation trafficLightMode="3">
    <realTrafficLight
        redNode="trafficRed"
        yellowNode="trafficYellow"
        greenNode="trafficGreen"
        hideInactive="true"/>
</advancedWeighingStation>
```

Attribute	Meaning
redNode	Node for the red signal.
yellowNode	Node for yellow. If present, the light is treated as a three-lamp light.
greenNode	Node for the green signal.
trafficLightMode	2 = red/green, 3 = red/yellow/green.
lampCount	Alias for trafficLightMode.
hideInactive	Default true. Hides inactive signal nodes.

Multiple traffic lights

```
<advancedWeighingStation trafficLightMode="3">
    <realTrafficLights>
        <realTrafficLight redNode="entryRed" yellowNode="entryYellow" greenNode="entryGreen"/>
        <realTrafficLight redNode="exitRed" yellowNode="exitYellow" greenNode="exitGreen"/>
    </realTrafficLights>
</advancedWeighingStation>
```

Multiple direct `<realTrafficLight>` elements are supported as well. The wrapper is recommended for readability.

5. Beacon lights

```
<advancedWeighingStation>
    <beaconLight
        node="warningBeacon"
        activeWhen="moving,measuring"
        hideInactive="true"/>
</advancedWeighingStation>
```

Attribute	Meaning
node	Required link/spawn node or existing beacon node.
filename	Optional external beacon XML filename; $data paths are supported.
xmlFilename	Alias for filename.
activeWhen	Defines the AWS states in which the light is active. Default: measuring.
hideInactive	Default true; hides the node while inactive.

activeWhen

```
measuring = during the actual measurement
waiting   = waiting for a valid/stable weighing state
moving    = vehicle is moving on/in the scale
done      = measurement completed
active    = waiting, moving, measuring, done or live-loading
notIdle   = every state except IDLE/EMPTY
all       = always active
never     = never active
```

Multiple values can be separated by comma, semicolon, | or +. Accepted aliases include measure, wait, move, finished, always and off. Unknown values fall back to measuring.

Multiple beacons

```
<advancedWeighingStation>
    <beaconLights>
        <beaconLight node="beacon01" activeWhen="moving"/>
        <beaconLight node="beacon02" activeWhen="measuring,done"/>
    </beaconLights>
</advancedWeighingStation>
```

6. Barrier animation

An existing placeable animation can optionally be connected to the weighing station. Traffic lights and barriers work independently and may be used at the same time.

Simple form

```
<advancedWeighingStation barrierAnimation="toggleGate01"/>
```

- Barrier starts closed.
- Successful measurement: animation runs forward.
- Empty weighing trigger: animation runs in reverse.

Custom behavior

```
<advancedWeighingStation>
    <barrierAnimation
        name="toggleGate01"
        openWhen="measurementDone"
        closeWhen="triggerEmpty"
        startState="closed"/>
</advancedWeighingStation>
```

openWhen

```
measurementDone = after a successful measurement
vehicleStopped  = as soon as the vehicle is stable/stopped
vehicleEntered  = when the scale is entered
always          = permanently open
never           = never open automatically
```

closeWhen

```
triggerEmpty    = when the scale becomes completely empty
afterDelay      = after a configured delay
nextMeasurement = at the start of the next measurement
vehicleMoving   = when the vehicle starts moving again
never           = never close automatically
```

```
<barrierAnimation name="toggleGate01" openWhen="measurementDone"
    closeWhen="afterDelay" closeDelay="5"/>
```

Start state and speed

```
<barrierAnimation
    name="toggleGate01"
    startState="closed"
    openSpeed="1"
    closeSpeed="0.75"/>
```

Speed factor 1 means normal animation speed. Values are clamped to at least 0.01. With `openWhen="always"`, AWS forces the start state to open and closeWhen to never.

Manual emergency control

```
manualOverrideReset="triggerEmpty"
manualOverrideReset="nextMeasurement"
manualOverrideReset="manual"
```

The legacy `manualToggle` attribute is registered for compatibility but intentionally ignored in V1.0.0.0; emergency barrier input is always enabled for configured barriers.

Multiple barriers

```
<advancedWeighingStation>
    <barrierAnimations>
        <barrierAnimation name="entryGate" openWhen="vehicleEntered"
            closeWhen="vehicleStopped" startState="open"/>
        <barrierAnimation name="exitGate" openWhen="measurementDone"
            closeWhen="triggerEmpty" startState="closed"/>
    </barrierAnimations>
</advancedWeighingStation>
```

7. Placeable animations for barriers

```
<animations>
    <animation name="toggleGate01">
        <part
            node="gate01"
            startTime="0"
            endTime="2"
            startRot="0 0 0"
            endRot="0 90 0"/>
    </animation>
</animations>
```

Attribute	Meaning
animation#name	Animation name; must exactly match the barrier reference.
part#node	Animated node.
part#startTime	Part start time in seconds, default 0.
part#endTime	Part end time in seconds.
part#duration	Alternative to endTime; end = start + duration.
part#startRot / endRot	Start/end rotation.
part#startTrans / endTrans	Start/end translation.
part#startScale / endScale	Start/end scale.

8. Drive-over axle scale

Short form

```
<advancedWeighingStation
    driveOverScale="true"
    driveOverScaleScanWidth="10"
    driveOverScaleScanHeight="16"
    driveOverScaleScanLength="16"/>
```

Recommended structured form

```
<advancedWeighingStation>
    <driveOverScale
        enabled="true"
        scanWidth="10"
        scanHeight="16"
        scanLength="16"/>
</advancedWeighingStation>
```

Default V1.0.0.0 scan volume: 10 m width × 16 m height × 16 m length. Values are treated as absolute values and internally clamped to a minimum total dimension of 0.10 m. If both forms are present, the child element values override the root attributes.

9. Internal savegame tags

AWS also registers savegame XML paths for user statistics and the most recent measurement. These are generated and maintained internally and are not placeable configuration options. Do not copy those paths into the normal placeable XML.

10. Key notes

- Animation names must match exactly.
- Prefer i3dMappings for node references.
- Traffic lights, beacons and barriers are completely optional and can be combined independently.
- Without additional tags, AWS still works on compatible basegame/mod weighing stations.
- Unknown enum-like values usually fall back to safe defaults.
- manualToggle is legacy compatibility only in V1.0.0.0.