

## Contents

<b>1</b>	<b>German/Deutsch</b>	<b>1</b>
1.1	Einbau in Fahrzeuge . . . . .	1
1.1.1	Einleitung . . . . .	1
1.1.2	Voraussetzungen und Vorbereitung . . . . .	1
1.1.3	Einbau der Echten Arbeitskamas . . . . .	1
1.1.4	Aufbau der XML-Einträge . . . . .	2
1.1.5	Kameragruppen . . . . .	3
1.1.6	Kameras . . . . .	4
1.1.7	Hide- und ShowNodes . . . . .	4
1.1.8	Assets . . . . .	5
1.1.9	Testen des Mods . . . . .	5
1.2	Einbau in Fahrzeuge – Erweiterte Optionen . . . . .	6
1.2.1	Konfigurationsabhängige Kameras und Displays . . . . .	6
1.2.2	Kameradisplays in dashboardCompounds . . . . .	6
1.2.3	AttachableCams mit zugehörigem Display . . . . .	8
<b>2</b>	<b>English</b>	<b>11</b>
2.1	Installation in Vehicles . . . . .	11
2.1.1	Introduction . . . . .	11
2.1.2	Requirements and Preparation . . . . .	11
2.1.3	Installing the Real Work Cameras . . . . .	11
2.1.4	Structure of the XML Entries . . . . .	12
2.1.5	Camera Groups . . . . .	13
2.1.6	Kameras . . . . .	14
2.1.7	Hide and Show Nodes . . . . .	14
2.1.8	Assets . . . . .	14
2.1.9	Testing the Mod . . . . .	15
2.2	Installation in Vehicles – Advanced Options . . . . .	15
2.2.1	Configuration-dependent Cameras and Displays . . . . .	15
2.2.2	Camera Displays in dashboardCompounds . . . . .	15
2.2.3	AttachableCams with Matching Display . . . . .	17



## 1 German/Deutsch

### 1.1 Einbau in Fahrzeuge

#### 1.1.1 Einleitung

Diese Anleitung beschreibt, wie ein Mod-Fahrzeug für die Nutzung des Real Work Cameras (Echte Arbeitskamas) Mods im LS25 vorbereitet wird. Der Mod stellt ein Skript bereit, das den Einbau von Kameras und Displays in einen Schlepper erleichtert.

#### 1.1.2 Voraussetzungen und Vorbereitung

Für den Einbau benötigst du das Skript aus dem Real Work Cameras Mod sowie die aktuelle Version des GIANTS Editors für LS25. Den Editor kannst du über die offizielle Website herunterladen: <https://gdn.giants-software.com/index.php>.

Die erforderlichen Skripte findest du im „sdk“-Ordner des Real Work Cameras Mods. Kopiere den darin enthaltenen Ordner „realWorkCamerasSetup“ in das „scripts“-Verzeichnis des GIANTS Editor-Installationsordners. Standardmäßig ist der Editor unter C:/Programme/GIANTS Software/GIANTS Editor 10.0.x/ installiert, sofern du den Installationsort nicht geändert hast.

Lege vor Änderungen am Mod unbedingt eine Sicherheitskopie an. So stellst du sicher, dass im Fehlerfall nur deine Anpassungen betroffen sind und nicht der gesamte Mod beschädigt wird.

Entpacke die ZIP-Datei deines Mods, bevor du mit der Bearbeitung beginnst. Dies kannst du mit den in Windows integrierten Tools oder Programmen wie 7Zip bzw. WinRAR erledigen. Kopiere den entpackten Mod-Ordner anschließend in deinen LS25 Mod-Ordner. Achte darauf, dass sich darin nur der entpackte Mod-Ordner befindet und keine weitere ZIP-Datei mit gleichem Namen.

#### 1.1.3 Einbau der Echten Arbeitskamas

Öffne nach dem Entpacken im Mod-Ordner die i3d-Datei deines Fahrzeugs mit dem GIANTS Editor. Im Editor navigierst du zum Reiter „Skripte“, wählst „Geteilte Skripte“, dann „Real Work Cameras Setup“ und schließlich „Real Work Camera Generate Vehicle XML“. Es öffnet sich ein Fenster mit verschiedenen Werkzeugen, mit denen du Kameras, Displays und visuelle Kameras einfach verbauen kannst.

Um eine Kamera samt Display zu platzieren, klicke im Skriptfenster auf "Add plane and camera for placement". Das Skript fügt automatisch eine neue Transform Group (Transformationsgruppe / TG) am Ende des Viewports hinzu, die als Vorlage für den späteren Einbau dient.

Positioniere die eingefügte Transformationsgruppe im GIANTS Editor an der gewünschten Stelle für das Display. Falls nötig, kannst du die TG in eine andere Gruppe verschieben, z.B. in die des Sitzes, damit sich das Display bei Bewegungen mitbewegt. Passe Position und Größe des Displays durch Skalieren der TG an – achte darauf, wirklich die TG selbst zu skalieren und nicht das Display, das in dieser liegt. Das gleiche machst du für die Kamera. Diese positionierst du an die gewünschte Stelle, z.B. am Heck des Schleppers oder am Rohr eines Dreschers. Achte auch hier darauf, dass die Kamera im richtigen Objekt verankert ist, dass sie sich bei Bewegungen des Fahrzeugs mitbewegen kann. Du kannst auch den FOV der Kamera im Attributes Bereich unter Kamera verändern. Der FOV wird später vom Skript automatisch übernommen. Wenn du die Ansicht der Kamera im Editor sehen möchtest, kannst du die Kamera auswählen indem du in der Menüleiste auf "Ansicht" klickst und dann im Menü "Kamera" die entsprechende Kamera auswählst. Für eine bessere Übersichtlichkeit kannst du die Kamera auch umbenennen.

Nach der Platzierung kannst du die TG und anschließend die Kamera in genau dieser Reihenfolge auswählen und im Skriptfenster auf „Create Vehicle XML“ klicken. In der Konsole des GIANTS Editors erscheint nun ein XML-Eintrag, den du für die weitere Bearbeitung benötigst. Wähle anschließend nur die TG an und entferne das Display in der TG mit „Delete Plane“ und speichere die i3d-Datei.

### 1.1.4 Aufbau der XML-Einträge

Öffne nun die XML-Datei deines Fahrzeugs, am besten mit Notepad++ oder Visual Studio Code, alternativ mit dem Windows-Editor. Füge den zuvor aus der GIANTS Editor-Konsole kopierten XML-Eintrag in die Fahrzeug-XML ein. Achte darauf, dass sich der Eintrag innerhalb des `<vehicle>`-Tags befindet, jedoch nicht innerhalb anderer spezifischer Tags wie `<i3dMappings>` oder `<designConfigurations>`.

Dein Eintrag sollte etwa wie folgt aussehen (Beispiel für eine Kamera und ein Display):

```
1 <vehicle>
2   [...]
```

# Echte Arbeitskamas

## Einbauanleitung

```
3      <realWorkCameras>
4          <cameraGroups>
5              <cameraGroup name="CAMS_DRIVE" title="
$110n_cameraGroup_driveCams" ignitionDependent="true" stateDelay="1000"
realGPSHideMapIndices="1"/>
6          </cameraGroups>
7
8          <workCameras>
9              <workCamera prio="1" group="CAMS_DRIVE" isReverseCam="true"
displayNode="0>0|6|12" cameraNode="0>0|6|13" cameraFovY="90"
nightReflectionScale="10" aspectRatio="1" reflectionScale="1" />
10             </workCameras>
11
12             <hideNodes>
13                 <hideNode node="displayRadio_decal" isDashboard="true" />
14             </hideNodes>
15
16             <showNodes>
17                 <showNode node="cameraDisplayNode" />
18             </showNodes>
19
20             <assets>
21                 <asset filename="$data/shared/assets/camera_03.i3d" linkNode="
0>0|4|2|0|0|0|0|0|0|0" node="0" position="0_0-0.624_1.002" rotation="-90
_0_0-180" scale="1_1_1" rotationNode="0|0" rotationNodeRotation="-45_0_0"
/>
22             </assets>
23         </realWorkCameras>
24         [...]
25 </vehicle>
```

Das Skript aus dem GIANTS Editor gibt immer eine komplette Basiskonfiguration aus. Möchtest du mehrere Kameras und Displays verbauen, genügt es ab der zweiten Kamera und dem zweiten Display, nur die Einträge innerhalb der `<workCameras>`-Tags zu kopieren und in die XML-Datei einzufügen.

### 1.1.5 Kameragruppen

Kameragruppen legen fest, welche Kameras zusammengefasst werden. Es empfiehlt sich, die Displays der Kameras einer Gruppe an gleicher Position und in gleicher Größe zu verbauen. Der Mod blendet dann automatisch alle nicht ausgewählten Kameras

aus. Die Attribute `name` und `title` sind frei wählbar, wobei der Name keine Leerzeichen enthalten darf. Der Titel wird im Spiel im F1-Menü angezeigt und kann eigene l10n-Einträge enthalten. Das Attribut `ignitionDependent` bestimmt, ob die Kameras dieser Gruppe nur bei aktiver Zündung (`true`) oder immer (`false`) verfügbar sind. Das Attribut `isMotorActiveDependent` legt fest, ob die Kameras nur bei laufendem Motor (`true`) oder immer (`false`) verfügbar sind. Wird `ignitionDependent` gesetzt, überschreibt es die Einstellung von `isMotorActiveDependent`. Mit `stateDelay` kann eine Verzögerung in Millisekunden festgelegt werden, bis die Kamera nach Motorstart aktiviert wird – sinnvoll, da manche Fahrzeuge etwas Zeit zum Starten benötigen. Das optionale Attribut `realGPSHideMapIndices` blendet RealGPS-Kartenelemente aus, wenn eine Kamera dieser Gruppe aktiv ist. Mehrere Indices werden durch Leerzeichen getrennt (z.B. "1 2 3"). Ist keine Kamera aktiv, wird die Karte wieder eingeblendet.

### 1.1.6 Kameras

Im `<workCameras>`-Tag werden die eigentlichen Kameras mit `<workCamera>` definiert. Das Attribut `prio` legt die Priorität fest: niedrigere Zahlen bedeuten höhere Priorität. Mit `group` wird die zugehörige Kameragruppe angegeben. Das Attribut `isReverseCam` bestimmt, ob die Kamera als Rückfahrkamera fungiert. Rückfahrkameras werden automatisch aktiviert, wenn der Rückwärtsgang eingelegt und der Automatikmodus aktiv ist. Für Rohrkameras gibt es das Attribut `isPipeCam`, das die Kamera bei ausgefahrenem Abtankrohr aktiviert. Die Attribute `displayNode` und `cameraNode` geben die Indexpfade bzw. i3dMappings von Kamera und Display an. Diese werden durch das Skript generiert und sollten übernommen werden, sofern du dich nicht auskennst. Mit `cameraFovY` wird das Sichtfeld der Kamera in Grad festgelegt. Höhere Werte erweitern das Sichtfeld, niedrigere schränken es ein. `reflectionScale` bestimmt den allgemeinen Reflexionsfaktor, `nightReflectionScale` den Reflexionsfaktor bei Nacht. Beide beeinflussen die Stärke der Reflexionen. Das Attribut `aspectRatio` legt das Seitenverhältnis des Displays fest. Meist genügt der Wert 1, es sei denn, das Kamerabild ist verzerrt.

### 1.1.7 Hide- und ShowNodes

Im `<hideNodes>`-Tag können Elemente definiert werden, die ausgeblendet werden sollen, wenn das Fahrzeug Kameras verbaut hat – etwa ein vorhandenes Radio-Display. Das

`<hideNode>`-Tag gibt hierfür den Indexpfad oder das i3dMapping der TG an, die ausgeblendet werden soll. Den Indexpfad findest du im GIANTS Editor unter "Attribute", das i3dMapping in der Fahrzeug-XML unter `<i3dMappings>`. Mit `isDashboard` wird festgelegt, ob das auszublendende Element in einem Dashboard-Eintrag registriert ist. So können z.B. `VISIBILITY`-Dashboards vom Mod überschrieben und dauerhaft ausgeblendet werden.

Im `<showNodes>`-Tag werden Elemente definiert, die eingeblendet werden sollen, wenn das Fahrzeug Kameras verbaut hat. Das `<showNode>`-Tag gibt hierfür den Indexpfad oder das i3dMapping der TG an.

### 1.1.8 Assets

Im `<assets>`-Tag können zusätzliche i3d-Dateien eingebunden werden, die für Kameras und Displays benötigt werden. Das `<asset>`-Tag funktioniert analog zu `<dynamicallyLoadedParts>` aus dem Grundspiel, mit dem Unterschied, dass hier eine `rotationNode` und die zugehörige `rotationNodeRotation` verwendet werden können, um die Drehung eines Kindknotens relativ zur TG des Assets anzugeben. Assets werden nur geladen, wenn das Fahrzeug eine gültige Kamera-Konfiguration besitzt – also mindestens eine Kamera und ein Display verbaut und im Shop konfiguriert sind.

### 1.1.9 Testen des Mods

Du kannst deinen Mod nun im LS25 testen. Es ist nicht zwingend erforderlich, ihn erneut als ZIP-Datei zu speichern, da der Landwirtschafts-Simulator im Singleplayer auch Ordner als Mods akzeptiert. Das erleichtert Anpassungen an Kameras und Displays. Funktioniert im Spiel alles wie gewünscht, kannst du den Mod wieder als ZIP-Datei packen. Falls eine Kamera oder ein Display nicht richtig positioniert ist oder Probleme auftreten, wiederhole die oben beschriebenen Schritte. Verwende dabei jedoch nicht erneut die vom Skript erstellte Vorlage, sondern wähle die zuvor erstellte Transformationsgruppe aus und klicke im Skriptfenster auf „Restore Plane“. Dadurch wird eine visuelle Darstellung des Displays erneut erzeugt. Nach der Anpassung kannst du den Einbauprozess wie zuvor abschließen.

## 1.2 Einbau in Fahrzeuge – Erweiterte Optionen

### 1.2.1 Konfigurationsabhängige Kameras und Displays

Manchmal ist es sinnvoll, dass bestimmte Kameras und Displays nur in bestimmten Fahrzeugkonfigurationen verfügbar sind. Dafür können die Attribute `configurationName` und `configurationIndex` im `<workCamera>`-Tag verwendet werden. `configurationName` gibt den Namen der Konfiguration an, z.B. `configurationName="design"` für `designConfigurations` oder `configurationName="pipe"` für `pipeConfigurations`.

Der Index der Konfiguration wird durch `configurationIndex` angegeben. Der Index beginnt bei 1 für die erste Konfiguration, 2 für die zweite usw. Soll eine Kamera in mehreren Konfigurationen verfügbar sein, muss für jede Konfiguration ein separater Eintrag mit den entsprechenden Attributen erstellt werden.

### 1.2.2 Kameradisplays in dashboardCompounds

Einige Fahrzeuge im LS25 nutzen sogenannte `dashboardCompounds`, um gleiche Displays in eine externe i3d-Datei auszulagern und in mehreren Fahrzeugen zu verwenden. Auch in diese extern geladenen Displays können Kameradisplays integriert werden. Im Fahrzeug wird dann für eine Kamera nur auf das extern geladene Display verwiesen.

Hierfür können im `<dashboardCompound>`-Tag der Display-XML folgende Einträge gemacht werden:

```
1 <dashboardCompound>
2   [...]
3   <realWorkCameras>
4     <workCameraDisplays>
5       <workCameraDisplay linkNode="realWorkCameras"
6       nightReflectionScale="10" aspectRatio="1" reflectionScale="1" />
7     </workCameraDisplays>
8
9     <hideNodes>
10      <hideNode node="g5Display_gpsPageTractor_static" isDashboard="
11      true" />
12    </hideNodes>
13
14    <showNodes>
15      <showNode node="showNode" />
16    </showNodes>
17  </realWorkCameras>
18 </dashboardCompound>
```

# Echte Arbeitskamas

## Einbauanleitung

```
14         </showNodes>
15     </realWorkCameras>
16     [...]
17 </dashboardCompound>
```

Die Funktionalitäten und Attribute der `<workCameraDisplay>`, `<hideNode>` und `<showNode>` entsprechen denen der bereits beschriebenen Einträge im Fahrzeug. Der einzige Unterschied: Im `<workCameraDisplay>`-Tag müssen keine Kamera-Attribute angegeben werden, da diese im Fahrzeug eingetragen werden.

Im Fahrzeug werden die Kameras dann wie folgt definiert:

```
1 <vehicle>
2     [...]
3     <realWorkCameras>
4         <cameraGroups>
5             <cameraGroup name="CAMS_DRIVE" title="
6 $l10n_cameraGroup_driveCams" ignitionDependent="true" stateDelay="1000"
7 realGPSHideMapIndices="1" />
8         </cameraGroups>
9
10        <workCameras>
11            <workCamera prio="1" group="CAMS_DRIVE" isReverseCam="true"
12 cameraNode="0>0|6|13" cameraFovY="90" nightReflectionScale="10"
13 aspectRatio="1" reflectionScale="1" dashboardCompoundIndex="2"
14 displayIndex="1" />
15        </workCameras>
16
17        <hideNodes>
18            <hideNode node="displayRadio_decal" isDashboard="true" />
19        </hideNodes>
20    </realWorkCameras>
21    [...]
22 </vehicle>
```

Das Attribut `dashboardCompoundIndex` gibt den Index der dashboardCompound-i3d-Datei an, in der sich das Kameradisplay befindet (beginnend bei 1). `displayIndex` gibt den Index des Kameradisplays innerhalb der dashboardCompound-i3d-Datei an (ebenfalls ab 1). Im Fahrzeug wird dann kein `displayNode`-Attribut benötigt, da das Display in der dashboardCompound-i3d-Datei definiert ist.





### 1.2.3 AttachableCams mit zugehörigem Display

AttachableCams sind Kameras, die am Anbaugerät verbaut sind, deren Bild aber auf einem Display im Zugfahrzeug angezeigt wird. Dafür werden zwei Stellen konfiguriert: Das Zugfahrzeug stellt eine Kameragruppe und ein Display bereit, das Anbaugerät stellt die eigentlichen Kameraeinträge bereit.

**Wichtig:** Pro Fahrzeug kann immer nur eine `AttachableCamGroup` verwendet werden. Markiere deshalb nur eine `<cameraGroup>` mit `isAttachableCamGroup="true"`. Werden mehrere Gruppen so markiert, nutzt der Mod nur die erste gefundene Gruppe und ignoriert weitere `AttachableCamGroups`.

Für ein normal im Fahrzeug verbautes AttachableCam-Display wird die Display-TG direkt in der `AttachableCamGroup` angegeben:

```
1 <vehicle>
2   [...]
3   <realWorkCameras>
4     <cameraGroups>
5       <cameraGroup name="CAMS_ATTACHABLE" title="
6 $l10n_cameraGroup_attachableCams" ignitionDependent="true"
7 isAttachableCamGroup="true">
8       <attachableCamDisplay displayNode="0>0|6|12"
9         nightReflectionScale="10" aspectRatio="1"
10        reflectionScale="1" prio="1" />
11     </cameraGroup>
12   </cameraGroups>
13 </realWorkCameras>
14   [...]
15 </vehicle>
```

Das Attribut `displayNode` verweist auf die Display-TG im Zugfahrzeug. `prio`, `nightReflectionScale`, `aspectRatio` und `reflectionScale` funktionieren wie bei normalen Kameradisplays.

Befindet sich das Display in einem `dashboardCompound`, wird es in der Display-XML als `AttachableCam-Display` markiert:

```
1 <dashboardCompound>
2   [...]
3   <realWorkCameras>
4     <workCameraDisplays>
5       <workCameraDisplay linkNode="realWorkCameras"
6         nightReflectionScale="10" aspectRatio="1" reflectionScale="1"
7         isAttachableCamDisplay="true" />
8     </workCameraDisplays>
9   </realWorkCameras>
10  </dashboardCompound>
```

# Echte Arbeitskamas

## Einbauanleitung

```
6         </workCameraDisplays>
7     </realWorkCameras>
8     [...]
9 </dashboardCompound>
```

Im Fahrzeug muss trotzdem eine AttachableCamGroup vorhanden sein, damit die Kameras des Anbaugeräts einer Gruppe zugeordnet werden können:

```
1 <vehicle>
2     [...]
3     <realWorkCameras>
4         <cameraGroups>
5             <cameraGroup name="CAMS_ATTACHABLE" title="
6 $l10n_cameraGroup_attachableCams" ignitionDependent="true"
7 isAttachableCamGroup="true" />
8         </cameraGroups>
9     </realWorkCameras>
10    [...]
11 </vehicle>
```

Soll das Display im `dashboardCompound` ausschließlich für AttachableCams verwendet werden und nicht zusätzlich als normales Kameradisplay zur Verfügung stehen, kann `isExclusiveAttachableCamDisplay="true"` ergänzt werden.

Im Anbaugerät selbst werden die Kameras unter `<attachableWorkCameras>` eingetragen. Ein `displayName` wird dort nicht angegeben, da das Display vom Zugfahrzeug kommt:

```
1 <vehicle>
2     [...]
3     <attachableWorkCameras>
4         <workCameras>
5             <workCamera cameraNode="0>0|5|0" cameraFovY="90" prio="1"
6 isReverseCam="true" />
7             <workCamera cameraNode="0>0|5|1" cameraFovY="80" prio="2"
8 isPipeCam="true" />
9         </workCameras>
10    </attachableWorkCameras>
11    [...]
12 </vehicle>
```

Die Attribute `cameraNode`, `cameraFovY`, `prio`, `isReverseCam`, `isPipeCam`, `displayMirrored`, `isOrthographic` und `orthographicHeight` entsprechen den normalen Kameraattributen. Optional können `configurationName` und `configurationIndices` verwendet werden, damit eine Attach-

# Echte Arbeitskamas

## Einbauanleitung

ableCam nur in bestimmten Konfigurationen des Anbaugeräts geladen wird.



## 2 English

### 2.1 Installation in Vehicles

#### 2.1.1 Introduction

This guide describes how to prepare a mod vehicle for use with the Real Work Cameras mod in FS25. The mod provides a script that simplifies the installation of cameras and displays in a tractor.

#### 2.1.2 Requirements and Preparation

To install, you need the script from the Real Work Cameras mod as well as the latest version of the GIANTS Editor for FS25. You can download the editor from the official website: <https://gdn.giants-software.com/index.php>.

You will find the required scripts in the "sdk" folder of the Real Work Cameras mod. Copy the "realWorkCamerasSetup" folder from there into the "scripts" directory of your GIANTS Editor installation. By default, the editor is installed at C:/Programme/GIANTS Software/GIANTS Editor 10.0.x/, unless you changed the installation path.

Always create a backup before making changes to your mod. This ensures that only your modifications are affected in case of errors, and not the entire mod.

Unpack your mod's ZIP file before starting. You can do this with the built-in Windows tools or with programs like 7Zip or WinRAR. Then copy the unpacked mod folder into your FS25 mods folder. Make sure that only the unpacked mod folder is inside, and not another ZIP file with the same name.

#### 2.1.3 Installing the Real Work Cameras

After unpacking, open your vehicle's i3d file in the mod folder using the GIANTS Editor. In the editor, go to the "Scripts" tab, select "Shared Scripts", then "Real Work Cameras Setup", and finally "Real Work Camera Generate Vehicle XML". A window will open with various tools that allow you to easily install cameras, displays, and visual cameras.

To place a camera and display, click "Add plane and camera for placement" in the script window. The script will automatically add a new Transform Group (TG) at the

end of the viewport, which serves as a template for later installation.

Position the inserted Transform Group in the GIANTS Editor at the desired location for the display. If necessary, you can move the TG into another group, e.g., the seat group, so the display moves with the seat. Adjust the position and size of the display by scaling the TG—make sure to scale the TG itself, not the display inside it. Repeat the same for the camera: position the camera at the desired location, e.g., at the rear of the tractor or on the pipe of a harvester. Ensure the camera is anchored to the correct object so it moves with the vehicle. You can also adjust the camera's FOV (field of view) in the Attributes section under Camera; the script will automatically use this value. If you want to see the camera's view in the editor, select the camera, then in the menu bar click "View" and under "Camera" choose the respective camera. For better clarity, you can also rename the camera.

After placement, select the TG and then the camera in exactly this order, and click "Create Vehicle XML" in the script window. An XML entry will now appear in the GIANTS Editor console, which you will need for further editing. Then select only the TG and remove the display in the TG with "Delete Plane", and save the i3d file.

### 2.1.4 Structure of the XML Entries

Now open your vehicle's XML file, preferably with Notepad++ or Visual Studio Code, or alternatively with the Windows Editor. Paste the XML entry you copied from the GIANTS Editor console into the vehicle XML. Make sure the entry is within the `<vehicle>` tags, but not inside other specific tags like `<i3dMappings>` or `<designConfigurations>`.

Your entry should look like this (example for one camera and one display):

```
1 <vehicle>
2   [...]
3   <realWorkCameras>
4     <cameraGroups>
5       <cameraGroup name="CAMS_DRIVE" title="
6 $110n_cameraGroup_driveCams" ignitionDependent="true" stateDelay="1000"
7 realGPSHideMapIndices="1"/>
8     </cameraGroups>
9
10    <workCameras>
11      <workCamera prio="1" group="CAMS_DRIVE" isReverseCam="true"
12 displayNode="0>0|6|12" cameraNode="0>0|6|13" cameraFovY="90"
13 nightReflectionScale="10" aspectRatio="1" reflectionScale="1" />
```



# Real Work Cameras

## Setup Guide

```
10         </workCameras>
11
12         <hideNodes>
13             <hideNode node="displayRadio_decal" isDashboard="true" />
14         </hideNodes>
15
16         <showNodes>
17             <showNode node="cameraDisplayNode" />
18         </showNodes>
19
20         <assets>
21             <asset filename="$data/shared/assets/camera_03.i3d" linkNode="
0>0|4|2|0|0|0|0|0|0|0" node="0" position="0 0 -0.624 -1.002" rotation="-90
 0 0 -180" scale="1 1 1" rotationNode="0|0" rotationNodeRotation="-45 0 0"
 />
22         </assets>
23     </realWorkCameras>
24     [...]
25 </vehicle>
```

The script from the GIANTS Editor always outputs a complete base configuration. If you want to install multiple cameras and displays, from the second camera and display onwards, it is sufficient to copy only the entries within the `<workCameras>` tags into the XML file.

### 2.1.5 Camera Groups

Camera groups define which cameras are grouped together. It is recommended to install the displays of cameras in the same group at the same position and size. The mod will then automatically hide all non-selected cameras. The attributes `name` and `title` can be chosen freely, but the name must not contain spaces. The title is shown in the game's F1 menu and can contain your own l10n entries. The attribute `ignitionDependent` determines whether the cameras in this group are only available when the ignition is on (true) or always (false). The attribute `isMotorActiveDependent` specifies whether the cameras are only available when the engine is running (true) or always (false). If `ignitionDependent` is set, it overrides the setting of `isMotorActiveDependent`. With `stateDelay` you can set a delay in milliseconds before the camera is activated after the engine starts—useful as some vehicles take time to start. The optional attribute `realGPSHideMapIndices` hides RealGPS map elements when a camera in this group is active. Multiple indices



are separated by spaces (e.g., "1 2 3"). If no camera is active, the map is shown again.

### 2.1.6 Cameras

Within the `<workCameras>` tag, the actual cameras are defined using `<workCamera>`. The attribute `prio` sets the priority: lower numbers mean higher priority. The `group` attribute specifies the camera group. The attribute `isReverseCam` determines if the camera acts as a reversing camera. Reversing cameras are automatically activated when reverse gear is engaged and automatic mode is active. For pipe cameras, use the attribute `isPipeCam`, which activates the camera when an unloading pipe is extended. The attributes `displayNode` and `cameraNode` specify the index paths or i3dMappings of the camera and display. These are generated by the script and should be used as is unless you know what you are doing. `cameraFovY` sets the camera's field of view in degrees. Higher values widen the field of view, lower values narrow it. `reflectionScale` sets the general reflection factor, `nightReflectionScale` the reflection factor at night. Both affect the strength of reflections. The attribute `aspectRatio` sets the display's aspect ratio. Usually, the value 1 is sufficient unless the camera image is distorted.

### 2.1.7 Hide and Show Nodes

Within the `<hideNodes>` tag, you can define elements to be hidden when the vehicle has cameras installed—such as an existing radio display. The `<hideNode>` tag specifies the index path or i3dMapping of the TG to be hidden. You can find the index path in the GIANTS Editor under "Attribute", and the i3dMapping in the vehicle XML under `<i3dMappings>`. With `isDashboard` you define whether the element to be hidden is registered in a dashboard entry. This allows, for example, `VISIBILITY` dashboards to be overridden by the mod and permanently hidden.

Within the `<showNodes>` tag, you can define elements to be shown when the vehicle has cameras installed. The `<showNode>` tag specifies the index path or i3dMapping of the TG to be shown.

### 2.1.8 Assets

Within the `<assets>` tag, you can include additional i3d files needed for cameras and displays. The `<asset>` tag works similarly to `<dynamicallyLoadedParts>` from the base game, with the difference that here you can use a `rotationNode` and the corresponding

`rotationNodeRotation` to specify the rotation of a child node relative to the TG of the asset. Assets are only loaded if the vehicle has a valid camera configuration—that is, at least one camera and one display are installed and configured in the shop.

### 2.1.9 Testing the Mod

You can now test your mod in FS25. It is not strictly necessary to repackage it as a ZIP file, since Farming Simulator also accepts folders as mods in singleplayer. This makes it easier to adjust cameras and displays. If everything works as desired in the game, you can repackage the mod as a ZIP file. If a camera or display is not positioned correctly or problems occur, repeat the steps described above. However, do not use the template created by the script again; instead, select the previously created Transform Group and click “Restore Plane” in the script window. This will recreate a visual representation of the display. After making your adjustments, you can complete the installation process as before.

## 2.2 Installation in Vehicles – Advanced Options

### 2.2.1 Configuration-dependent Cameras and Displays

Sometimes it makes sense for certain cameras and displays to be available only in specific vehicle configurations. For this, you can use the attributes `configurationName` and `configurationIndex` in the `<workCamera>` tag. `configurationName` specifies the name of the configuration, e.g., `configurationName="design"` for `designConfigurations` or `configurationName="pipe"` for `pipeConfigurations`.

The configuration index is specified by `configurationIndex`. The index starts at 1 for the first configuration, 2 for the second, and so on. If you want a camera to be available in multiple configurations, you must create a separate entry for each configuration with the corresponding attributes.

### 2.2.2 Camera Displays in dashboardCompounds

Some vehicles in FS25 use so-called dashboardCompounds to outsource identical displays to an external i3d file and use them in several vehicles. Camera displays can also be installed in these externally loaded displays. In the vehicle, a camera then simply refers to the externally loaded display.



# Real Work Cameras

## Setup Guide

For this, the following entries can be made in the `<dashboardCompound>` tag of the display XML:

```
1 <dashboardCompound>
2   [...]
3   <realWorkCameras>
4     <workCameraDisplays>
5       <workCameraDisplay linkNode="realWorkCameras"
6       nightReflectionScale="10" aspectRatio="1" reflectionScale="1" />
7     </workCameraDisplays>
8   <hideNodes>
9     <hideNode node="g5Display_gpsPageTractor_static" isDashboard="
10    true" />
11  </hideNodes>
12  <showNodes>
13    <showNode node="showNode" />
14  </showNodes>
15 </realWorkCameras>
16 [...]
17 </dashboardCompound>
```

The functionalities and attributes of `<workCameraDisplay>`, `<hideNode>`, and `<showNode>` are identical to those already described for the vehicle entries. The only difference: In the `<workCameraDisplay>` tag, no camera attributes need to be specified, as these are entered in the vehicle.

In the vehicle, the cameras are then defined as follows:

```
1 <vehicle>
2   [...]
3   <realWorkCameras>
4     <cameraGroups>
5       <cameraGroup name="CAMS_DRIVE" title="
6       $110n_cameraGroup_driveCams" ignitionDependent="true" stateDelay="1000"
7       realGPSHideMapIndices="1"/>
8     </cameraGroups>
9     <workCameras>
10      <workCamera prio="1" group="CAMS_DRIVE" isReverseCam="true"
11      cameraNode="0>0|6|13" cameraFovY="90" nightReflectionScale="10"
12      aspectRatio="1" reflectionScale="1" dashboardCompoundIndex="2"
```



```
10     displayIndex="1" />
11     </workCameras>
12
13     <hideNodes>
14         <hideNode node="displayRadio_decal" isDashboard="true" />
15     </hideNodes>
16 </realWorkCameras>
17 [...]
```

The attribute `dashboardCompoundIndex` specifies the index of the dashboardCompound i3d file in which the camera display is located (starting at 1). `displayIndex` specifies the index of the camera display within the dashboardCompound i3d file (also starting at 1). No `displayNode` attribute is needed in the vehicle, as the display is defined in the dashboardCompound i3d file.

### 2.2.3 AttachableCams with Matching Display

AttachableCams are cameras installed on an implement, while their image is shown on a display in the towing vehicle. This requires configuration in two places: The towing vehicle provides the camera group and display, while the implement provides the actual camera entries.

**Important:** Each vehicle can only have one `AttachableCamGroup`. Therefore, mark only one `<cameraGroup>` with `isAttachableCamGroup="true"`. If multiple groups are marked this way, the mod uses only the first group found and ignores additional `AttachableCamGroups`.

For a normally installed `AttachableCam` display in the vehicle, specify the display TG directly inside the `AttachableCamGroup`:

```
1 <vehicle>
2   [...]
3   <realWorkCameras>
4     <cameraGroups>
5       <cameraGroup name="CAMS_ATTACHABLE" title="
6         $l10n_cameraGroup_attachableCams" ignitionDependent="true"
7         isAttachableCamGroup="true">
          <attachableCamDisplay displayNode="0>0|6|12"
            nightReflectionScale="10" aspectRatio="1"
            reflectionScale="1" prio="1" />
        </cameraGroup>
```



# Real Work Cameras

## Setup Guide

```
8         </cameraGroups>
9     </realWorkCameras>
10    [...]
11</vehicle>
```

The `displayNode` attribute points to the display TG in the towing vehicle. `prio`, `nightReflectionScale`, `aspectRatio`, and `reflectionScale` work like they do for regular camera displays.

If the display is located in a `dashboardCompound`, mark it as an `AttachableCam` display in the display XML:

```
1 <dashboardCompound>
2     [...]
3     <realWorkCameras>
4         <workCameraDisplays>
5             <workCameraDisplay linkNode="realWorkCameras"
6               nightReflectionScale="10" aspectRatio="1" reflectionScale="1"
7               isAttachableCamDisplay="true" />
8         </workCameraDisplays>
9     </realWorkCameras>
10    [...]
11</dashboardCompound>
```

The vehicle still needs an `AttachableCamGroup` so the implement cameras can be assigned to a group:

```
1 <vehicle>
2     [...]
3     <realWorkCameras>
4         <cameraGroups>
5             <cameraGroup name="CAMS_ATTACHABLE" title="
6               $l10n_cameraGroup_attachableCams" ignitionDependent="true"
7               isAttachableCamGroup="true" />
8         </cameraGroups>
9     </realWorkCameras>
10    [...]
11</vehicle>
```

If the display in the `dashboardCompound` should be used only for `AttachableCams` and not also be available as a regular camera display, add `isExclusiveAttachableCamDisplay="true"`.

On the implement itself, define the cameras under `<attachableWorkCameras>`. Do not specify a `displayNode` there, because the display is provided by the towing vehicle:



# Real Work Cameras

## Setup Guide

```
1 <vehicle>
2   [...]
3   <attachableWorkCameras>
4     <workCameras>
5       <workCamera cameraNode="0>0|5|0" cameraFovY="90" prio="1"
isReverseCam="true" />
6       <workCamera cameraNode="0>0|5|1" cameraFovY="80" prio="2"
isPipeCam="true" />
7     </workCameras>
8   </attachableWorkCameras>
9   [...]
10 </vehicle>
```

The attributes `cameraNode`, `cameraFovY`, `prio`, `isReverseCam`, `isPipeCam`, `displayMirrored`, `isOrthographic`, and `orthographicHeight` match the normal camera attributes. Optionally, use `configurationName` and `configurationIndices` to load an `AttachableCam` only for specific implement configurations.