Content Philosophy

Inhaltsverzeichnis

1 Huge thanks to thor2950 for the translation!

2 Principle

2.1 Container

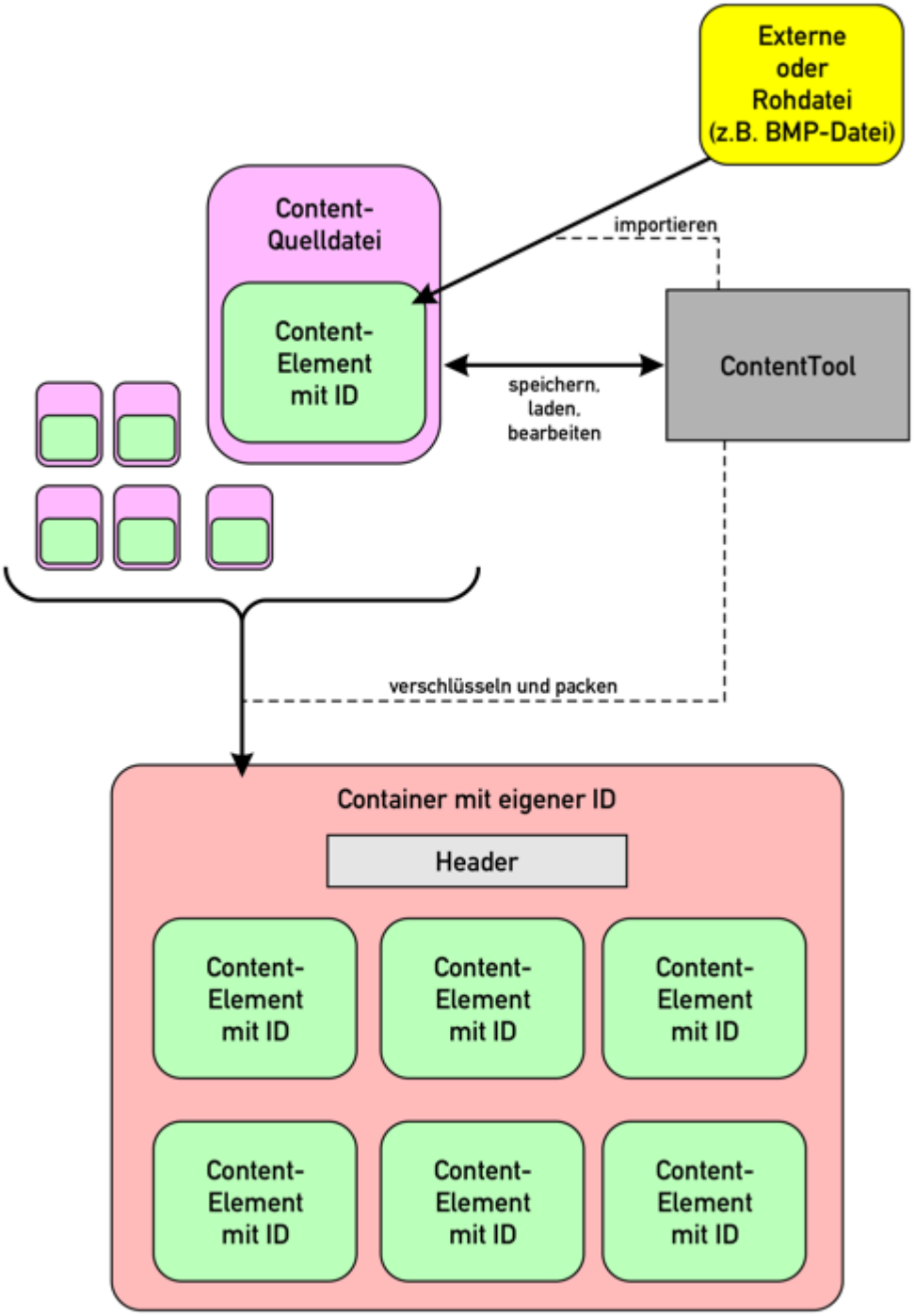
LOTUS reads the content from containers (*.lct-files), which can be found in one of these four sub folders:

- **Base** and **BaseAddons**: Content provided with LOTUS by default. Caution: These folders are managed by Steam. You're not allowed to put your own files here!
- **Addons**: Here you can place addon containers from other people. E.g. the payware addon "Düsseldorf" is placed here automatically.
- **MyContent**: Your own content containers from the Content Tool and the Map Editor are contained here.

If the container belongs to a workshop item it is then placed in a folder given by Steam, from which LOTUS also reads.

A container often contains more than one item, for example a complete group of street objects. Moreover, containers are locked and can only be loaded by the actual simulation. Containers cannot be loaded or edited by the LOTUS-Tools.

2.2 ContentTool



The raw data in the form of *.bmp- and *.dds-files for textures, *.x3d-files for 3D models and *.wav-files for sounds are imported and configured with the ContentTool. After that, you can save the imported and configured scenery objects, vehicles, textures, soundsets, languages or typefonts as content source-files.

For a future container, a working folder is created outside any LOTUS-folder or special program folder. Much

better is to use the "Documents"-folder from Windows and create a working directory in there. Every belonging raw data is then placed here and at the same time, the source data from the ContentTool is saved here. The tool creates an "Export"-folder, where it saves important data which normally should never be deleted!

Now, the ContentTool can "pack" this folder, in which every source files are joined into one container, and this container is then automatically moved to the LOTUS-MyContent directory.

The folder with the source data must never be deleted!!! If deleted, the content belongning to this folder cannot be edited in the ContentTool!

2.3 MapEditor

The Map Editor is obviously used to create and edit maps. A working directory for the Map Editor is chosen, which contains a sub folder for every map, which then again contains all the source data for the map. These can then be loaded and saved by the Map Editor. **This folder must also never be deleted**, as long as the map should be editable: **The Map Editor is not able to load maps from containers!**

2.4 Content sourcefiles

These are the source file types, which can be loaded ans saved by the ContentTool/Map Editor:

- *.ltx: Textures
- *.lob: Objects (Scenery objects or vehicles)
- *.lge: Additional information for an object, always belongs to a *.lob-file
- *.lgo: Graphical object, always belongs to a *.lob-file
- *.lgp: Additional information to a graphical object, lways belongs to a *.lgo-file
- *.lsd: Soundset
- *.llg: Language data
- *.mgl: Main map data
- *.mge: Additional map information
- *.mtl: Map tile
- *.mte: Additional map tile information

2.5 Content-ID

2.5.1 To verify what content is made by who, in what version it is and with what permissions the content is distributed, every container carries a Content-ID. It is assigned once by creation of the container and consists of three elements:

- Content-User-ID or Team-ID, this identifies the author
- Content-SubID, this separates and identifies the different content by an author
- Version, here this means the creation date.

The User-ID is automatically assigned to every LOTUS user by the first login with the user's Steam account and is then associated with this Steam account and never changed afterwards.

The assignment of Sub-IDs happens automatically: When work on a new content item is begun (for example on an empty area on the map or a new building), a new Sub-ID is then generated for this item as a random number between 1.000.001 and 999.999.999.

When the Content Tool needs to assign a new Content ID during the work on a foreign object (for example when importing a new texture to a foreign vehicle), in this case the User-ID of the original author (aka the author of the foreign object) is used and a new random Sub-ID is then generated, but this time between 1.001.000.001 and 1.999.999.999 to avoid errors.

2.6 Teams

Every user can create a Team (and become the admin of it) and add users to the team. Every Team is automatically assigned an ID. All members of the Team are then able to choose by the start of the Content Tool/Map Editor if they want to develop content privately or for the Team. The advantage here is, that all content made for/by the team carries a uniform Content-Team-ID and and therefore doesn't "mix up" the individual User-ID's of the team members.

.....

2.7 User rights

For commercial purposes, content can be provided with an activation mechanism which requires payment and activation, before the content can be used. The term "usage" covers:

- Driving on maps (Usage of the map) or
- Driving vehicles (Usage of the vehicles)
- Using the vehicles for the AI on your own map (Usage of the vehicles)
- Using the scenery objects and other content on your own map (Usage of the scenery objects)

Textures and soundsets cannot be protected.

2.7.1 Structure

- In the Content Tool you can assign one or more Workshop/DLC-identifiers to scenery objects and vehicles. These identifiers are then included in the header of the item
- In the Map Editor, certain areas on maps can be protected by Workshop/DLC-Items. This includes both tiles and tracks where driving with street/rail vehicles is impossible until the user has activated these areas.

2.7.2 Implementation

- By the next startup, LOTUS checks which addons or (commercial) Workshop items (WS/DLC-Items) the user has acquired
- In the next step (also at startup), it is checked for every vehicle or scenery object if one or more WS/DLC-Items exist in it's header.
    - If the list is empty, the item is freeware and useable. The usage check was successful.
    - If the user has acquired at least one of the listed WS/DLC items, the object is considered acquired and can be used. The usage check was successful.
    - Otherwise the object is considered locked and the usage check fails.
- Before the map is loaded, it is checked whether it requires blocked scenery objects or blocked vehicles as AI and it is checked which protected areas of the map may be accessed by the user.

2.7.3 Effects

2.7.3.1 Maps

The map may only be loaded if at least one condition is fulfilled for every scenery object used on the map:

- Usage check successful
- Map and object have at least one common Workshop/DLC-ID

The same conditions apply to the usage of AI-vehicles in AI-lists. If the condition is not met for a vehicle, it isn't used as an AI-vehicle on this map and removed without replacement.

If the map can be loaded and as long as the user is in the free camera or ego-mode, the user can view or fly across all tiles or drive along with AI vehicles on all tiles. The user may not, however:

- place either the active or new vechicle on a locked tile,
- take control of an AI vehicle, as long as it is located outside the unlocked area or
- leave the unlocked area with the user vehicle. In this case however, it wouldn't drive hard against a glass wall, but rather braked down gently.

2.7.3.2 Vehicles

If the usage check of the vehicle was successful, the user may use the vehicle without limitations (provided that the user has purchased the respective module).

If the usage check fails, the vehicle still appears as follows:

- To display the vehicles of other players in a multiplayer session
- As AI-traffic (provided that the conditions for loading the map have been met, i.e. the vehicle belongs to the same WS / DLC item as the map)

However, the user cannot drive the vehicle himself. This manifests itself in the fact that the user cannot select or take over an affected vehicle.

2.7.3.3 Map-Editor

Both for the use of foreign, commercial scenery objects when building a map, as well as for the use of foreign, commercial vehicles for the AI lists, the map author must have previously acquired the associated WS/DLC items "normally". There are no differentiated licenses or similar; everything that can be driven can also be used in foreign maps. However, the author should be aware that his map may then only be loaded by people who must have bought all the WS/DLC items used in the map for themselves.

2.8 Editing rights

Normally, a user can only edit/load/save his own content in the Content Tool. The ownership is checked via the User-ID.

For a user to be able to edit foreign content, he must fetch the original source data from the author, since the container can't be edited.

In addition, the user must be given the right to edit:

- Either the content source files were marked as public domain by the original author (this setting is off by default)
- or the original author must explicitly grant him editing rights.

In the case of public domain content, the content may be edited and saved either under the original ID (with the user ID of the original author) or under a new ID (with the user ID of the new author).

Otherwise the original author can grant editing rights to a selected user. Here he can optionally grant another user all rights to his entire content or explicitly assign rights for individual content elements. In the latter case, he can decide whether to allow overwriting (same content ID including user ID) and/or saving under a new content ID including the new user ID.

As soon as there is more than one author, all authors are entered in a list in the header of the file so that co-authorship can be checked retrospectively.