

Hallo,

habe auch erst vor 3 Wochen angefangen, mich mit dem Thema zu beschäftigen. So mach ich es:

(Keine Ahnung ob das der einfachste Weg ist, ich machs halt so...)

Ich nutze

- 1) den RDA-Explorer von lysannschlegel, hier: <https://github.com/lysannschlegel/RDAExplorer>
- 2) Ultraedit von IDM (weil man mit dem in Dateien auf der Festplatte nach Zeichenfolgen suchen und XML-Files schön zur Ansicht formatieren kann, es geht aber auch der kostenlose Notepad++, hier: <https://notepad-plus-plus.org/>)
- 3) den aktuellen AssetViewer von Miraak zum Auffinden von Assets und Items, die ich manipulieren möchte, hier: <https://github.com/Miraak7000/AssetViewer>
- 4) die ganzen MODs, die man in der Zone, im Nexus und bei github so finden kann - als Vorbild und Anschauungsobjekte
- 5) den File Viewer Plus 3 von Microsoft, zum Syntaxcheck und Filevergleich der XML-Dateien, hier: <https://www.microsoft.com/de-de/p/file-viewer-plus/9wzdnrcdcs7?activetab=pivot:overviewtab>
- 6) ein Tool zum Vergleich von Textdateien wie z.B. BeyondCompare, hier: <http://www.scootersoftware.com/download.php> oder das kostenlose WinMerge, hier <https://winmerge.org/?lang=de>
- 7) ein Tool zum Anzeigen von XML-Baumstrukturen, wie z.B. hier: <https://www.mindfusion.eu/xml-viewer.html> das vereinfacht das Navigieren in XML-Dateien

Ich weiß (dank [xforce](#) und [taubenangriff](#)):

- wie man die entsprechenden MODs und den notwendigen MOD-Loader installiert
- wie die Verzeichnisstruktur der MOD-Ordner aufgebaut ist
- das es besser ist, nach Änderungen an der MOD-Ordnerstruktur im Anno 1800 Verzeichnis den .cache-Ordner zu löschen (oder hab ich das irgendwo in der Zone gelesen?)

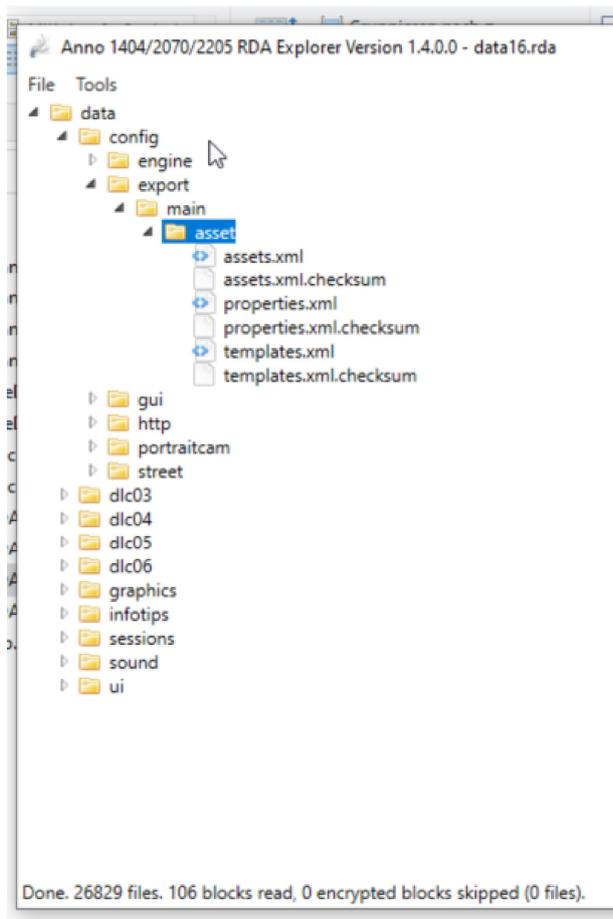
Nun los:

Bei Anno werden die Items und deren Eigenschaften (und fast alles andere ;-)) in XML-Files definiert. Seit dem Ldl-Update steckt das interessanteste für unsere Zwecke in der Datei data16.rda im Verzeichnis [PlatteaufderdasSpielinstalliert]\Ubisoft\Ubisoft Game Launcher\games\Anno 1800\maindata. In diesem Verzeichnis stecken noch mehr "data"s. Mit dem Ldl-Update kam bei mir noch die de_de5.rda hinzu, die data1.rda wurde durch eine neuere, größere Version ersetzt.

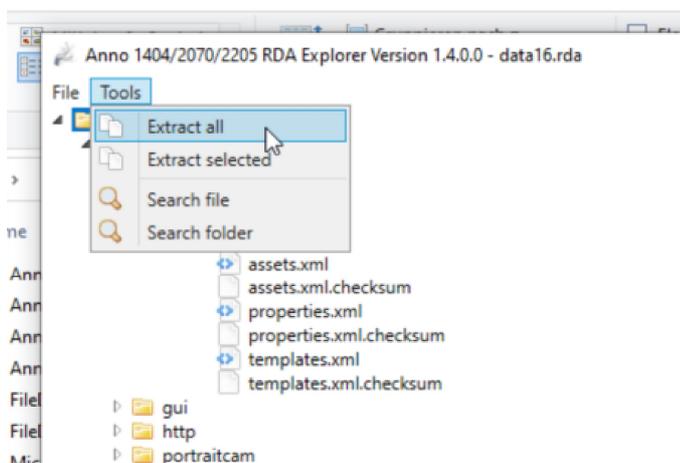
Ubisoft > Ubisoft Game Launcher > games > Anno 1800 > maindata

Name	Änderungsdatum	Typ	Größe
de_de5.rda	22.10.2020 19:00	RDA-Datei	177.384 KB
file.db	22.10.2020 19:00	Data Base File	356.990 KB
data16.rda	22.10.2020 18:59	RDA-Datei	6.483.381 KB
data1.rda	22.10.2020 18:49	RDA-Datei	5.323.395 KB
checksum.db	22.10.2020 18:48	Data Base File	1 KB
data15.rda	11.08.2020 18:08	RDA-Datei	1.027.320 KB
de_de4.rda	08.05.2020 13:00	RDA-Datei	29.893 KB
data14.rda	08.05.2020 13:00	RDA-Datei	1.073.659 KB
data9.rda	24.03.2020 19:44	RDA-Datei	474.799 KB
de_de3.rda	17.03.2020 12:44	RDA-Datei	41.869 KB
de_de1.rda	17.03.2020 12:44	RDA-Datei	29.559 KB
de_de0.rda	17.03.2020 12:44	RDA-Datei	493.808 KB
de_de2.rda	17.03.2020 12:44	RDA-Datei	1 KB
data13.rda	17.03.2020 12:42	RDA-Datei	6.085.850 KB
data12.rda	17.03.2020 12:32	RDA-Datei	2.354.586 KB
data11.rda	17.03.2020 12:28	RDA-Datei	2.509.125 KB
data10.rda	17.03.2020 12:24	RDA-Datei	523.371 KB
data8.rda	17.03.2020 12:22	RDA-Datei	11.340.619 ...
data7.rda	17.03.2020 12:02	RDA-Datei	143.169 KB
data6.rda	17.03.2020 12:01	RDA-Datei	8.958.512 KB
data5.rda	17.03.2020 11:46	RDA-Datei	25.720 KB
data4.rda	17.03.2020 11:46	RDA-Datei	44.470 KB
data3.rda	17.03.2020 11:46	RDA-Datei	8.879.677 KB
data2.rda	17.03.2020 11:21	RDA-Datei	692.975 KB
data0.rda	17.03.2020 11:17	RDA-Datei	223.454 KB

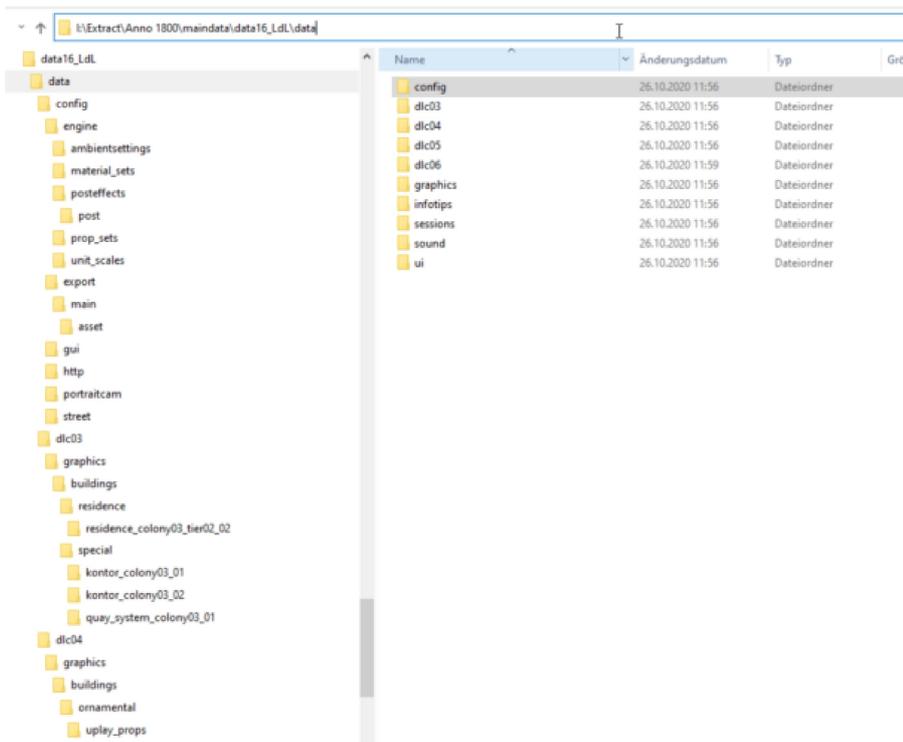
Ich habe die ganzen *.rda (ja alle! - weil ich wissen wollte, was drin ist ;-)) jeweils mit dem RDA-Explorer geöffnet (File - File Open Read Only). Es erscheint dann im RDA-Explorer das "root"-Verzeichnis der RDA-Datei (z.B. bei der data16.rda heißt es "data").



Man kann im RDA-Explorer die Inhalte browsen, ich habe jedoch zunächst den Inhalt der Datei in ein entsprechendes Verzeichnis entpackt (mit Tools - Extract all).



Dadurch kommt man an alle Dateien, die in der jeweiligen data*.rda liegen. Im entsprechenden Verzeichnis der entpackten Daten findet man jetzt alle Dateien aus der *.rda in der Verzeichnisstruktur, so wie sie im RDA-Explorer auch angezeigt werden. Ich habe die Dateien jeweils auf Laufwerk I: in einen Ordner entpackt, der so heißt wie die RDA-Datei. Also liegt die data16.rda entpackt im Verzeichnis I:\data16\



In unserem Fall ist fürs das "einfache" Modden eine Datei ganz besonders interessant. Diese liegt in meinem Fall im Verzeichnis [I:\data16] \data\config\export\main\asset und heisst - taataa: assets.xml. Die ist bei mir 154 MB groß. Bei mir liegen noch zwei andere xml-Dateien in diesem Verzeichnis (properties.xml. und templates.xml, sowie die checksum-Dateien für alle drei xml's).

Nun sind wir für das einfache Mod-Erstellen gut vorbereitet. Sagen wir mal, wir wollen (das war mein erster Mod-Versuch) die Lagerkapazität für die Öl-Depots in der Alten und Neuen Welt, sowie im Land der Löwen auf 2500 erhöhen.

Im ersten Schritt starten wir den AssetViewer (natürlich in der aktuellen Version) und suchen die GUID (die eindeutige Nummer, die das Item/Asset identifiziert) der zu ändernden Assets. Da es unterschiedliche Welten gibt, in denen das Öldepot gebaut werden kann, ist das Öldepot in jeder dieser Welten mit einer eigenen GUID vertreten. Der AssetViewer verrät uns hierzu, wenn wir nach dem "Gegenstand" "Gebäude" mit dem Wort "Öllager" im "Suchen"-Feld nachschauen, dass es drei Öllager gibt: Jeweils mit dem Namen "Öllager" für die Neue Welt mit der ID 101330, für die Alte mit der ID 100784 und für das LandOfLions mit der ID 119034.



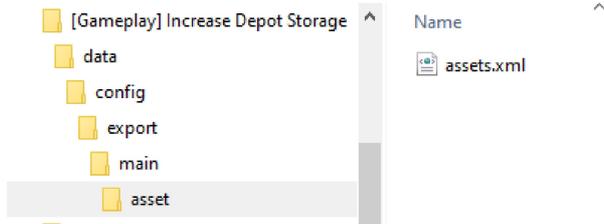
So weit so gut.

Nun gibt es zwei Wege, daraus einen erfolgreichen MOD für's Öllager zu machen:

Der erste (den ich auch zunächst experimentell einschlug) besteht einfach in der Anpassung eines bestehenden MOD's. Ich nahm für diesen Zweck den MOD für's Depot in "[Cheat] Increased Storage]", der zu diesem Zeitpunkt noch in der github Anno Collection unter <https://github.com/anno-mods/Collection> zu finden war.

Schauen wir uns diesen MOD an.

der Mod hat genau eine Datei mit dem Namen assets.xml im Verzeichnis [Gameplay] Increase Depot Storage\data\config\export\main\asset. Erste Erkenntnis: Ein MOD hat also den selben Aufbau wie die Dateistruktur des entpackten rda-Files. Zweite Erkenntnis: Die Datei in der der MOD steckt, heisst genau so wie die entsprechende Datei im entpackten rda-File: assets.xml.



Der Inhalt der MOD-Datei für die Erhöhung des Depot-Speichers sah so aus:

```
ModOps>
  <ModOp Type="merge" GUID="1010519" Path="/Values/Warehouse/WarehouseStorage">
    <StorageMax>2500</StorageMax>
  </ModOp>
</ModOps>
```

Erste Hinweise was das im Einzelnen bedeutet erhielt ich aus den Erläuterung von xforce zu seinem MOD-Loader unter <https://github.com/xforce/anno1800-mod-loader>.

Hier stehen (unten auf der Seite <https://github.com/xforce/anno1800-mod-loader#how-to-write-a-modop>) einige Erläuterungen zu den XML-Nodes und deren Bedeutung:

How to Write a ModOp

Step 1) Look up and select the XML node you want to edit with XPath using the Path argument.

Example:

```
<ModOp Path = "/Templates/Group[Name = 'Objects']/Template[Name = 'Residence7']/Properties">
```

For the assets file, you can also use the GUID argument. This selects all the child nodes of the asset with the given GUID as new roots for your XPath for cleaner code and is also much faster, performance-wise.

Example:

```
Standard way:      <ModOp Path = "///Asset[Values/Standard/GUID = '1137']/Values/Standard/Name">
Better, with GUID arg: <ModOp GUID = '1337' Path = "/Values/Standard/Name">
```

Step 2) Give a type for a ModOp, to change the selected node.

Currently supported types:

- Merge Replaces all given child nodes or Arguments
- Remove Removes the selected Node
- Add Adds inside the selected Node
- Replace Replaces the selected Node
- AddNextSibling Adds a sibling directly after the selected node
- AddPrevSibling Adds a sibling directly in front of the selected node

This was just a quick initial implementation (~3h), very open for discussions on how to make that better or do something entirely different

Step 3) Add the XML code that you want to have added, merged or as replacement inside the ModOp. example:

```
<ModOp Type = "replace" GUID = '1337' Path = "/Values/Standard/Name">
  <Name>ThisIsATestNameForGUID1337</Name>
</ModOp>
```

This ModOp will replace the node under /Values/Standard/Name of the asset with GUID 1337 with:

```
"<Name>ThisIsATestNameForGUID1337</Name>"
```

Ich dachte mir also beim Betrachten des "Depot-Mods", dass offensichtlich das Hafen-Depot die GUID 1010519 hat und das es in der Definition des Depots einen Wert StorageMax gibt, der bestimmt, wieviel Lagerkapazität das Depot hat. Also einfach die GUID fürs Öllager an die Stelle der GUID des Depots für die Alte Welt und die Lagerkapazität entsprechend auf den Wunschwert geändert und das ganze sah so aus:

```
<ModOps>
  <ModOp Type="merge" GUID="100784" Path="/Values/Warehouse/WarehouseStorage">
    <StorageMax>2500</StorageMax>
  </ModOp>
</ModOps>
```

In einer Datei assets.xml gespeichert und in die entsprechende Verzeichnisstruktur [Gameplay] Increase OilDepot Storage\data\config\export\main\asset gepackt, wanderte das Ganze ins MOD-Verzeichnis meiner Anno 1800 -Installation und -funktionierte! So weit so gut. Im nächsten Schritt packte ich die GUIDs der Öllager der Neuen Welt und dem LoL auch noch dazu (zur Erinnerung: die hatte mir der AssetViewer verraten) :

```
<ModOps>
```

```

<!-- Oil-Storage Alte Welt -->
<ModOp Type="merge" GUID="100784" Path="/Values/Warehouse/WarehouseStorage">
  <StorageMax>2500</StorageMax>
</ModOp>
<!-- Oil-Storage Neue Welt -->
<ModOp Type="merge" GUID="101330" Path="/Values/Warehouse/WarehouseStorage">
<StorageMax>2500</StorageMax>
</ModOp>
<!-- Oil-Storage Land of Lions -->
<ModOp Type="merge" GUID="119034" Path="/Values/Warehouse/WarehouseStorage">
  <StorageMax>2500</StorageMax>
</ModOp>
</ModOps>

```

(das Ergebnis kann man hier auch nochmal nachvollziehen:

<https://github.com/JohnSpoc/Collection/tree/patch-1/%5BCHEAT%5D%20Increase%20Oil%20Storage>)

Der experimentelle Ansatz war also erfolgreich - fürs Erste. Glück gehabt, dass das Lager für Öl die gleiche Struktur in der beschreibenden XML-Datei aufweist, wie das ‚normale‘ Depot.

Doch warum hat das funktioniert, war meine nächste Frage, die mich zum zweiten Weg, dem geplanten Ansatz, führte. Es musste also irgendwo Item- oder Asset-Definitionen geben, die Anno 1800 sagen, wie viel Holz in ein normales Lagerhaus geht, wie weit ein Rathaus seinen Einflussradius hat oder wie viel schneller die legendäre Verbesserung "Flinker Flaschenzug" arbeitet, wenn sie statt mit 100% mit 1000% arbeitet (Warum ausgerechnet der Flinker Flaschenzug?

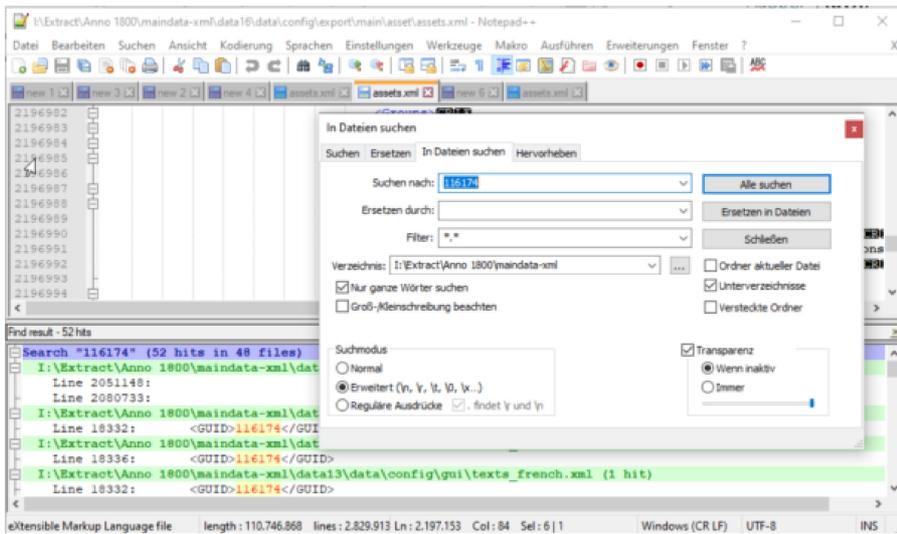
Deshalb: <https://www.annozone.de/forum/index.php?page=Thread&postID=273840#post273840>

Der "Erfreulich Flinker Flaschenzug" ist eine Verbesserung, die in Luftschiffen genutzt werden kann. Laut Assetviewer hat der die GUID 116174 und beschleunigt das Laden um 100%.



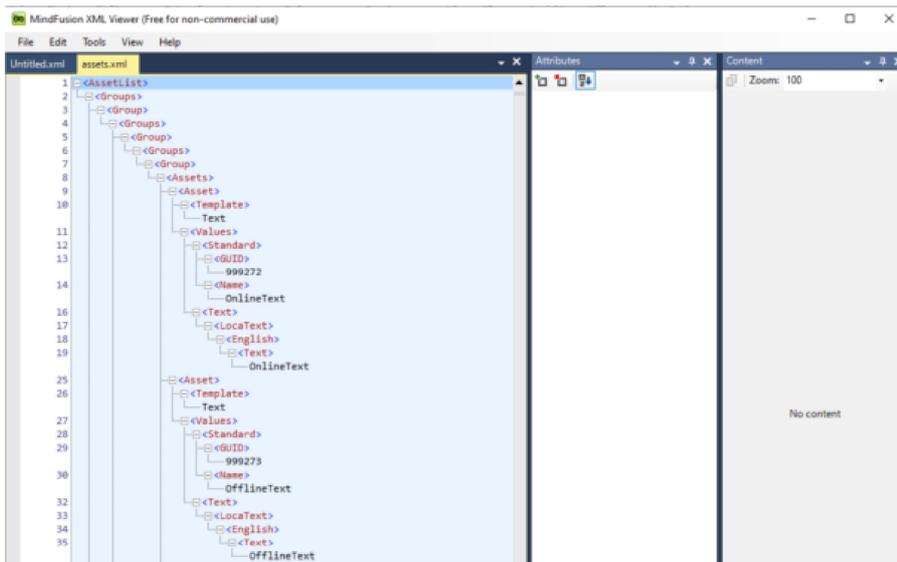
Aber welche Werte (Values) besitzt das Item und wie kann man sie modden? Zumindest muss es einen Eintrag für die Erhöhung der Bewegungsgeschwindigkeit um 20%, der Ladegeschwindigkeit um 100%, den Expeditionsbonus „Kampfkraft“ von 25 und den Verkaufspreis geben. Diese Werte müssten also "in der Nähe der GUID 116174 im betreffenden XML-File zu finden sein.

Da ich alle rda-Dateien ja schon entpackt hatte, suchte ich im Verzeichnisbaum der entpackten Daten nach xml-Dateien in denen die Zeichenkette "116174" vorkommt (das kann z.B. sowohl Notepad++ als auch Ultra Edit). Als Ergebnis wird einem angezeigt, dass die Zeichenkette 52 mal in 48 von allen über 11.600 insgesamt aus den rda-Dateien entpackten xml-Dateien vorkommt.



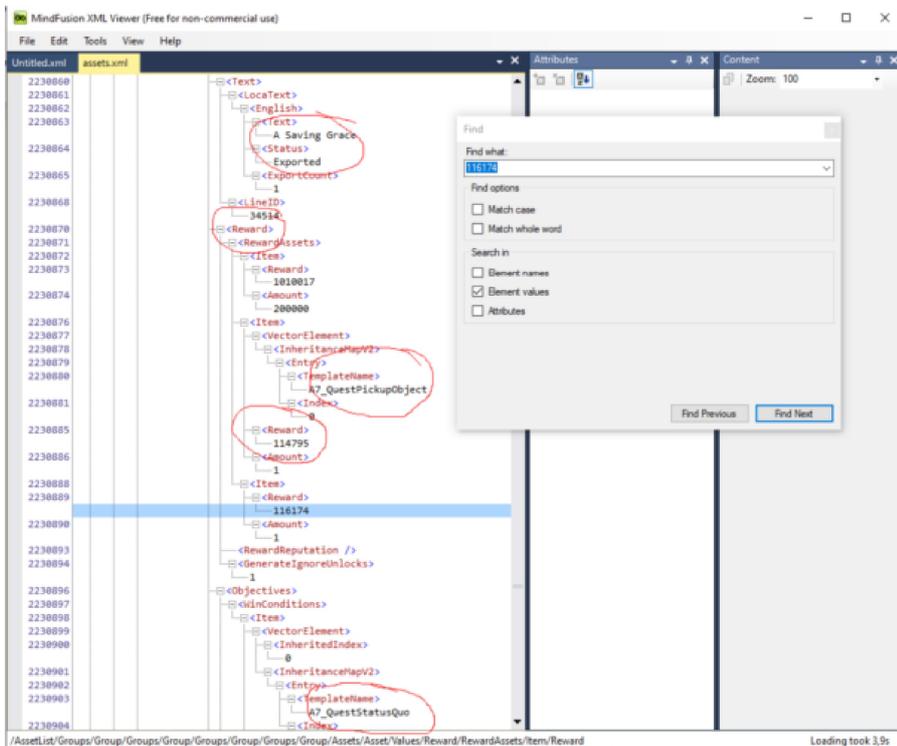
Ein paar davon kann man sofort ausschließen, wenn man eine geöffnet hat: die Dateien mit den Übersetzungen, die Namen wie texts_polish.xml haben. Hier stehen die Übersetzungen in die jeweilige Sprache drin, aber keine weiteren Definitionen. Als interessante Verdächtige bleiben die "großen" Dateien mit dem Namen assets.xml, die sich in den Verzeichnissen der großen rda-Dateien finden. Wenn man die älteren mit der neuesten assets.xml (z.B. die aus der data14.rda mit der aus data16.rda) z.B. mit Hilfe von BeyondCompare oder WinMerge vergleicht, stellt man fest, dass sehr viele Inhalte identisch sind, die data16.rda offensichtlich eine Datei mit vielen Ergänzungen ist und es einen Haufen von GUID's (neben denen die der AssetViewer ausweist) gibt.

Öffnet man die assets.xml aus der data16.rda mit dem o.a. xml-viewer erhält man die Inhalte in der Gruppen / Baumstruktur der Datei. Das ist beim Stöbern nützlich, da man Tags und Gruppen ein- und ausklappen kann und somit die Übersichtlichkeit gewinnt, die man braucht.

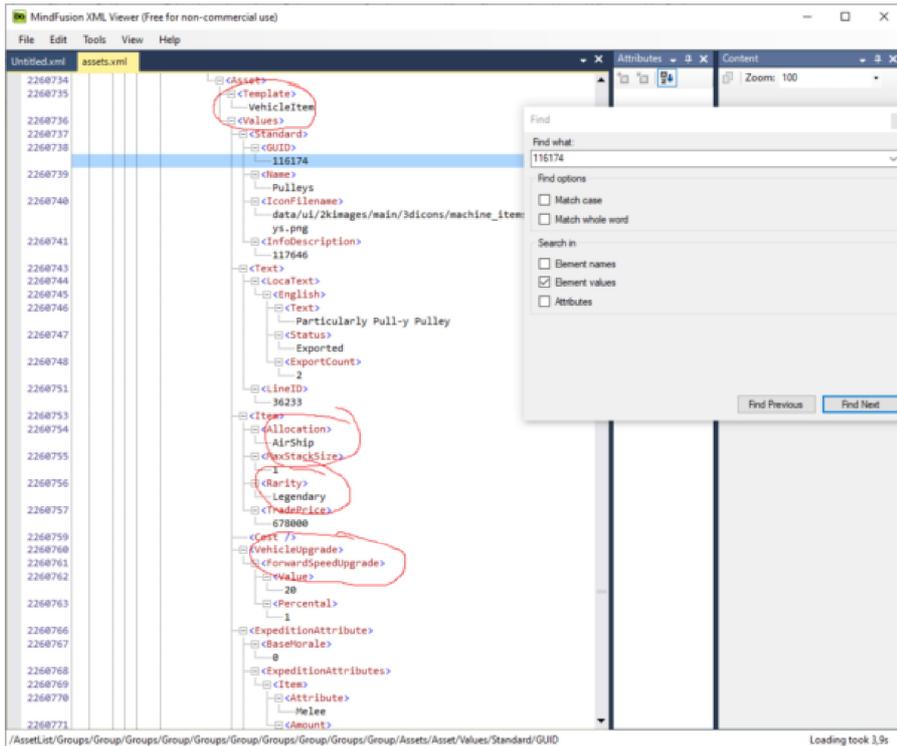


Der Interessierte mag gerne stöbern - für die eiligen "HowTo MOD-Interessierten" kurz: in der Datei nach dem Text der GUID des Flaschenzugs suchen (116174).

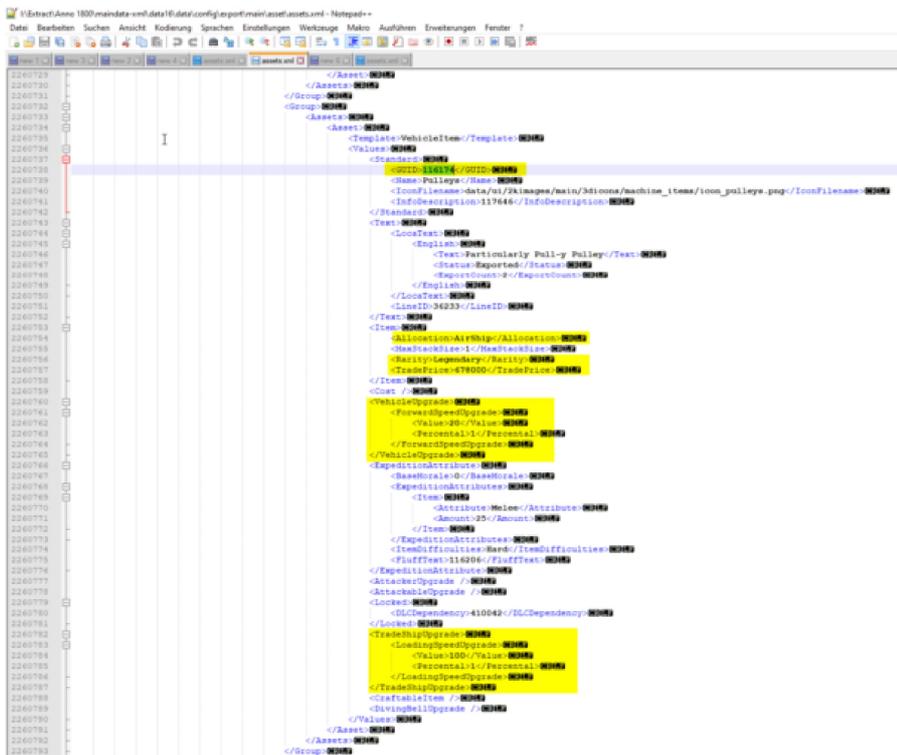
Die GUID findet man an zwei Stellen. In der Zeile 2230889, in einem Bereich in dem offensichtlich der Ablauf von Quests definiert wird (keine Ahnung, was da alles genau eingestellt wird, aber die Zeilen um die Fundstelle legen diese Annahme nahe - schaut selber nach;-))



und in der Zeile 2260738. Volltreffer - in diesem Bereich gehts offensichtlich um die Definition der Eigenschaften des Items mit der GUID 116174:



In der Ansicht des Notepad++ ist das dann „komprimierter“ dargestellt (dafür fehlt die schöne Baumstruktur, die hier unübersichtlicher dargestellt wird – man kann nicht alles haben 😞):



Einige Definitionen sprechen für sich: Allocation Airship legt wohl fest, dass das Item nur im Luftschiff eingesetzt werden kann. Zusätzlich zur Erhöhung der Geschwindigkeit um 20 % (ForwardSpeedUpgrade) gibts auch noch ein Upgrade der Ladegeschwindigkeit um 100 %. Will man jetzt in einem MOD die Eigenschaften ändern gilt es den XML-Baum richtig zu lesen: der Wert für das Upgrade der Ladegeschwindigkeit steckt im Tag <Value>100</Value>. Der Wert steckt im Pfad (von Innen nach Außen anhand der Ebenen im XML gelesen) <LoadingSpeedUpgrade> → <TradeShipUpgrade> → <Values>. Für die Pfandangabe im Mod-File muss man die Reihenfolge von außen nach innen lesen: also Values - TradeShipUpgrade - LoadingSpeedUpgrade.

So ergibt sich für das MOD-File die Anweisung:

„Ersetze den Wert im Value-Tag der GUID 116174, den man im Pfad Values/TradeShipUpgrade/LoadingSpeedUpgrade findet durch 1000“

oder in xml:

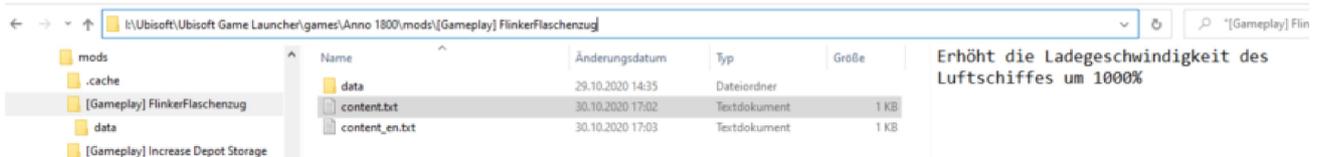
```
<ModOp Type="merge" GUID="116174" Path="/Values/TradeShipUpgrade/LoadingSpeedUpgrade">
  <Value>1000</Value>
</ModOp>
```

das wird schön in ein <ModOps></ModOps> eingepackt, wie es sich für MOD-xml's gehört und für die Erinnerung noch ein wenig kommentiert und dann so:

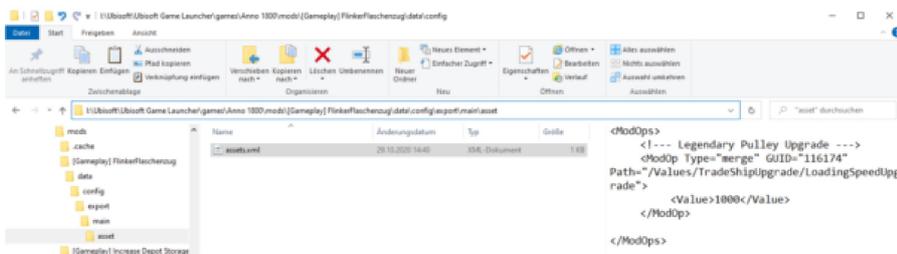
```
<ModOps>
<!-- Legendary Pulley Upgrade --->
<ModOp Type="merge" GUID="116174" Path="/Values/TradeShipUpgrade/LoadingSpeedUpgrade">
  <Value>1000</Value>
</ModOp>
</ModOps>
```

als assets.xml in einem Mod-Verzeichnis (denkt dran: die Verzeichnisstruktur sieht immer so aus: [MOD-Name]\data\config\export\main\asset\) gespeichert. Fertig ist der MOD.

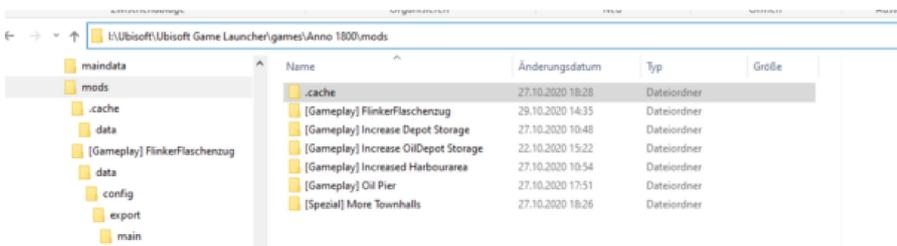
Für die ganz genauen (und für die, die wollen, dass Andere sehr schnell erkennen können, was der MOD macht) wird dann noch im „root“-Ordner des MODS eine content.txt angelegt (oft findet man auch eine content_de.txt oder content_en.txt), in der dann eine kurze Beschreibung des MODS steht:



Kopiert man dann den MOD ins Spielverzeichnis



Löscht den .cache -Ordner der bisherigen MODs unter



Steht dem Test im Spiel nicht entgegen

Spiel starten und voila:



Viel Spaß beim selber suchen nach den passenden GUID's und modden 😊

Wer zu lesen versteht, besitzt den Schlüssel zu großen Taten, zu unerträumten Möglichkeiten.

Aldous Huxley