

# NTF EDITING GUIDE



<b>1. NOTES</b>	<b>2</b>
Vorwort	2
Programme und Tools	2
<b>2. MISSIONSSTANDARDS</b>	<b>3</b>
I. MISSIONSARTEN	3
Zeus-Missionen	3
Cooperative Missionen (COOP-Missionen)	3
Team vs. Team Missionen (TvT-Missionen)	3
II. PLANUNG	3
Auf welcher Karte findet die Mission statt?	3
Welche Fraktion wird gespielt?	4
Wer sind die Feinde?	4
Was ist der Zweck der Mission?	4
In welcher Umgebung und bei welchem Wetter soll die Mission stattfinden?	5
Welche Slots gibt es in der Mission?	5
Medicsystem und Respawn?	5
<b>3. EDITOR-STANDARDS</b>	<b>6</b>
III. IM EDITOR	6
Spielercharaktere	6
Headless-Client	7
Zeus	7
Simulation bei Objekten	8
Wettereinstellungen	9
<b>4. MISSIONSDATEI</b>	<b>11</b>
IV. GROBES LAYOUT	11
Missionsordner	11
Lokalitäten	12
description.ext	13
Respawn (Inhalt der description.ext)	13
V. DIE INIT DATEIEN	14
init.sqf	14
initPlayerLocal.sqf	14
initServer.sqf	15
VI. DAS LOADOUT SYSTEM	15
Loadout Script	15
Loadout System anwenden	15
Loadout Datei	16
<b>5. ANBIETEN DER MISSION</b>	<b>17</b>
VII. ABSCHLIEßENDE MAßNAHMEN	17
Mission zur PBO packen	17
VIII. FORUMSPOST	18
Missionspost erstellen	18
Name	18
Inhalt	18
IX. MISSIONSPRÜFUNG	20
Ticket erstellen / Prüfvorgang	20
Kalendereintrag	20
X. MISSION AUF DEM SERVER	21
Mission laden	21
Mission starten	21
<b>6. CHECKLISTE</b>	<b>22</b>

# 1.NOTES

## Vorwort

Du möchtest mit dem Missionsbau in ArMA 3 beginnen, und weißt nicht so ganz, worauf du achten sollst? Dieses Dokument soll dabei helfen, den Einstieg in den Missionsbau zu erleichtern und Standards zu definieren. Grundsätzlich gilt, fangt klein an! Je kleiner desto weniger kann schiefgehen. Aber keine Sorge, wenn man sich an diesem Dokument orientiert, ist es ganz einfach.

## Programme und Tools

Alle Tools die für den Missionsbau benötigt werden.

<b>Tool</b>	<b>Link</b>
Arma 3 Tools	<a href="https://store.steampowered.com/app/233800/Arma_3_Tools/">https://store.steampowered.com/app/233800/Arma_3_Tools/</a>
Notepad++	<a href="https://notepad-plus-plus.org/downloads/">https://notepad-plus-plus.org/downloads/</a>
NP++ SQF Syntax	<a href="https://www.armaholic.com/page.php?id=8680">https://www.armaholic.com/page.php?id=8680</a>
PBO Manager V1.4 Beta	<a href="https://www.armaholic.com/page.php?id=16369">https://www.armaholic.com/page.php?id=16369</a>

## **2. MISSIONSSTANDARDS**

### **I. MISSIONSARTEN**

#### **Zeus-Missionen**

Zeus-Missionen werden größtenteils von einem „Zeus“ (Spielleiter) im laufendem Spielbetrieb erarbeitet und können spontan angeboten werden. Dafür werden Zeus-Templates zur Verfügung gestellt. Man kann aber auch ein eigenes Template / Basis Mission bauen. Das erfolgt dann aber unter eigener Verantwortung.

#### **Cooperative Missionen (COOP-Missionen)**

Dies sind die „normalen“ Missionen, die vollständig im Editor erarbeitet und anschließend getestet werden. Dabei wird ausschließlich gemeinsam gegen KI gekämpft. Sie können nicht spontan angeboten werden und sollten mindestens zwei Wochen vor dem Starttermin im Forum angekündigt werden.

#### **Team vs. Team Missionen (TvT-Missionen)**

TvT Missionen sind sehr aufwendig zu bauen, da sie aufwendig gebalanced werden müssen. Das schließt meist auch die Verwendung von komplexen Scripts ein. Sie werden in der Regel als ein Events clanübergreifend angeboten.

<b>Missionstyp</b>	<b>Kürzel</b>
Zeus Mission	ZM
Cooperative Mission	CO
Team vs Team Missionen	TVT

### **II. PLANUNG**

#### **Auf welcher Karte findet die Mission statt?**

Hier sind folgende Punkte zu beachten:

- Wie ist das Gelände und was hat es für Konsequenzen?
- Welche Mittel stehen den eigenen Kräften zur Verfügung?
- Wie kommen die Mittel zur Geltung oder werden eingeschränkt?
- Wie passt das Einsatzgebiet zur Story?

## Welche Fraktion wird gespielt?

Die Wahl der eigenen Fraktion ist der größte Faktor für die Mission. Hier ist auf Authentizität zu achten. Das heißt, die Loadouts, Fahrzeuge, Waffen, Lage und Logistik sollten im Rahmen der Mission Sinn ergeben. Dabei ist der große Wissensschatz unserer Mitspieler nicht zu unterschätzen. Also: Fragt bei erfahrenen Spielern nach!

## Wer sind die Feinde?

Die Wahl der Feinde ist ein ähnlich großer Faktor, wie die Wahl der eigenen Fraktion. Hier sollte beachtet werden, dass man einen Feind wählt, der ebenfalls im Rahmen der Mission Sinn ergibt.

## Was ist der Zweck der Mission?

Ebenfalls sollte man sich Gedanken über den Grund der Mission machen. Ein militärischer Einsatz findet auch nicht einfach ohne Weiteres statt. Dabei hilft es, sich folgende Fragen zu stellen:

- Um welchen Konflikt handelt es sich?
- Wie kam es zu diesem Konflikt?
- Was ist bisher in diesem Konflikt passiert?
- Welche Gegnerstärke ist zu erwarten?
- Welche Art von militärischem Szenario möchte ich darstellen
- Will ich besondere Ausrüstung / Unterstützung hervorheben
- Will ich ein spezielles Script oder Funktionalität unseres Modsets benutzen?

## In welcher Umgebung und bei welchem Wetter soll die Mission stattfinden?

Die Umgebung und auch das Wetter sind wichtig für die Gefühle, die man bei den Spielern auslösen will (Immersion). Die Immersion ergibt sich aus der Vorgeschichte, der eigenen Lage, der Karte, dem Auftrag, Wetter und Uhrzeit. Die folgenden zwei Bilder zeigen z.B., dasselbe Motiv aber die Stimmung durch Wetter und Uhrzeit sind anders.



## Welche Slots gibt es in der Mission?

Aus der Wahl der eigenen Fraktion und dem Zweck der Mission ergibt sich, welche Slots in der Mission verbaut werden sollten. Je nach Missionsgröße ist die Anzahl der Slots sehr flexibel es empfiehlt sich jedoch um die ~20 maximal 30 Slots zu verbauen.

## Medicsystem und Respawn?

Der Schwierigkeitsgrad einer Mission wird maßgeblich durch das Medicsystem beeinflusst. Man hat dort über die Addonoptionen die Möglichkeit, an vielen Stellen etwas anzupassen. **Hier muss unbedingt auf eine gute Balance zwischen Risiko und Spielspaß geachtet werden!** Außerdem sollte man sich Gedanken machen, ob und wie Respawn in die Mission integriert werden soll.

Respawn ist kein Muss für die Mission, wenn er nicht zur Mission passt, jedoch sollte dann die Zuschauerfunktion ermöglicht werden.

Und damit sollte die Planungsphase abgeschlossen sein und es geht jetzt ab in den Editor.

## 3. EDITOR-STANDARDS

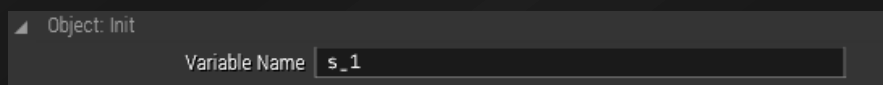
### III. IM EDITOR

Folgende Standards müssen bei Missionen unbedingt eingehalten werden!  
Hierbei handelt es sich um **das Minimum**, das eingehalten werden muss.

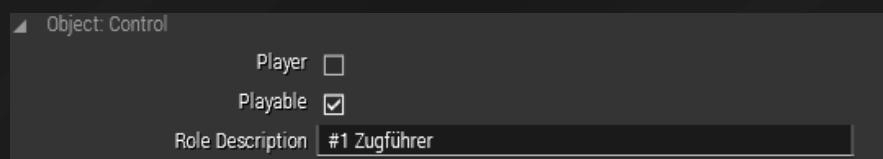
#### Spielercharaktere

Als Spielercharaktere sind Vanilla-Einheiten zu verwenden. So wird sichergestellt, dass CBA und ACE richtig arbeiten. Als Jetpiloten muss der Vanilla Pilot (B\_Pilot\_F) platziert werden. Zu erkennen an dem ACE-Icon neben der Einheit.

Jeder Spieler Slot muss in den Attributen einen Variablennamen bekommen. Empfohlen wird das Schema [s\_1, s\_2, s\_3, ..., s\_<n>] zu verwenden, da so beim Kopieren der Einheit automatisch der Name iteriert wird.

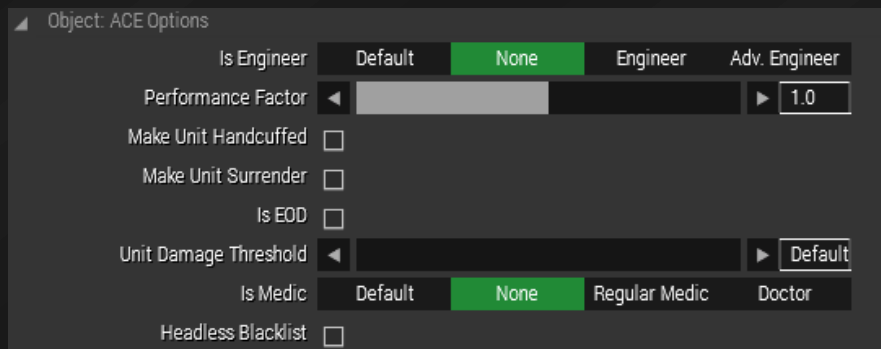


Des Weiteren muss jede Einheit als „spielbar“ markiert werden und in der Rollenbeschreibung muss die Nummer aus der Slotliste und der Name des Slots eingetragen werden.



*Tip: Wenn die Server Slotliste nicht in der richtigen Reihenfolge erscheint, könnt ihr die fertigen Slots ausschneiden (Strg + X) und wieder einfügen (Strg + V). So erscheint der Slot am Ende der Liste. Das wiederholt ihr solange in der richtigen Reihenfolge, bis keine Einheiten mehr über der ersten erscheinen.*

Um spezielle Rollen zu verteilen z. B. **EOD** oder **Medic**, müssen außerdem unter ACE Optionen die entsprechenden Einstellungen getroffen werden. Hier kann man ebenfalls die Einheit für den Headless-Client „Blacklisten“, was später wichtig wird.



## Headless-Client

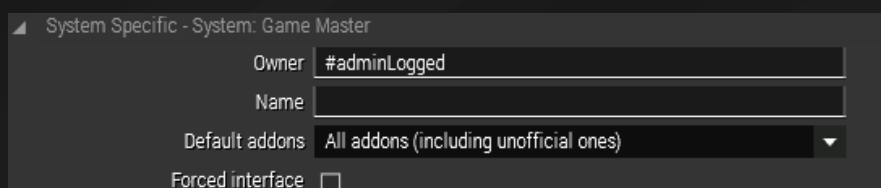
Der Headless-Client übernimmt die KI vom Server, um diesen zu entlasten.

Eine Mission muss den Headless-Client nicht benutzen, aber es muss **immer** ein Slot für den HC bereitgestellt werden. Grund dafür ist, dass wenn der HC keinen Slot hat, belegt er immer noch einen Platz auf dem Server. So kann ein Spieler weniger auf den Server verbinden. Wichtig zu wissen ist, dass KI-Scripte oder DAC im Zusammenspiel mit dem Headless-Client nicht funktionieren können. Um hier sicherzugehen, dass nichts schiefgeht, kann man die Einheiten „blacklisten“.

Das entsprechende Modul findet man unter Reiter F5 → Logik → virtuell → Headless Client. Wenn das Modul platziert ist muss man in den Attributen der Variablen Name hc1 eingetragen werden und als spielbar markiert werden.

## Zeus

Um Zeus benutzen zu können, muss man das „Game-Master-Modul“ platzieren und in den Attributen den Owner auf #adminLogged setzen und bei Default Addons All Addons auswählen.





## Simulation bei Objekten

Um die Performance in der Mission auf dem bestmöglichen Level zu halten, sollte man die Simulation der platzierten Objekte optimal einstellen.

**Keine Einstellungen** vorzunehmen führt zwangsläufig zu Performanceeinbußen. Lässt aber sämtliche Funktionalität für das Objekt zu, ohne sie auf die Entfernung zu Spielern zu begrenzen. **Objekt bleibt immer zerstörbar.**

**Dynamische Simulation** ist ein guter Mittelweg zwischen Funktionalität und Performance. Hier müssen aber auch noch zusätzliche Einstellungen zur Aktivierungreichweite vorgenommen werden. Außerdem muss bei KI-Gruppen eingestellt werden, ob sie die Dynamische Simulation, so wie Spieler, auslöst oder nicht. **Achtung! Objekt kann nur unter der eingestellten Entfernung zerstört werden!** (Wichtig bei Jets, Helikopter etc.)

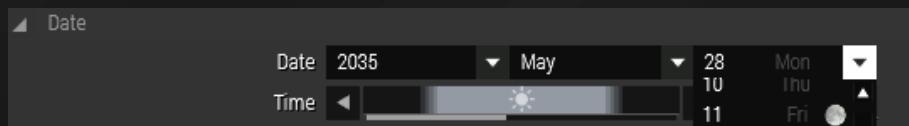
**Simple Object / (Keine Simulation)**. Mit einem simplen Objekt erzielt man die maximale Performance. Das Objekt verliert aber jegliche Funktionalität. Das Gleiche gilt für Objekte, bei denen die Simulation deaktiviert ist. Der Performancezuwachs ist hier aber geringer. Es gilt bei allen Objekten, bei den man die Simulation deaktiviert, sollte man, wenn immer möglich die Funktion „Simple Object“ ergreifen. **Achtung Objekt wird unzerstörbar!**



## Wettereinstellungen

### Datum

Hier wird das Datum Jahr, Monat, Tag und Uhrzeit eingestellt. Beachtet dabei die geografische Lage der Karte auf der Erde. So sind auf Tanoa im Juni die Tage kürzer und im Dezember länger. Für Nachtmissionen ist außerdem der Mond zu beachten. Neben der Auswahl des Tages wird gezeigt, ob Neu- oder Vollmond ist.



### Weather Forecast

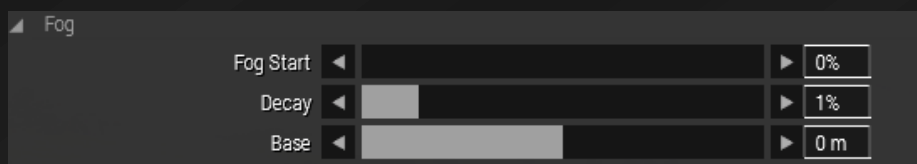
Diese Einstellung legt fest, wieviel Zeit vergehen soll, bis das Wetter vom Start- bis, zum Forecast-Wert gewechselt ist. Dieser Wechsel wird flüssig zwischen diesen beiden Werten im angegebenen Zeitraum interpoliert.

### Overcast

Hier wird der Bewölkungsgrad festgelegt. **Achtung!** Ab einem Wert über 50 % können Schatten schlagartig verschwinden und es kann, wenn kein Regen geforced wird, anfangen zu regnen.

### Fog (Nebel)

Die ersten drei Einstellungen sind hier für den Startnebel,



- Fog Start: Stellt die allgemeine Nebeldichte ein
- Decay: Stellt die Stärke der Nebelgrenze ein. **Achtung** bei 0 % tritt der Nebel Bug auf! Also immer >1% einstellen.
- Base: Stellt die Höhe der Nebelgrenze ein



### **Rain (Regen)**

Hier stellt man die Stärke des Regen ein, wenn man „Manual Override“ aktiviert. Wenn man das nicht tut, wird der Regen automatisch über den Overcast gesteuert.

### **Lightnings (Blitze)**

Hier stellt man die Häufigkeit der Blitze ein, wenn man „Manual Override“ aktiviert. Wenn man das nicht tut werden die Blitze automatisch über den Regen und Overcast gesteuert.

### **Waves (Wellen)**

Hier stellt man die Höhe der Wellen ein, wenn man „Manual Override“ aktiviert. Wenn man das nicht tut, werden die Wellen automatisch über den Wind gesteuert. Die Wellen sind erst ab einer größeren Wassertiefe vorhanden und haben keinen Effekt auf Flüsse oder Tümpel.

### **Wind**

Hier stellt man die Windrichtung und Windstärke ein. **Wichtig!** „Gusts“ (Böen) Immer auf das **Minimum** stellen. Außerdem muss hier das Häkchen bei „Manual Override“ gesetzt werden, um den Wind-Bug zu verhindern!

## 4. MISSIONSDATEI

### IV. GROBES LAYOUT

#### Missionsordner

Im folgenden Abschnitt befassen wir uns mit der generellen Ordnerstruktur im Missionsordner. Dieser Ordner wird von Arma 3 automatisch beim Speichern der Mission angelegt und enthält die „*mission.sqm*“. Er ist meist unter folgendem Pfad zu finden:

```
C:\Users\<username>\Documents\Arma 3 - Other Profiles\<Arma3Profile>\mpmissions\
```

Beim ersten Öffnen wird man nur die „*mission.sqm*“ vorfinden. Bei dieser Gelegenheit direkt überprüfen, ob die Missionsdatei binarisiert ist. Wenn ja, dann muss man im Editor oben unter „Attribute“ ganz unten den Haken entfernen und die Mission erneut speichern. Jetzt muss man noch einige Dinge hinzufügen.

Die vorgegebene Minimum-Ordnerstruktur sieht man im folgenden Bild:

Name	Änderungsdatum	Typ	Größe
loadout	02.12.2020 21:31	Dateiordner	
scripts	02.12.2020 21:54	Dateiordner	
description.ext	02.12.2020 21:53	EXT-Datei	2 KB
init.sqf	02.12.2020 21:53	SQF-Datei	1 KB
initPlayerLocal.sqf	02.12.2020 21:48	SQF-Datei	2 KB
initServer.sqf	02.12.2020 22:00	SQF-Datei	1 KB
mission.sqm	02.12.2020 21:36	SQM-Datei	17 KB

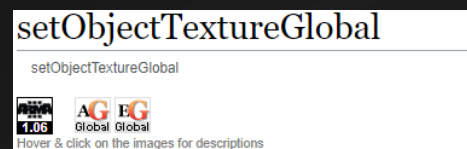
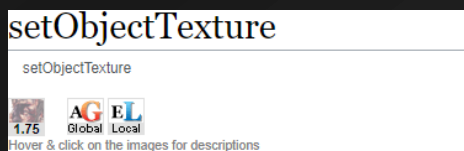
Folgende Dinge müssen also vorhanden sein:

- description.ext
- init.sqf
- initPlayerLocal.sqf
- initServer.sqf
- Ein „scripts“- Ordner
  - Das Loadoutscript
- Ein „loadout“- Ordner
  - Die Loadouts

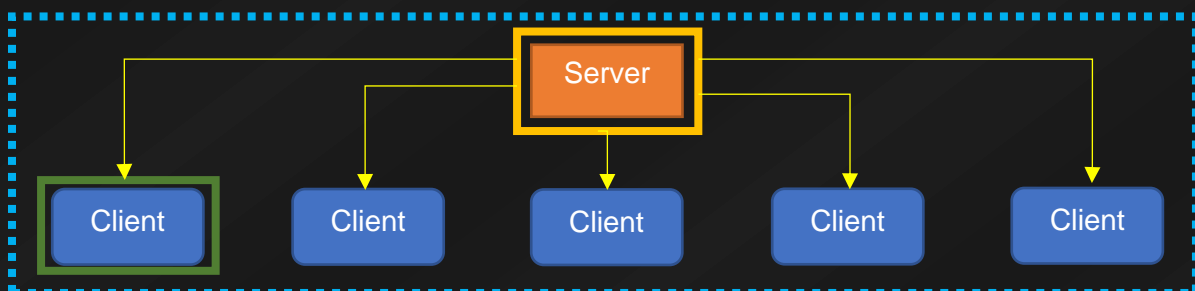
## Lokalitäten

Um die folgenden Dinge gut zu verstehen, sollte man ein Grundverständnis über die Lokalitäten im Arma 3-Multiplayer haben. Viele denken, die Lokalitäten in Arma seien schwer zu verstehen, aber im Grunde ist es ganz einfach, wenn man keine äußerst komplexe Scripte schreibt. Im Arma 3 Multiplayer gibt es **2 Ebenen** den **Server** und die **Clients**. Auf diesen Ebenen kann man auf drei verschiedene Weisen Scripte ausführen, „Client-Lokal“, „Server-Lokal“ und „Global“.

Dabei hilft die offizielle „Arma 3 Wiki“ enorm. Dort wird für die meisten Befehle angegeben, auf welcher Ebene sie arbeiten. Z. B. bei „setObjectTexture“ steht **AG** und **EL** für Argument Global und Effect Local.



- (Global)  
diese Ebene existiert nicht wirklich. Sie beschreibt nur, das jeder Client etwas lokal ausführt
- Server-Lokal  
Hier führt man Scripte aus, die globale Effekte haben z. B. „setObjectTextureGlobal“
- Client-Lokal  
Hier führt man Scripte aus, die lokale Effekte haben z. B. Texteinblendungen oder spielerbezogene Scripte mit globalen Effekten z. B. Loadout-Dateien



Mehr dazu: [https://community.bistudio.com/wiki/Multiplayer\\_Scripting](https://community.bistudio.com/wiki/Multiplayer_Scripting)

## description.ext

Dies ist die Konfigurationsdatei eurer Mission. Auf alles einzugehen, was hier eingestellt werden kann, würde den Rahmen dieses Dokuments sprengen.

Mehr dazu: <https://community.bistudio.com/wiki/Description.ext>

Folgende Eintragungen müssen in der Datei vorhanden sein:

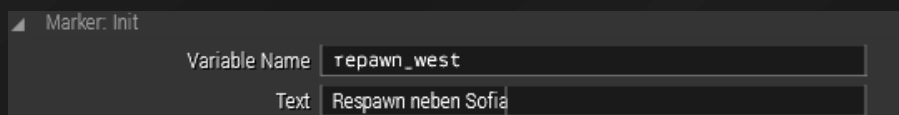
```
//----- onLoad -----  
  
author = "Weltbester Missionsbauer";  
onLoadName = "Megakrasse Mission";  
loadScreen = "unschlagbarerBanner.jpg"; // 16:9 .jpg oder 2:1 .paa  
  
//----- Header -----  
  
class Header  
{  
    gameType = COOP;  
    minPlayers = 1;  
    maxPlayers = 23; // Headless Client beachten  
};  
  
//----- Misc -----  
  
disabledAI = 1;  
enableDebugConsole = 1;  
disableChannels[]={0,true,true},{1,true,true}};
```

## Respawn (Inhalt der description.ext)

Wenn **Respawn gewünscht** ist, bitte folgenden Code verwenden:

```
//----- Respawn -----  
  
respawn = 3;  
respawndelay = 10;  
respawnTemplates[] = {"Counter"};  
respawnDialog = 1;
```

Zusätzlich muss auf der Karte einen Respawnmarker mit dem Variablennamen „respawn“ oder „respawn\_<Seitenname>“ (z. B. „respawn\_west für BLUFOR) platziert werden. Den entsprechenden Marker findet man unter Reiter F6 → Respawn → Respawn. Man kann aber auch jeden anderen Marker so benennen ohne, dass es zu Problemen führt.



Wenn **kein Respawn** gewünscht ist, bitte folgenden Code verwenden

```
//----- Respawn -----  
  
respawn = 3;  
respawndelay = 10;  
respawnTemplates[] = {"Counter"};  
respawnDialog = 1;
```

## V. DIE INIT DATEIEN

### init.sqf

Diese Datei wird bei Spielstart von allen Spielern, Headless-Clients und dem Server ausgeführt (auch bei JiP jedoch nur für den verbindenden Client). Deswegen hier bitte nur sehr grundlegende Dinge ausführen. Zu komplexe Scripte und Befehle mit globalen Effekten sollen hier nicht ausgeführt werden.

Folgendes muss vorhanden sein:

```
CHVD_allowNoGrass = false;  
CHVD_maxView = 2500; // Sichtweite in Metern  
CHVD_maxObj = 2400; // Objektsichtweite muss kleiner als Sichtweite sein
```

### initPlayerLocal.sqf

Diese Datei wird bei Spielstart (auch bei JiP) von allen Spielern lokal ausgeführt. Hier alle für den einzelnen Spieler relevanten Scripte ausführen z. B. Loadoutsript, Teleport und Ausrüster.

```
//=====   
==  
// LOADOUT  
  
private ["_loadout"];  
_loadout = player getVariable ["loadout", ""];  
  
if (_loadout != "") then {  
    [_loadout] execVM "scripts\loadout.sqf";  
};  
  
//=====   
==  
// TELEPORT  
  
tele addAction ["<t color='#d12313'>Teleport zu Bacon</t>", "(_this select 1) setPos position s_1", nil, 99, false, true, "", "];
```

```
//=====
// AUSRÜSTER

loadout addAction ["<t color='#3399ff'>Zugführung</t>", "execVM 'load-
out\crew.sqf';", nil, 99, false, true, "", ""];
```

## initServer.sqf

Diese Datei wird bei Spielstart vom Server ausgeführt. Hier alle Scripte ausführen, die einen globalen Effekt haben. Es kann sein, dass man die InitServer.sqf nicht benötigt. Aber sie soll trotzdem immer vorhanden sein, auch, wenn sie leer ist. Folgender Code dient nur als Beispiel zur Nutzung der initServer.sqf:

```
fill_funkbox = compile preprocessFile "boxen\funkbox.sqf";
sleep 0.25;
fill_granatbox = compile preprocessFile "boxen\granatbox.sqf";

nul = [funkbox_sb] spawn fill_funkbox;
nul = [granatbox_sb] spawn fill_granatbox;
```

*Auszug aus der initServer.sqf vom Übungsplatz. Dient zum Auffüllen von Ausrüstungskisten*

## VI. DAS LOADOUT SYSTEM

### Loadout Script

Um den in der initPlayerLocal.sqf eingefügten Teil des Loadoutscripts zu vervollständigen, benötigt man noch die „loadout.sqf“-Datei, die man in den „scripts“ Ordner speichert. In der Datei muss sich folgendes Script befinden:

```
private ["_loadout"];
_loadout = _this select 0;

if (_loadout == "") exitWith {};

private ["_script", "_handle"];
_script = format ["loadout\%1.sqf", _loadout];
_handle = execVM _script;
waitUntil {scriptDone _handle};
```

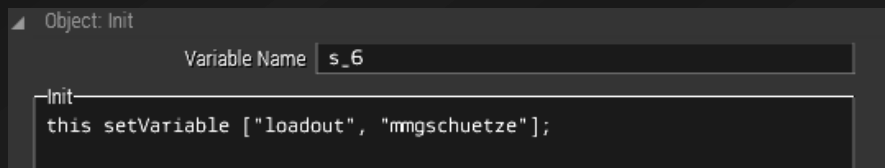
### Loadout System anwenden

Um von dem Loadoutsystem Gebrauch zu machen, muss man noch folgenden Code in das Init-Feld der spielbaren Einheit kopieren:

```
this setVariable ["loadout", "<loadoutDateiName>"];
```



Dort wo `<loadoutDateiName>` steht, trägt man den Namen der Loadout-Datei ohne die Endung „.sqf“ ein.



## Loadout Datei

Was jetzt noch fehlt, ist die Loadout-Datei selbst. Die bekommt man, indem man sie aus dem Vanilla Arsenal / Editor exportiert. Das kann man mit dem „Exportieren“ Button unten am Bildschirmrand oder mit „Strg + C“. Die Zwischenablage muss man dann einfach in ein leeres Textdokument kopieren.

**Achtung!** Bevor man die Datei benutzen kann, müssen noch einige Änderungen vorgenommen werden!

In der Datei muss der Teil mit „Unit must be Local“ entfernt werden und alle Aufrufe von „this“ in „player“ geändert werden.

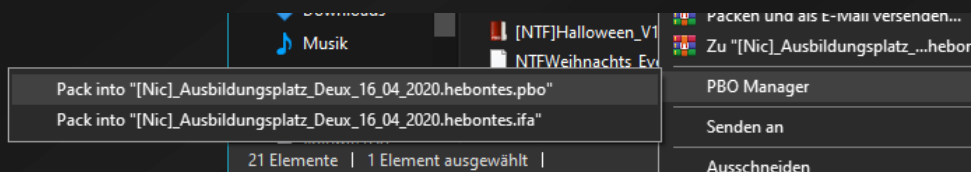
*Tipp: in Notepad++ könnt ihr mit „Strg + H“ die „Ersetzen“-Funktion aufrufen*

## 5. ANBIETEN DER MISSION

### VII. ABSCHLIEßENDE MAßNAHMEN

#### Mission zur PBO packen

Sobald man dann mit dem Bau der Mission endgültig fertig ist, packt man die Mission nun als \*.pbo Datei. Wenn man den PBO Manager installiert hat, findet man die entsprechende Funktion, indem man auf den Missionsordner im Windows Explorer rechtsklickt und über „PBO Manager“ hovert.



Wenn man die \*.pbo Datei hat, benennt man sie nach folgendem Schema.

[<Missionbauer Kürzel>]<Art><Slots>\_<Missionsname>\_<Version>.<Kartename>

Beispiel:

[Nic]CO24\_Operation\_Desert\_Storm\_V7.takistan

[Bear]CO20\_Unternehmen\_Sonnenblume\_V2.iron\_excelsior\_Tobruk

Euer Kürzel darf höchstens 4 Zeichen haben.

**Achtung!**

**WICHTIG!**

Achtet darauf, keine ö, ä, ü, ß, Leerzeichen oder sonstige Sonderzeichen zu benutzen!  
Es kommt ansonsten zu schweren Lade- / Initialisierungs-Fehlern und zum **Absturz des Servers**.

## VIII. FORUMSPOST

### Missionspost erstellen

Als erstes erstellt man einen Thread im Unterforum „Missionsankündigungen“. Keine Sorge, wenn man einen Post in diesem Unterforum schreibt und absendet ist er noch nicht sichtbar. Er muss erst vom Missionsprüfer freigeschaltet werden. Also kann man ruhig seinen Post absenden und später oder von einem anderen Gerät aus wieder bearbeiten und so weiter Schreiben, ohne jemanden zu stören 😊.

### Name

Der Name des Threads ist ähnlich, wie der .pbo Name aufgebaut jedoch in einem leserlichen Format.

<Datum> | [<Missionbauer Kürzel>]<Art><Slots> <Missionname>

Beispiel: 24.12.2020 | [Lft] CO24 Operation Merkur

### Inhalt

Folgende Dinge sollten in gleicher Reihenfolge im Missionspost vorhanden sein:

1. Ein Bild
  - Format: 1280 x 400 Pixel
2. Wann die Mission stattfindet
  - Datum
  - Start der Führungsbesprechung
  - Start der Slotphase
  - Verfall der Slotreservierungen
  - Missionsbeginn
3. Welcher Server
4. Welche Fraktion die Spieler spielen
5. Auf welcher Karte wird gespielt
6. Vorgeschichte
  - Hintergrund zur Mission. Hierzu nochmal Seite 4 & 5 beachten.

## 7. Die Lage

- Feind (Stärke, Art, Verhalten, Absicht)
- Eigene (Lage, Auftrag, Absicht)
- Zivilisten (Anwesenheit, Umgang)
- Wetter

## 8. Auftrag

- Was ist der Auftrag der eigenen Kräfte

## 9. Durchführung

- Die „geplante“ Durchführung des eigenen Auftrags
- Aufträge unterteilt nach Truppenteil. Welche Gruppe tut was?
- Maßnahmen zur Koordinierung (z. B. Anmarsch)

## 10. Einsatzunterstützung

- Welche Unterstützungen stehen den eigenen Kräften zur Verfügung?
- CAS, Mörser, Artillerie, Logistik etc.

## 11. Aufklärung

- Informationen zum Feind (Stellungen, Fahrzeuge, Routen, etc.)  
Der vorhergehende Punkt ist wichtig, um einen guten Missionsfluss zu ermöglichen.

## 12. Gerät

- Welche Art von Fahrzeugen / Ausrüstung stehen den eigenen Kräften zur Verfügung?

## 13. Slot Liste

- Die nummerierte Auflistung aller Slots der Mission
- **leere** Slots sind **grün** zu markieren
- **besetzte** Slots sind **rot** zu markieren
- **gesperrte** Slots sind **orange** zu markieren

## 14. Technisches

- Einstellung Medic-System
- Join in Progress (JiP) (ja/nein)

## IX. MISSIONSPRÜFUNG

### Ticket erstellen / Prüfvorgang

Wenn alles fertig ist, muss man ein Ticket in der Kategorie „Missionsprüfung“ entweder Intern oder Event eröffnen. Dem Ticket muss ein Link zum Forumseintrag und die Mission-PBO beiliegen. Außerdem kann man sein Wunschdatum zur Veröffentlichung der Mission oder andere Anmerkungen dazuschreiben. Die Mission und der Forumseintrag werden dann von einem Missionsprüfer geprüft.

Der Missionsprüfer wird anschließend eine Liste mit Fehlern oder Anmerkungen zur Mission und zum Forumseintrag verfassen. Wenn man die entsprechenden Anpassungen vorgenommen hat, muss man die Mission erneut zum Missionsprüfer schicken. Wenn dieser die Mission freigibt, wird sie auf den Server hochgeladen.

Im Anschluss muss spätestens jetzt ein Termin vereinbart werden, an dem die Mission stattfindet, sofern der Wunschtermin nicht angegeben wurde. Der Forumseintrag wird dann freigeschaltet.

Der Termin für eine reguläre Mission sollte so gewählt werden, dass die Mitspieler mindestens eine Woche Zeit haben, sich anzumelden.

### Kalendereintrag

Sobald man vom Missionsprüfer grünes Licht bekommen hat, muss ein Kalendereintrag erstellt werden, damit die Mitspieler eine einfache Übersicht über die kommenden Wochen haben.

Als **Kategorie** wählt man „Missionstermine“ aus.

Bei **Datum und Uhrzeit** stellt man den Starttermin der Mission ein.

Bei der **Benennung** des Kalendereintrags muss derselbe Titel wie für den Forumseintrag verwendet, jedoch lässt man das Datum weg, da es ja bereits im Kalender steht.

[<Missionbauer Kürzel>]<Art><Slots> <Missionname>

Beispiel: [Lft] CO24 Operation Merkur

Der **Inhalt** des Kalendereintrags muss ein Link sein, der direkt zum Forumseintrag der Mission führt.

## X. Mission auf dem Server

### Mission laden

Wenn man auf den Server verbindet, muss man sich als Admin einloggen, um seine Mission laden zu können

**Achtung!**

Wenn man das erste Mal auf den Server verbinden will und den „Map-Hintergrund“ sieht und man denkt der Server ist vielleicht aus, dann müsst ihr euch einloggen. Der Server ist an und ihr seid verbunden aber der Server hat noch nichts geladen.

Dafür mit der „-“ Taste den Ingame-Chat öffnen und mit dem Befehl `#login Kunstbanause` als Admin einloggen. Man bekommt keine Rückmeldung, ob es funktioniert hat oder nicht.

Als nächstes muss man mit dem Befehl `#missions` die Missionsauswahl öffnen und man kann die gewünschte Mission laden.

Beim Laden oben Links auf den **Schwierigkeitsgrad** achten. Dieser sollte normalerweise auf „**Custom**“ fest eingestellt sein. Dies ist auf dem WW2 Server nicht erforderlich.

Diesen **immer auf Custom stellen**.

### Mission starten

Vor dem Start der Mission ist die Anwesenheit zu prüfen und ggf. abwesende Personen sind im Forum aus der Slot Liste auszutragen. Dies ist wichtig für die Badgebeantragung.

Wenn man Start drückt, kommt man auf die Karte, auf der grundlegende Strategien besprochen und Zeichnungen für alle erstellt werden können. Wenn alle „Blue-Up“ sind, kann die Mission gestartet werden. **Achtung ab hier wird man in den ACRE Channel verschoben.**

Wenn die Mission dann gestartet ist, überprüft, ob ihr Zugriff auf Zeus habt. Sollte das nicht der Fall sein, geht zurück in den Slotscreen und spawnt neu.

Sollte beim Missionsbau etwas essentielles vergessen worden sein, z. B. Ohrenstöpsel oder Funkgeräte, ist das per Zeus nachzuliefern.

## 6. Checkliste

Kategorie	Punkt	Check
Planung	Missionstyp gewählt	
	Karte gewählt	
	Zweck der Mission gewählt	
Spielercharaktere	Spielercharaktere platziert	
	Variablennamen vergeben	
	Loadout erstellt und Variable vergeben	
	Rollenname vergeben	
Headless Client	HC Entität gesetzt	
	Entität den Variablennamen „hc1“ vergeben	
Zeus	Game Master Modul platziert	
	Besitzer auf #adminLogged gesetzt	
	Alle Inoffiziellen Addons freigegeben	
Performance in der Mission	Dynamische Simulation, wenn möglich verwendet	
	Reichweiten für dynamische Simulation eingestellt	
	dekorative Objekte zu Simplen Objekten gemacht	
	Garbage Collector eingestellt	
Missionsordner	Ordnerstruktur Ordnungsgemäß angelegt	
	description.ext mit Pflichteinträgen erstellt	
	Init.sqf mit Pflichteinträgen erstellt	
	initPlayerLocal.sqf mit Pflichteinträgen erstellt	
	initServer.sqf erstellt	
	„Scripts“ Ordner mit loadout.sqf erstellt	
	„loadout“ Ordner erstellt	
Ankündigung	.pbo Datei erstellt und nach Schema benannt	
	Formuseintrag erstellt und mit benötigten Infos gefüllt	
	Prüfticket mit PBO Datei und Link zum Thread eröffnet	
	Kalendereintrag erstellt und nach Schema benannt	