

stdout.txt - Erklärung & typische Fehlermeldungen

Inhaltsverzeichnis

- [1 Einleitung](#)
- [2 Allgemeines](#)
- [3 Arten von Crashes](#)
- [4 Typische stdout-Meldungen ohne Absturz](#)
- [5 Fehlermeldungen Liste](#)

Vielleicht noch ein passendes Zitat von [VacuumTube](#) zur Einleitung:

Zitat von @VacuumTube

Die stdout ist eine ungeordnete Sammlung von Logmeldungen (ohne Zeitstempel), die sowohl vom Spiel als auch von einzelnen Mods kommen.

Es gibt nicht den einen Weg, eine stdout richtig zu lesen. Je nach Problem, muss man sich das genau anschauen. Erst wenn man sich schon länger damit beschäftigt hat, versteht man den prinzipiellen Aufbau und die Bedeutung der Meldungen.

[Danto](#) Danke für das Erstellen. Ich hab den Artikel mal erweitert und unten die Liste angefangen, wie ich mir das etwa vorgestellt habe.

1 Einleitung

Dieser Lexikon-Artikel soll den Umgang mit Crashes/Fehlermeldungen vereinfachen. Bei der Fehlersuche geht es meist darum, die verantwortliche Mod zu finden. Dieser Artikel soll helfen, die Fehlerart und die Ursache anhand der **stdout.txt** festzustellen, welche für weniger erfahrene Spieler offenbar nur schwer zu durchschauen ist. Außerdem sollen in einer Liste (unten) bekannte Fehlermeldungen mit Erläuterung festgehalten werden. Diese ist auch für Modder bei der Fehlersuche interessant.

Dies soll aber keine Schritt-für-Schritt Anleitung sein, denn in vielen Fällen muss man einfach genauer hinschauen oder ist auf die Erfahrung der Modder hier im Forum angewiesen. Daher gilt, wenn euch die Informationen hier und die Fehlerliste nicht weiterhelfen: Schreibt euer Problem ins Forum. Achtet dabei darauf, genau zu beschreiben, wann/wie der Fehler auftritt und welche Mods ihr benutzt. Das wichtigste: immer die **stdout.txt** anhängen, denn manchmal stehen auch vor der eigentlich Fehlermeldung entscheidende Informationen.

2 Allgemeines

- Grundlagen/Speicherorte: [Speicherorte der Spieldateien, stdout.txt, Speicherstände, Mods](#)
- Die Datei wird mit jedem neuen Spielstart überschrieben
- Im selben Ordner befinden sich auch die Minidumps (.dmp) - diese können aber nur Urban Games lesen, wir können damit nichts anfangen
- Lest die **stdout.txt** am besten immer von unten nach oben - tendenziell steht unten das, was euer Spiel zum Absturz gebracht hat

- Die Zeilen kurz vor der Fehlermeldung können relevant sein, müssen aber nicht (siehe gewöhnliche Meldungen)
- Man sollte nicht anfangen, die stdout zu überinterpretieren. Am Ende stehen dort nur begrenzte Informationen, ggf. von Mods direkt, aber nur wenn sich der Modder entscheidet eine Information via `print(...)` zu loggen. D.h. oft sind die Informationen die dort stehen nicht ausreichend. Nur weil 2 Zeilen untereinander stehen, heißt es nicht, dass es einen Zusammenhang gibt. Denkt daran, dass es keine Zeitstempel gibt. Die nächste Zeile wurde ggf. 30min später geschrieben. Evtl. macht es daher Sinn, das Log mithilfe der Konsole während des Spiels zu beobachten.

3 Arten von Crashes

Ich unterscheide mal 3 grundlegende Möglichkeiten, wie das Spiel crashen kann.

Ihr solltet nach einem Crash mit der Fehlermeldung schnell merken, um was es sich handelt, somit lässt sich das Problem eingrenzen.

- **Lua Fehler:** Tritt auf, wenn im Lua-Code eine nicht-geschützte *Exception* auftritt, z.B. ein Null-Pointer-Fehler wegen fehlender Überprüfung. Die Ursache ist meistens ein Mod. Diese Fehler sind leicht zu analysieren, weil direkt die Fehlerzeilen in den Lua Dateien angegeben sind. z.B.: `stack traceback: error: .../local/mods/modxy_1/mod.lua:36: attempt to index global 'var1' (a nil value)`
- **Assertion:** Diese "Tests" sind von den Entwicklern in den Quellcode eingebaut, um fehlerhaftes Verhalten oder ungültige Zustände zu erkennen. Die eigentliche Ursache kann jedoch sehr viel früher entstehen. Grundsätzlich ist es ein Fehler im Spiel, aber auch Mods können verantwortlich sein. Das Spiel bricht sofort ab, um Folgefehler oder korrupte Savegames zu vermeiden. Im Log wird zusätzlich zur fehlgeschlagenen Assertion die Quelldatei und Zeile sowie die Funktion, in der der Fehler auftrat, angegeben. Wir können das nicht nachsehen, da der C++ Quellcode nicht öffentlich ist, aber man kann oft spekulieren, was das Problem sein könnte. Außerdem findet ihr in alten Threads oder in der Liste (unten) mehr Informationen, falls der Fehler bekannt ist.
- **Andere:** Leider passiert es auch manchmal, dass das Spiel ohne irgendeine Art von Fehlermeldung abstürzt. Erscheint "An error just occurred" oder "heap corruption" beendet das Spiel durch einen unerwarteten Fehler sofort, weil ein ungültiger Speicherzugriff/zustand erkannt wurde. Die Ursache ist dann sehr tief im Spiel selbst. Oder aber das Programm wird von außen gekillt (ist zB der Fall, wenn es nicht reagiert und man es zum Abbruch zwingt). In der stdout ist dann die letzte Zeile "normal" und jede Art von Fehlermeldung oder *Goodbye* fehlt, außerdem gibt es kein Minidump. Die Ursache ist nur schwer rauszufinden. Auch Hardwareprobleme sind möglich.

4 Typische stdout-Meldungen ohne Absturz

Hier können wir "gewöhnliche" Meldungen sammeln, die zwar anzeigen, dass etwas nicht in Ordnung ist, aber in der Regel nicht der Grund für einen Crash sind.

- `texture load error:` Eine Textur kann nicht geladen werden. Im Spiel macht sich das durch eine lila Fläche bemerkbar, führt aber nicht zum Absturz
- `error loading dds file:` Eine dds Datei kann nicht geladen werden
- `vehicle/x.mdl is missing boundingInfo:` Im Modell ist keine boundingInfo angegeben
- `* x.mdl was removed because the following resources were missing:` Fehlende Modelle werden durch Dummys ersetzt
- `[RESOURCE ERROR] Referenced model not found: 'x.mdl' in 'x.lua' (has been replaced with fallback):` Fehlende Spieldaten
- `WARNING: alutCreateBufferFromFile failed for "x.wav":` Zu viele Soundquellen, Sound wird nicht abgespielt
- `[ModUtil|UserSettings] Previous settings for id "x" are overwritten:` Doppelte id beim [Modutil](#) -> Mod doppelt geladen?

5 Fehlermeldungen Liste

Hier sollen alle bekannten Assertion-Fehlermeldungen inklusive vermutlicher Ursache gesammelt werden. Dies ermöglicht außerdem einen gewissen Einblick in den Aufbau und die Programmierung des Spiels.

Der Aufbau ist immer gleich. Die Datei im Quellcode, in der der Fehler auftritt, beginnt mit `c:\build\tpf2_steam` oder `c:\build\tpf2_gog` bzw. seit neustem mit `C:\GitLab-Runner\builds\...\ug\urban_games\train_fever\`, deshalb wird das weggelassen.

Im wiki gibt es [eine ähnliche Fehlerliste](#). Diese ist aber unvollständig und teilweise unklar.

Beachtet auch, dass zur selben Assertion verschiedene Ursachen führen können. Auch sind viele Assertions unbekannt oder nicht aussagekräftig.

Die Liste darf gerne von jedem, der neue Erkenntnisse hat, erweitert werden. Bei Unsicherheiten bitte Kommentar schreiben.

Quelldatei/zeile	Error message / Assertion ... failed
	Error message: cannot open ../mod.lua: No such file or directory
\src\lib\modelio\converter_model.h:179	invalid material type: REFLECTIVE_NRM_MAP_OP
\src\lib\util\idrep.h:84	false
\src\game\terrain\renderdatamanager.cpp:806	m_subscriptions.empty()
\src\Game\Terrain\render_data\MaterialData.cpp:46	detailTextures.size() < 255
\src\game\application.cpp:637	false / Out of memory
\src\lib\renderer\tex_load.cpp:343	img.width % 2 == 0 && img.height % 2 == 0
\src\lib\renderer\tex_load.cpp:528	numBits == 8 numBits == 16 numBits == 24 numBits == 32 numBits == 64
\src\lib\renderer\tex_load.cpp:821	redGreen == (format.Internal == gli::gl::INTERNAL_RG_ATI2N_UNORM)
\src\lib\renderer\texturemanager.cpp:105	it != m_textures.end()
\src\lib\renderer\model\DynamicModelRenderer.cpp:215	mesh->groups.size() == materials.size()
\src\lib\renderer\model\staticbuffer.cpp:507	vbo > 0

Quelldatei/zeile	Error message / Assertion ... failed
\\src\\lib\\renderer\\model\\vaomanager.cpp:112	m_meshes->Get(meshId)- >va2vertexAttr[va.index].numComp == va.size * va.arrSize
\\src\\lib\\model\\edge_geometry_util.cpp:123	result.length > .0f
\\src\\lib\\model\\edge_geometry_util.h:215	len > .0f
\\src\\lib\\model\\edge_geometry_util.cpp:216	false
\\src\\lib\\ui\\combobox.cpp:174	index < (int)m_createItemFns.size()
\\src\\lib\\ui\\core.cpp:1446	pr.second && "Duplicate id found."
\\src\\Lib\\Geometry\\Streets\\Transitions\\transition_util.cpp:125	angle >= .0f && angle <= Math::PI
\\src\\Game\\construction\\apply_proposal.cpp:440	proposalData.errorState.Empty() g_allowApplyWithErrorsHack
\\src\\game\\construction\\make_proposal.cpp:692	(int)construction.stocks.size() == (int)r.size
\\src\\Game\\construction\\street\\ProposalStreetGraph.cpp:54	!Contains(m_proposal->removedNodes, entity)
\\src\\game\\init_game_util.cpp:206	con.simBuildings.size() == 1
\\src\\game\\procedural\\buildingtyperep.cpp:66	false
\\src\\game\\procedural\\buildingtyperep.cpp:373	it != m_buildingTypes.end()
\\src\\game\\procedural\\trackgenerator.cpp:120	(int)trackType.trackStraightModel.size() == 4
\\src\\game\\terrain\\groundtexturerep.cpp:131	false

Quelldatei/zeile	Error message / Assertion ... failed
\\src\\game\\transport\\street\\streetshapefactory.cpp:931	Valid(result.laneConfigs)
\\src\\game\\transport\\street\\streettyperep.cpp:226	false
\\src\\game\\transport\\cargotyperep.cpp:117	int(index) >= 0
\\src\\game\\transport\\vehicle_cargo_util.cpp:160	it != format2model.end()
\\src\\game\\transport\\path_util.cpp:174	dist < .0f
\\src\\Game\\transport\\vehicle_util_3.cpp:840	nextEdgeId.entity == edgeEntity
\\src\\game\\urbansim\\pathfinder_util.cpp:365	itCache != m_station2stationsAndDistances.end()
\\src\\game\\urbansim\\parcel_util.cpp:1153	face[0] != face[1]
\\src\\game\\ui\\components\\addmodulecomp.cpp:299	it.second
\\src\\game\\ui\\util\\consumersupplierdataprovder.cpp:112	it != result.end()
\\src\\lib\\ecs\\engine.h:275	it != components.end()
\\src\\Lib\\ecs\\Engine.cpp:660	!m_notificationInProgressDEBUG
\\src\\game\\ecs\\trainmovesystem.cpp:666	mp.speed >= .0f
\\src\\Game\\ecs\\SimPersonAtVehicleSystem.cpp:97	spav.place >= 0 && spav.place < (int)vehiclePartInfo.seats.size()

Quelldatei/zeile	Error message / Assertion ... failed
\src\Game\ecs\AircraftMoveSystem.cpp:738	mp.aircraftState.flightState == component::MovePathAircraft::AircraftState::LA
\src\game\ecs\simpersonsystem.cpp:1823	fromEdgesDrive.empty() !toEdgesDrive.empty()
\src\Game\Game.cpp:349	m_data->gameStates[1]->ScriptSave() == m_data- >gameStates[0]->ScriptSave() it->second.second == 1

Die Funktion ist jetzt rechts (4. Spalte) und hat die kleinste Schriftgröße mit Tahoma, um nicht so viel Platz wegzunehmen. Spoiler wäre perfekt, aber innerhalb einer Tabelle leider nicht möglich.