

ut_console

User Manual

Contents

Introduction.....	1
Expression Syntax Explanation.....	1
=> Object References.....	1
=> Root Properties.....	2
Command Overview.....	3

Introduction

This Mod is aimed at Mod Developers to Test functionality and confirm values that usually need a Script change or a dedicated script to get. Additionally it might turn out to also be a useful Cheat Mod, but that is not the primary goal, so be careful when used as such. This Mod is versatile and powerful, and can mess up many things in a Savegame.

Expression Syntax Explanation

Expressions in this Console are very similar to the Expressions in the Scripting Languages (aiscript and md) since they are based on them, though there are some differences and peculiarities which will be pointed out.

Object References

Objects can be referenced with many methods. This can happen at the start of a property chain, or even in the middle of it if it makes sense in context. Possible ways to reference Objects are:

ABC-123	Its idcode. Note that i consider the idcode non-unique, but its easy to use. Will use the first match, but output an error if other matches are found.
0x1234 (hex id)	Its Object identifier, that is unique within a game session. Note that its not persistent over sessions.
12345 (decimal id)	(might change when loading a game) Can be entered in any supported numeric form.
\$console_var	If you set a console Var before you can use it as Object reference if it contains an Object.
ptarget	player target
gtarget	guidance target
atarget	player autopilot target
pcontainer	player container (ship or station)
pobject	
htargets	Holomap Targets (Not yet used by any command)
mapobjects	Holomap Visible Objects (Not yet used by any command)

Root Properties

The following Properties (and enforced start properties) are available:

Object references	see above for explanation
\$console_var	If you set a console Var before you can access it as root property
null	simply a null if you need it
True	actually numeric 1
false	numeric 0 in this case
component.<number> object.<number>	The Component/object referenced by the following numeric identifier. Use this if the autodetect fails, or if you want to start an expression chain with such an object (no autodetect here)
gameparam.<param>	Reading from the game params (Startup params?) - not sure yet what this is..
param.<groupname>.<paramname>	reading scriptparams data from libraries/parameters.xml
macro.<macroid>	getting a macro via its id as starting point
md.<scriptname>.<cuename>	getting an MD cue via its Script and Cuename. Can be used further for remote var reading
mdvar,\$varname	getting an MD var via its name. Equal function to md.\$varname in the game scripts.

Command Overview

Currently the following commands are implemented: (a detailed explanation follows)

Evaluating one or more Expressions:

&eval \$exp_1 *\$exp_2* ...

Setting a Value at the end of a Property Chain/Expression:

&setval \$exp_set \$exp_get1 *\$exp_get2*

Instantly teleporting an Object into another Space or near another Object:

&warp \$exp_object \$exp_destination

Spawning a Ship for and generating a Report about it for Ship Modding Purposes:

&checkship \$macro_id_or_exp *crewskill.<number> paintcycletime.<number> ware.paintjob_#1 ware.paintjob_#2*

Destroying an arbitrary Object (requires additional confirmation; will attempt to bypass protections):

&destroy \$exp_object *noexplosion*

Reading a Text from the Text DB

&readtext \$Page_id \$t_id *<output_method>*

Ingame Help Command (depracted):

&help *<topic>*

Copying an existing Object (WiP; not working yet):

©object (unsure about Syntax)

Print all UI events that happen to the Console (except 'Frame_Advanced' and 'chatwindow_opened'):

&watchui *<stop>*

Print all Events that happen for a certain Object (not all implemented yet; only complete if the watched object is the event.object):

&watch_event_object \$exp_object*stop*

Enable/Disable additional internal Output:

&debugmode

Do certain Actions on Cues (can be used to force or re-start missions; be careful, can break stuff):

&cue *<action>* \$exp_cue *\$exp_param*

Unlock certain commands that are considered dangerous (description marked in red above):

&unlock

Color Explanation:

Actions that can be dangerous to continuing use of Savegames (will mark the Player Characters name with a red #)

Optional Parameter that can be omitted if not needed

Optional Parameter that can be omitted or repeated if wanted