ui_console

User Manual

Contents

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.

Enabling the Chatwindow / Console

The Chatwindow/Console is usually not bound to any Key, So you first have to assign one to it. You can find it under

Settings => Controls => General Controls => Toggle Chat Window (7th from the bottom)
(Note to self: check what „Focus Chat Window Input" does – 6th from bottom)

Usually the [^] Key is used for that purpose.

Just as side note there are some useful Commands by Egosoft you can Input here too:

/refreshmd	Reloads the MD Scripts
/refreshai	Reloads the AI Scripts
/reloadui	Reloads the UI / lua scripts
/exportlogbook <filename>	Save logbook as file in personal folder (must have .txt ending)
/thereshallbewings	Unlocks the advanced behavior tab (you only need to do this once on each Computer)
/aicompat check	Check AI script compatibility using in-game aicompat.xml
/aicompat pcheck	Check AI script compatibility using personal aicompat.xml
/aicompat check	Generate new aicompat.xml in personal folder

Less relevant Commands for Modding, but might still be relevant:

/chat|c <message>	Send chat message (no online feature using that yet)
/whisper|w <username <message>	Send direct chat message to user (no online feature using that yet)
/dumpholo	Dump holomap camera data
/dumpmouse	Dump mouse data
/dumpconv	Dump conversation system data

(Might not work/be available in X4; needs testing)
/lua <luascript>	This executes the given lua command in the Detail Monitor environment

Using the Commands of this Mod

Since the usual prefix Char for Commands (/) is already taken by Egosofts internal Commands and there is an error printed to the Log if it is used by this Mod I decidd to use a diffrent Letter for my Commands, namely it is [&] (Ampersand) as Prefix. You can change this in Letter in the t/0001.xml if you want. Just remember that you did when reffering to this explaination.

Command Overview

Currently the following commands are implemented: (currently working on a a detailed explaination)

Evaluating one or 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 arbitary 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):
&copyobject (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 Explaination:
Actions that can be dangerous to continuing use of Savegames (will mark the Player Characters name with a red #)
Optional Parameter that can be omited if not needed
Optional Parameter that can be omited or repeated if wanted

Expression Syntax Explaination

Expressions in this Console are used as Input for most Commands, and 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 pelicularities which will be pointed out.
Basically it is a Chain of Properties seperated by dots (.), starting with a Root property and with each step you get a certain property of the previous result.

Additionally some commands might take their own „Root Properties" to differ between diffrent types of input values. This has nothing to do with the property chain though, it is just to differ between possible input values. They might start with a property chain from the second step onwards if applicable, but often just take a single value here.

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 always use the first match, but additionally output an error to the Console if other matches are found.
0x1234 (hex id) 12345 (decimal id)	Its Object identifier, that is unique within a game session. Note that its not persistent over sessions. (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 pobject	player container (ship or station)
htargets	Holomap Targets (Not yet used by any command)
mapobjects	Holomap Visible Objects (Not yet used by any command)

Root Properties

The following Start Properties are available:

Object references	see above for explaination
$console_var	If you set a console Var before you can access it as root property
null	simply a null if you need it
true	actually numeric 1
false	numeric 0 in this case
component.<number> object.<number>	The Component/object referenced by the following numeric identifier. Use this if the autodetect fails, or if you want to start an expression chain with such an object (no autodetect here)
gameparam.<param>	Reading from the game params (Startup params?) - not sure yet what this is..
param.<groupname>.<paramname>	reading scriptparams data from libraries/parameters.xml
macro.<macroid>	getting a macro via its id as starting point
md.<scriptname>.<cuename>	getting an MD cue via its Script and Cuename. Can be used further for remote var reading
mdvar.$varname	getting an MD var via its name. Equal function to md.$varname in the game scripts.
global.$varname	getting a global var by its name
faction.<factionname>	getting a faction by its name
tag.<tagname>	getting a tag by its name. Will never fail since the tag is created if it doesnt exist yet.
ware.<warename>	getting a ware by its name
R.<number>	getting the result of a previous expression in the same command (as in Repeat X; must be supported by each command individually)

Following here are more root properties, but i consider them less useful for use in this Console, so i will save myself the explaination. Also they are largely untested and might not work.

position.<x>.<y>.<z>	vector.<x>.<y>.<z>	rotation.<yaw>.<pitch>.<roll>	readtext.<pageid>.tid>
stat.<statname>	userdata.<userdataentry>	userdatasigned.<userdataentry>	terraforming.<entries .....
lookup.<category>	assignment.<assignment>	controlpost.<controlpost>	entityrole.<entityrole>
entitytype.<entitytype>	notification.<type>	race.<race>	unlock.<type>
warebasket.<id>	waregroup.<id>	attention.<level>	relationchangereason.<reason>
killmethod.<method>	unitcategory.<category>	deployablecategory.<category>	command.<command>
commandaction.<action>	activity.<activity>	skilltype.<skill>	controlposition.<position>
formationshape.<shape>	level.<level>	alertlevel.<level>	roompopulationtype.<type>
purpose.<purpose>	controlpaneltype.<type>	signalleaktype.<type>	boardingbehaviour.<behaviour>
boardingphase.<phase>	blacklisttype.<type>	blacklistgroup.<group>	waretransport.<waretransport>
orderstate.<state>	moodlevel.<level>	weaponmode.<mode>	dronemode.<mode>
dockstate.<state>	shiptype.<type>	stationtype.<type>	roomtype.<type>
regiongroup.<group>	flightbehaviour.<behaviour>	flightcontrolmodel.<flightcontrolmodel>	objective.<objective>
missiontype.<type>	missiongroup.<group>	cuestate.<group>	quota.<quota>

Standard Properties

The following properties can be evaluated on each step. These here are entirely based on what is available in the scriptproperties.html but especially the Properties with Pseudo Data Types/using two steps at once are often not yet implemented. Please refer to said the scriptproperties.html to know what is available for a certain data type/at a certain point in the chain and also what each property is. The Order chosen in this rundown schould mostly match the order of an *unsorted* scriptproperties.html (because this groups related properties nicely)

exists	isoperational	iswreck	isconstruction	issurfaceelement	canbeattacked	extension
isextensionpersonal	class	realclass	state	macro	grouptag	parent
container	controllable	defensibleobject	ship	station	cluster	sector
zone	room	module	canbelivestreamed	isinternallystored	generationseed	seed
name	rawname	knownname	hasbeenrenamed	debugname	position	rotation
velocity	speed	relvelocity	relspeed	attention	size	length
height	width	owner	coverowner	trueowner	isplayerowned	isownerless
isknown isrelationchangedisabled	wares	revealedpercentage	scannedlevel	gravidarfactor	secrecylevel
isactive	isphysicsready	canbedismantled	isfunctiona	ishackabl	isindestructible	ishacked
ishullunrepairable	isrepairable	isinvulnerable	isshieldunrepairable	ismodular	hull	maxhull
hullpercentage	shield	maxshield	shieldpercentage	drop	canclaimownership	hasstorage
modulesets	makerraces	primarypurpose	ismilitary	threatscore	lastattacker	lastattacktime
isinliveview	currentradarrange	maxradarrange	formationleader	currentformationwingmen	idcode
isformationleader	isformationwingman	iscommandeerable	iscommandeered	isgodobject	isgodobjectentry	godentry
isgodshipentry	isgodstationentry	isgodproductionentry	isgamestartgodentry	isrespawnablegodobject
godentryname	godentrytags	ismasstraffic	ishighwaytraffic	istemptraffic	iszonechangepending
numwaypoints	region	ismission	missioncue	suspicious	isradarvisible
canbepickedup	canbetowed	tugship	throttle	value	buildresourcevalue	repairprice
numsignalleakslots	scuttleeffect	isonlineobject	isvisitor isvisitormissionobject	venture	venturename
masstrafficstartobject	masstrafficendobject	hasenginemod	haspaintmod	isunit	isdeployable	unitcategory
deployablecategory	formationshape	formationradius	isformationrolling	isformationrollmembers	formationmaxshipsperline
isinhazardousregion	isatnavigableposition	accesslicence	accessrestricted	canberecycled
dismantlingobject	salvageclaimants	isdatavault	islandmark	isfriendfoe	friendlist	foelist
target	isarmed	safetydistance	pilot	assignedpilot	aipilot	assignedaipilot
defencenpc	tradenpc	tradecomputer	engineer	shiptrader	shadyguy	roleentities
isnpcassignmentrestricted	order	nextorder	defaultorder	orders	buildorders
tradeorders	hasorderloop	commander	toplevelcommander	assignment	commanderentity
activesubordinategroupids	subordinates	subordinategroupdockoverride	subordinategroupid
subordinategroupprotectedsector	subordinategroupprotectedposition	subordinategroupreinforcefleet
subordinategroupresupplyatfleet	subordinategroupattackonsight	allsubordinates	allcommanders	controlroom
haswalkableroom	canhavedynamicinterior	hasscanner	longrange	maxscanlevel	combinedskill
iscapturable	boardingoperation	boardingoperations	boardingbehaviour	allmarinesdispatched	boardingresistance
baseboardingresistance	boardingstrength	constructionsequence	defencedronemode	loadoutlevel
rawloadoutlevel	minloadoutlevel	loadoutvariation	loadout	hasarmeddefencedrones	hasarmedminingdrones
hasarmedtransportdrones	miningdronemode	transportdronemode	isenemylockingon	ismissileincoming	hasweaponmod
hasshieldmod	productions	storagemodules	buildmodules	processingmodules	dockareas	piers
habitationmodules	welfaremodules	defencemodules	connectionmodules	buildmodule	buildprocessor	buildprocessors
freebuildprocessor	freebuildprocessors	dismantletarget	isdismantlingassignedobject	claimedsalvage	haswaveprotection
hasmasstraffic	buildingmodule	buildingprocessor	build	buildanchor	canbuildships	canequipships
cansupplyships	hasplannedconstruction	cargo	products	research	dockingenabled
originalproduct	resources	supplyresources	tradewares	buyprices	sellprices	buildbuyprices
buildsellprices	waretransport	productionmoney	wantedmoney	istraderestricted	hastradesubscription
haspermanenttradesubscription	supplies	supplyorders	money	hasownaccount	isaccountholder
minbudget	maxbudget	buildpricefactor	tradewaremoney	spawntime	type	typename
numfreeactorslots	actors	assigneddock	docksize	dock	dockslot hidecargochance
maxspeed	maxyawspeed	maxpitchspeed	maxrollspeed	speedlimit	canenterhighway	parkedat
pickedupobject	regiondamageprotection	job	jobname	isjobship	isrequestedjobship	jobexpired
jobcommander	jobmainsector	jobmainzone	encounterid	isencountership	constructionmodule	warebasket
isdeployedconstructionvessel	iscapitalship	islasertower	canhaveattackablemodules	hasshipmod
canbeclaimed	issupplyship	movementrefobject	environment	cantowobjects	towedobject
takesregiondamage	oxygenempty	oxygenlow	oxygenpercentage	oxygentimeremaining	buildstorage
canhavebuildstorage	istradestation	isshipyard	iswharf	isequipmentdock	isdefencestation
isrecyclingfacility	ispiratebase	isheadquarters	isfactionheadquarters	isplannedshipyard	splannedwharf
isplannedequipmentdock	isplanneddefencestation	isplannedrecyclingfacility	isscheduledforconstruction
isguided	destinations	istransitiondestination	istransitionsource	launcher	destination
exit	isaccelerator	entryhighway	exithighway	waitingship	highway	entry
sequence	islocked	ispartiallyunlocked	isunlocked	isadvanced	currentyield	maxyield
rarity	unbundle	isdroppedcontainer	blueprints	base	ispersistent	isprivate
rooms	isbusy	isbuilding	buildmacros	constructingcomponents	constructingmodule
requiresconstructionvessel	mayrequireconstructionvessel	iswaitingforconstructionvessel	iswaitingforresources
iswaitingforstorage	constructionvessel	constructionvesseldeployed	elapsedtime	elapsedsteptime	steptime

totaltime canabortbuild queuedbuild currentslotresources stepresources laststepresources
neededslotresources neededsequenceresources recycled isventuremodule venturedocks ventureships
numpierdocks isproducing ispaused ispausedmanually furnacestartpos requireditems iswalkable
dockarea walkablemodule dynamicinterior freemissionactorslot slotcontext assignedship pier
docked external isstorage isdockingallowed isunitdockingallowed isundockingallowed
isunitundockingallowed todockpos launchpos dockstate isbuildingallowed istradingallowed
isplayeronly isventureronly ventureplatform hasentrypos entrypos hasexitpos exitpos
economy security sunlight god jobs isexclusiveforextensionjobs factionlogic
locationtags alllocationtags yieldthresholds representatives isnormalcluster ispresentation systemid
planets averagedistancetolockbox averagemaxyield iscontested contestingfactions hasgravidarobscuringregion
hashazardousregion containsthewave waveprotectionresources policefaction typicallockboxmacro coreposition
coresize adjacentzones haspriority istempzone isnormalzone islocalhighway issuperhighway
isdisabled origin junctions entryjumpgate exitjumpgate closestpointtoplayer customhandler
controlpost role roleobject titleoverride iconoverride occupationname icon
race page isfemale isavatar iscommable ismissionactor
isremotecommable iscontrolentity isintransit istransferscheduled istemporary ishidden podarrivaltime
roomslot floortags iswalking slowwalkspeed walkspeed runspeed isspeaking
isinspeakrange lastspeaktime experienceprogress neededexperience controlled assignedcontrolled spacesuit
spacesuitmacro inventory stock shouldinformplayer shouldwarnplayer istradecomputer isshadyguy
isindependent tradesvisible facecutscene npctemplate targetslot currentchair hasclothingmod
hastool isinconversation isreadytofire mode isrepairing ismining iscombat
barrelposition component tags group offset staticoffset staticrotation
controlposition available buyer seller isbuyoffer isselloffer isoffer
isorder ware amount desiredamount offeramountt transferredamount minamoun
destroyedamount volume offervolume minvolume price unitprice minprice
relativeprice quantityfactor hasdynamicprice stocklevel bundle tradeoffer buyfree
sellfree iswareexchange isshiptoship iscrewexchange isbuyerpassive issellerpassive iscancelled
isshipbuild isrecycleshipbuild isexpansion isupgrade issoftwareonlyupgrade isrestock
isshipmodification isprocessing buildobject construction faction time consumables
boardee attackers approachingpods boardingphase boardingapproachthreshold boardinginsertionthreshold
description minrelation precursor issellable count list random
id script canplayercancel isinfinite isinloop isoverride isrunning
trade operation requiredskill clone last randominrange min
max average x y z yaw pitch roll inverse forward
right up shortname prefixname spacename homespacename primaryrace
knowntoplayer defaultpolicefaction coalition teamname customteamname isrelationlocked licences
heldlicences isaggressive iseconomic ispolice isprotective willclaimspace representative
headquarters doesresupply averageprice maxprice pricerange transporttag isinventory
isammo iscargo iscrafting isvolatile isgift israre isdropallowed
isresearchable illegal objectmacro objectcount ssoftware isweaponmod isenginemod
isshieldmod isshipmod isequipmentmod equipmentmodquality ispaintmod paintmodquality
isclothingmod clothingmodquality isprocessed table free capacity purposemacro
purposename docksizes productionvolume consumptionvolume cargocapacity allowedowners allowedwares
maxrelation enemy active version static staticbase namespace
library isinstance hasmissionoffer hasmission hasguidance offerlocations missiontype
missionendtime objective objectiveendtime actor isnumeric isstring femalename
pluralname tag hirable tiers skilltypes maxmk age
occupiedship platform entity galaxy computer activity autopilottarget
conversation conversationactor hasactivemission activemissiontype activemissionwaypoint hasacceptedonlinemission
isincontrolposition scanlevel debug isscreenshotmode screenshotmode ismapeditoractive
isinfullscreencutscene isinfullscreenmenu isvrversion isvrmode chardensity modulename
moduledescription speakpriority flightassist isdockuiactive hasemergencyeject isemergencyejectactive
shouldplayertaxiwaitforplayer shouldplayershipswaitforplayer isonline canactivateliveview
recommendedmodule isgameover index element key null
integer largeint float angle hitpoints largefloat cue
keyword xmlkeyword componentstate killmethod relationchangereason flightbehaviour
flightcontrolmodel entitytype entityrole skilltype missiongroup level alertlevel
command commandaction roompopulationtype purpose controlpaneltype signalleaktype blacklisttype
blacklistgroup moodlevel orderstate datatype cuestate notification string
componentslot licence vector quadrant unlock weaponmode dronemode
shiptype stationtype roomtype regiongroup none down left
front back param param2 param3

Special Properties

These Properties are not part of the original Property Chain Syntax, but they emulate other functions of the Scripting Language that might be useful. Namely Type conversions and such stuff.

$variable |
=> if the current result is a a Cue, Entity or Table wit will read a Variable off them (Cue Variable, Entity Blackboard Var, Table Value from a Key) -
     baiscally the same syntax as with global.$variable or mdvar.$variable

<number> |
=> If the current result is a Group, List or Table it will result in the numeric nth Entry of it (1-based). In case of Tables it obviously needs to have the
     respective numeric key

     (Note: For tables other Keys than Numeric or String are not supported properly yet; planned is a
     workaround-y way of using Console Vars, but details have to be ironed out yet)

rl rt rg rb rc   Internal Call: $currentStep.{$nextStep}
=> as in 'readlist','readtable','readgroup','readblackboard' or 'readcue' enforces the previous behavior in case it doesnt work automatically,
     with the next Value in the Chain being the Var/Key Name

comp   Internal call: $CurrentStep == $NextStep
=> compare hack (temporary); compares the current result of the Chain with the next single Step
     (useful for eg chekcing against an object reference); makes the result boolean.

cv:i   => convert to integer   Internal call: ($CurrentStep)i

cv:f   => convert to float   Internal call: ($CurrentStep)f

cv:component   Internal call: component.{$CurrentStep}
=> convert to component – turns a number into an object if possible. This should usually happen automatically by each command if an Object
     expected, but you can enforce it with this

typeof   => results in the datatype of the current Property   Internal Call: typeof $CurrentStep

distanceto   Internal Call: $CurrentStep.distanceto.{$NextStep}
=> results in the Distance between the current Object and the following Object. Next Object must be a single location Step.
     (like a direct Object reference)

Using scriptproperties.html locally (and some additions)

Work in Progress