

# ut\_console

## User Manual

### Contents

Introduction.....	2
Enabling the Chatwindow / Console.....	2
=> Egosofts official Chatwindow Commands.....	2
=> Using the Commands of this Mod.....	2
Command Overview.....	3
-->&eval.....	3
-->&setval.....	3
-->\$warp.....	3
-->&checkship.....	3
Expression Syntax Explanation.....	5
=> Object References.....	5
=> Root Properties.....	6
=> Standard Properties.....	7
=> Special Properties.....	9
Using scriptproperties.html locally (and some additions).....	10
=> Useful additional Tricks to improve the reference files.....	11
-->Show a full list of Vanilla Macros and also print their respective File Path as Description.....	11
-->Show Script Param Values in Scriptproperties.....	11
-->Show Masstraffic Jobs in Jobeditor.....	13
-->Add a Column for Loadout Levels in the Jobeditor.....	13

# 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.

## Egosofts official Chatwindow Commands

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 deciddd to use a different 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 referring to this explanation.

# Command Overview

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

Parameters:	An explanation of the Parameters if they are specific
Usage:	general usage showing all possible inputs with placeholders
Example call:	a concrete example how this Command is intended to be used. Depending on the command there might be more than one.
Output:	The Output the Example Call generated for me. There will be slight differences for you.
Effect:	Description of the Effect the example command has, if any.

## &eval

Evaluating one or more Expressions

Usage:	<code>&amp;eval \$exp_1 \$exp_2 ...</code>
Example call:	<code>&amp;eval pobject.cluster.debugname pobject.pilot.debugname</code>
Output:	<code>&amp;eval pobject.cluster.debugname pobject.pilot.debugname</code> pobject.cluster.debugname => Halikvah's Choice (AYX-867) [0x32dd6] pobject.pilot.debugname => Yahanis Hebalisos Tumulis X (VPR-008) [0x36542]

## &setval

Setting a Value at the end of a Property Chain/Expression. Please refer to the Properties Section for a List of Set-able properties.  
(needs to be unlocked)

Usage	<code>&amp;setval \$exp_set \$exp_get1 \$exp_get2</code>
Example call:	<code>&amp;setval pobject.minhull 90</code>
Output:	<code>&amp;setval pobject.minhull 90</code> set property 0x1551f.minhull to 90 (secondary null) – cannot confirm since its not property-accessible
Effect:	Hull of the current Player Object will never drop below 90% (its indestructible). Note that you need to do this sepearately for each surface Element too if you want.

## \$warp

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

Usage:	<code>&amp;warp \$exp_object \$exp_destination</code>
Example Call:	<code>&amp;warp pobject gtarget</code>
Output:	<code>&amp;warp pobject gtarget</code>
Effect:	Instantly Teleports the Player Object close to the current guidance Target

## &checkship

Spawning two Ships, one of which gets instantly wrecked, for and generating a Report about it for Ship Modding Purposes. The Ship will change its Paintjob every few seconds from a pre-selected Set, and you can add as many additional ones to the cycle as you want.

Usage:	<code>&amp;checkship \$macro_id_or_exp crewskill.&lt;number&gt; paintcycletime.&lt;number&gt; ware.paintjob_###1 ware.paintjob_###2 ....</code>								
Parameters:	<table><tr><td><code>\$macro_id_or_exp</code></td><td>a ship macro name or expression that results in a ship macro for spawning/testing</td></tr><tr><td><code>crewskill.&lt;number&gt;</code></td><td>a Number from 0 to 15 to define the Crews skill for the Spawn. Will be random if omitted.</td></tr><tr><td><code>Paintcycletime.&lt;number&gt;</code></td><td>time for which each paintjob stays in seconds. Default is 3. 0 is changing every Frame.</td></tr><tr><td><code>Ware.paintjob_####</code></td><td>Adding a Paintjob to the cycle. Can be repeated as often as needed.</td></tr></table>	<code>\$macro_id_or_exp</code>	a ship macro name or expression that results in a ship macro for spawning/testing	<code>crewskill.&lt;number&gt;</code>	a Number from 0 to 15 to define the Crews skill for the Spawn. Will be random if omitted.	<code>Paintcycletime.&lt;number&gt;</code>	time for which each paintjob stays in seconds. Default is 3. 0 is changing every Frame.	<code>Ware.paintjob_####</code>	Adding a Paintjob to the cycle. Can be repeated as often as needed.
<code>\$macro_id_or_exp</code>	a ship macro name or expression that results in a ship macro for spawning/testing								
<code>crewskill.&lt;number&gt;</code>	a Number from 0 to 15 to define the Crews skill for the Spawn. Will be random if omitted.								
<code>Paintcycletime.&lt;number&gt;</code>	time for which each paintjob stays in seconds. Default is 3. 0 is changing every Frame.								
<code>Ware.paintjob_####</code>	Adding a Paintjob to the cycle. Can be repeated as often as needed.								
Example Call:	<code>&amp;checkship ship_par_1_destroyer_02_a_macro crewskill.15 paintcycletime.1</code>								
Output:	<code>&amp;checkship ship_par_1_destroyer_02_a_macro crewskill.15 paintcycletime.1</code> converting ship_par_1_destroyer_02_a_macro from string to macro: ship_par_1_destroyer_02_a_macro Output File is: logs/ut_console/ ship_par_1_destroyer_02_a_macro report.txt								
Effect:	Spawns tow Odysseus E in front of teh Player, one of which immediately explodes to leave a Wreck. The other one has a full 5 Star Crew and changes its Paintjob every second in the default cycle.								

Additionally a Report about this Ship Type is put in the X4 Documents folder under  
logs/ut\_console/ ship\_par\_1\_destroyer\_02\_a\_macro report.txt

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):  
&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

**unlock - All Console Commands unlocked! This includes the Commands that can break our game!  
If you use them your Save will be marked accordingly. There will be no further Warnings!**

Color Explanation:

Commands and their Calls:

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

Output to the Console:

Repeat of the Command (safe Commands)

Repeat of the Command (non-safe Commands)

Standard Output

Debug Output (usually hidden)

Help/Info Output

# Expression Syntax Explanation

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 peculiarities which will be pointed out. For example ut\_console does not use { } (curly brackets) at all, but some properties require it internally.

Basically it is a Chain of Properties separated 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 different 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: These references are only valid in my Console and do not apply to the xml scripting languages.

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)	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 Start Properties are available: (most of them are taken directly from the xml scripting languages unless indicated otherwise)

Object references	see above for explanation (ut_console only)
\$console_var	If you set a console Var before you can access it as root property (ut_console only)
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. (ut_console omits the {} )
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 explanation. 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>	userdataassigned.<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 should 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	group	tag
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	currenttradarrange	maxradarrange	formationleader	currentinformationwingmen		idcode
isformationleader	isformationwingman	iscommandeerable	iscommandeered	isgodobject	isgodobjectentry	godentry
isgodshipentry	isgodstationentry	isgodproductionentry	isgamestartgodentry		isrespawnablegodobject	
godentryname	godentrytags	ismassstraffic	ishighwaytraffic	istemptraff	iszonechangedepending	
numwaypoints	region	ismission	missioncue	suspicious	isradarvisible	
canbepickedup	canbetowed	tugship	throttle	value	buildresourcevalue	repairprice
numsignalleakslots	scuttleeffect	isonlineobject	isvisitor	isvisitormissionobject	venture	venturename
massstrafficstartobject	massstrafficendobject	hasenginemod	haspaintmod	isunit	isdeployable	unitcategory
deployablecategory	formationshape	formationradius	isformationrolling	isformationrollmembers		formationmaxshipsperline
isinhazardousregion	isatnavigableposition		accesslicence	accessrestricted	canberecycled	
dismantlingobject	salvagedclaims	isdatavault	islandmark	isfriendfoe	friendlist	foelist
target	isarmed	safetydistance	pilot	assignedpilot	apilot	assignedapilot
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		
subordinategroupresupplya fleet		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	
hasmassstraffic	buildingmodule	buildingprocessor	build	buildanchor	canbuildships	canequipships
cansupplyships	hasplannedconstruction		cargo	products	research	dockingenabled
originalproduct	resources	supplyresources	tradewares	buyprices	sellprices	buildbuyprices
buildsellprices	waretransport	productionmoney	wantedmoney	istraderrestricted		hasradesubscription
haspermanenttradesubscription		supplies	supplyorders	money	hasownaccount	isaccountholder
minbudget	maxbudget	buildpricefactor	tradewaremoney	spawnntime	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	islaser tower	environment	cantowobjects	towedobject	hasshipmod
canbeclaimed	issupplyship	movementrefobject				
takesregiondamage	oxygenempty	oxygenlow	oxygenpercentage	oxygen timer remaining		buildstorage
canhavebuildstorage		istradestation	isshipyard	iswharf	isequipmentdock	isdefencestation
isrecyclingfacility	ispiratebase	ishedquarters	isfactionheadquarters		isplannedshipyard	splannedwharf
isplannedequipmentdock		isplanneddefencestation	isplannedrecyclingfacility		isscheduledfordeconstruction	
isguided	destinations	istransitiondestination		istransitions source	launcher	destination
exit	isaccelerator	entryhighway	exithighway	waitingship	highway	entry
sequence	islocked	ispartiallyunlocked	isunlocked	isadvanced	currentlyield	maxyield
rarity	unbundle	isdroppedcontainer	blueprints	base	ispersistent	isprivate
rooms	isbusy	isbuilding	buildmacros	constructingcomponents		constructingmodule
requiresconstructionvessel		mayrequireconstructionvessel		iswaitingforconstructionvessel	iswaitingforresources	
iswaitingforstorage	constructionvessel	constructionvesseldeployed		elapsedtime	elapsedstep time	step time

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	isventureonly	ventureplatform	hasentrypos	entrypos	hasexitpos	exitpos
economy	security	sunlight	god	jobs	isexclusiveforextensionjobs	factionlogic
locationtags	allocationtags	yieldthresholds	representatives	isnormalcluster	ispresentation	systemid
planets	averagedistancetolockbox	averagemaxyield	iscontested	contestingfactions	hasgravidarobscuringregion	
hasazardousregion	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	
isremotecomable	iscontrolentity	isintransit	istransferscheduled	istemporary	ishidden	podarrivaltime
roomslot	floortags	iswalking	slowwalkspeed	walkspeed	runspeed	isspeaking
isinspeakrange	lastspeaktime	experienceprogress	neededexperience	controlled	assignedcontrolled	spacesuit
spacesuilmacro	inventory	stock	shouldinformplayer	shouldwarnplayer	istradedcomputer	isshadyguy
isindependent	tradesvisible	facecutscene	npctemplate	targetslot	currentchair	hasclothingmod
hasfool	isinconversation	isreadytofire	mode	isrepairing	ismining	iscombat
barrelposition	component	tags	group	offset	staticoffset	staticrotation
controlposition	available	buyer	seller	isbuyoffer	isselloffer	isoffer
isorder	ware	amount	desiredamount	offeramounttt	transferredamount	minamoun
destroyedamount	volume	offervolume	minvolume	price	unitprice	minprice
relativeprice	quantityfactor	hasdynamicprice	stocklevel	bundle	tradeoffer	buyfree
selffree	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	free	ispaintmod	paintmodquality
isclothingmod	clothingmodquality	isprocessed	table	free	capacity	purposemacro
purposeaname	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	hasactiveemission	activemissiontype	activemissionwaypoint	hasacceptedonlinemission	
isincontrolposition	scanlevel	debug	isscreenshotmode	screenshotmode	ismapeditoractive	
isinfullscreenscene		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	controlpanelttype	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.

<code>\$variable</code>	Internal Call: <code>\$currentStep.{ \$nextStep }</code>
=> 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) - baisscally the same syntax as with <code>global.\$variable</code> or <code>mdvar.\$variable</code>	
<code>&lt;number&gt;</code>	Internal Call: <code>\$currentStep.{ \$nextStep }</code>
=> 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)	
<code>rl rt rg rb rc</code>	Internal Call: <code>\$currentStep.{ \$nextStep }</code>
=> 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	
<code>comp</code>	Internal call: <code>\$CurrentStep == \$NextStep</code>
=> 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.	
<code>cv:i</code>	Internal call: <code>( \$CurrentStep )i</code>
<code>cv:f</code>	Internal call: <code>( \$CurrentStep )f</code>
<code>cv:component</code>	Internal call: <code>component.( \$CurrentStep )</code>
=> 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	
<code>typeof</code>	Internal Call: <code>typeof \$CurrentStep</code>
<code>distanceto</code>	Internal Call: <code>\$CurrentStep.distanceto.( \$NextStep )</code>
=> 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)	

# The following Stuff are just additional Tips and have nothing to do with my mod itself:

## Using scriptproperties.html locally (and some additions)

How to get an Apache Server set up to view scriptproperties and jobeditor (sidestepping CORS):  
(i know it is overkill for this, but im more familiar with it than other methods)

- Download and install XAMPP from <https://www.apachefriends.org/de/index.html>  
(NOTE: Install to C:/XAMPP or similar, since the Program needs to write to its own Folder, which windows restricts in program files)  
Minimum Installation is sufficient.
- After the Setup ran you should find yourself in the XAMPP control panel.  
In the Line with Apache click on [Config] -> Browse Apache  
Then go in the folder config -> extra and open httpd-vhosts.conf for edit
- Add this Section at the end of the File and point it to your extracted Game Files:

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs"
    ServerName localhost
    Alias /x4mod "C:/X4_extracted"
    <Directory "C:/X4_extracted">
        Options Indexes FollowSymLinks Includes ExecCGI
        AllowOverride All
        Order allow,deny
        Allow from all
        Require all granted
    </Directory>
</VirtualHost>
```

(Edit both of the C:/X4\_Extracted to point to your extracted game files)  
Save and close the File.

- back in the XAMPP Control Center click the box left of Apache to set it up as Service (so it starts with Windows), and then click [Start] to the right of it to start the Webserver and open localhost/x4mod
- in a Browser to confirm it is working. You can open the jobeditor.html and scriptproperties.html from here.

Also you can reference the xsd files without having them to move around, just add this to your ai/md files in the root node:

```
<mdscript name= ..... xsi:noNamespaceSchemaLocation="http://localhost/x4mod/libraries/md.xsd" >
<aiscript name= ..... xsi:noNamespaceSchemaLocation="http://localhost/x4mod/libraries/aiscripts.xsd" >
```

## Useful additional Tricks to improve the reference files

Show a full list of Vanilla Macros and also print their respective File Path as Description:

- libraries/scriptproperties.xml - replace the `<keyword name="macro">` section:

```
<keyword name="macro" description="Macro lookup">
  <property name="&lt;:macroname&gt;:" result="Macro" type="macro" />
  <import source=" ../index/macros.xml" select="/index/entry">
    <property name="@name" result="@value" type="macro" />
  </import>
  <import source=" ../extensions/ego_dlc_split/index/macros.xml" select="/index/entry">
    <property name="@name" result="@value" type="macro" />
  </import>
  <import source=" ../extensions/ego_dlc_terran/index/macros.xml" select="/index/entry">
    <property name="@name" result="@value" type="macro" />
  </import>
  <import source=" ../extensions/ego_dlc_pirate/index/macros.xml" select="/index/entry">
    <property name="@name" result="@value" type="macro" />
  </import>
  <import source=" ../extensions/ego_dlc_boron/index/macros.xml" select="/index/entry">
    <property name="@name" result="@value" type="macro" />
  </import>
  <import source=" ../extensions/ego_dlc_timelines/index/macros.xml" select="/index/entry">
    <property name="@name" result="@value" type="macro" />
  </import>
</keyword>
```

Show Script Param Values in Scriptproperties:

three changes needed:

- libraries/scriptproperties.xml- replace the `<keyword name="param">` section:

```
<keyword name="param" description="Access to script parameters in parameters.xml">
  <property name="&lt;:categoryname&gt;.&lt;:paramname&gt;:" result="Script parameter" />
  <import source="parameters.xml" select="/parameters/scriptparams/category/param">
    <property name=" ../@name" subname="@name" result="@value" type="string" />
  </import>
</keyword>
```

- libraries/scriptproperties.xsl - replace the `<xsl:template match="property">` section:

```
<xsl:template match="property">
  <tr>
    <td width="20%" class="property">
      <span class="propertyname">
        <xsl:value-of select="@name" />
        <xsl:if test="@subname">
          <xsl:text>.</xsl:text>
          <xsl:value-of select="@subname" />
        </xsl:if>
      </span>
    </td>
    <td width="10%" class="property">
      <xsl:if test="@type">
        <xsl:apply-templates select="@type" mode="datatypeeref" />
      </xsl:if>
    </td>
    <td class="property">
      <xsl:if test="@result">
        <span class="comment">
          <xsl:value-of select="@result" />
        </span>
      </xsl:if>
    </td>
  </tr>
</xsl:template>
```

- scriptproperties.html- replace the *function importExternalProperties()* function:

```
function importExternalProperties() {
    var loadedfiles = new Object(); // Store loaded files here, so we don't import the same file twice
    var textdbdef = evaluateXPath(xmlDoc.documentElement, "/*/textdb");
    if (textdbdef && textdbdef.length) {
        var textdbfilename = textdbdef[0].getAttribute("source");
        if (textdbfilename) {
            textdbDoc = loadXMLFile(textdbfilename);
        }
    }
    var foundnodes = evaluateXPath(xmlDoc.documentElement, "/*/keyword/import/property");
    for (var i = 0; i < foundnodes.length; ++i) {
        var propertynode = foundnodes[i];
        var importnode = propertynode.parentNode;
        var keywordnode = importnode.parentNode;
        var sourcefile = importnode.getAttribute("source");
        var selectexp = importnode.getAttribute("select");
        var nameexp = propertynode.getAttribute("name");
        var subnameexp = propertynode.getAttribute("subname"); // Add this line to get the subname expression
        var resultexp = propertynode.getAttribute("result");
        var type = propertynode.getAttribute("type");
        var ignoreprefix = propertynode.getAttribute("ignoreprefix");
        if (sourcefile && selectexp && nameexp) {
            var importedxml = loadedfiles[sourcefile];
            if (!importedxml) {
                importedxml = loadXMLFile(sourcefile);
                loadedfiles[sourcefile] = importedxml;
            }
            if (importedxml) {
                var importedtypes = evaluateXPath(importedxml.documentElement, selectexp);
                for (var j = 0; j < importedtypes.length; ++j) {
                    var curtype = importedtypes[j];
                    var namenodes = evaluateXPath(curtype, nameexp);
                    var subnamenodes = evaluateXPath(curtype, subnameexp); // Add this line to get the subname nodes
                    if (namenodes && namenodes.length > 0 && (namenodes[0].data || namenodes[0].nodeValue)) {
                        var newproperty = xmlDoc.createElement("property");
                        // Set property name
                        var name = String(namenodes[0].data || namenodes[0].nodeValue);
                        if (ignoreprefix && (ignoreprefix == "1" || ignoreprefix == "true") && name.lastIndexOf(".") > 0) {
                            name = name.substr(name.lastIndexOf(".") + 1);
                        }
                        newproperty.setAttribute("name", name);
                        // Set property subname
                        if (subnamenodes && subnamenodes.length > 0 && (subnamenodes[0].data || subnamenodes[0].nodeValue)) {
                            var subname = String(subnamenodes[0].data || subnamenodes[0].nodeValue);
                            newproperty.setAttribute("subname", subname);
                        }
                        // Set property result (either from text node or attribute value)
                        if (resultexp) {
                            var resultnodes = evaluateXPath(curtype, resultexp);
                            if (resultnodes && resultnodes.length > 0) {
                                newproperty.setAttribute("result", convertFormatString(resultnodes[0].data || resultnodes[0].nodeValue));
                            }
                        }
                        // Set property type
                        if (type) {
                            newproperty.setAttribute("type", type);
                        }
                        keywordnode.appendChild(newproperty);
                    }
                }
            }
        }
    }
}
```

## Show Masstraffic Jobs in Jobeditor

- libraries/editor.xml :  
Replace this:  

```
<table id="jobs" name="Jobs">  
  <file name="jobs.xml" select="/jobs/job[ship and not(masstraffic)]" />
```

change to:

```
<table id="jobs" name="Jobs">  
  <file name="jobs.xml" select="/jobs/job" />
```

## Add a Column for Loadout Levels in the Jobeditor

- libraries/editor.xml - Add this at the end of the `<column group name="Ship" base="ship">` section:

```
<column value="loadout/level/@min" name="Loadout Min" type="float" />  
<column value="loadout/level/@max" name="Loadout Max" type="float" />
```