




# Lua function overview


 This list can contain details about planned functions in yet unreleased versions.

Please understand that the mentioning of such a version in this list does not imply that the version will become available in the near future (it's not even a confirmation that the version will be released at all). It's merely to be understood as a heads-up information of what a following version might provide and ensures that if that version is released, the documentation is already up to date at the time of the release.


Also note that any information provided here for not yet released versions can change at any time before the version is released without any further notice (that includes even the removal of a function which was mentioned here before the release of a particular version).


 **UNSUPPORTED** markers imply that the function is technically available but not part of our provided public Lua API set. We might change these functions to be in the supported set of Lua functions in the future. Until then, the function can change its behavior any time and/or even be entirely removed (without a replacement). Changes to these functions are usually not listed in the UI changelog and are also (usually) not covered in the overview of breaking changes between X Rebirth releases.

Version	Function	Note
deprecated in 4.21 Beta 1 (VR: 4.20)	<b>DEPRECATED AbilityMenuClosed()</b> <i>Notifies the system that the ability menu was closed.</i>	 <b>DEPRECATED</b> - use FFI function CircularMenuClosed() instead
added in 4.00 RC 2	<b>AbilityMenuOpened()</b> <i>Function which needs to be called, if the ability menu is opened.</i>	
	<b>AbortMission</b> (missionID) <i>Aborts the specified mission.</i>	
	<b>ActivateDroneAbility</b> (droneSlotNumber) <i>Activates the drone in the given drone slot number (valid range: 1-8).</i>	
	<b>ActivatePassiveDroneAbilities()</b> <i>Activates all drones with passive abilities.</i>	
changed in 4.00 Beta 4	addedamount = <b>AddAmmo</b> (defensible, ammomacroname, amount, [checkonly]) <i>Tries to add the specified amount of the ammo to the defensible. Returns actually added amount.</i>	Added "checkonly" parameter in 4.00 Beta 4
	<b>AddCargo</b> (containerID, ware, amount) <i>Adds the specified amount of the ware to the object.</i>	
changed in 4.00 Beta 4	<b>AddInventory</b> (entityID, ware, amount[, crafting]) <i>Adds the specified amount of the ware to the inventory of the entity. Optionally flag this as the result of crafting.</i>	Added "crafting" parameter in 4.00 Beta 4
	<b>AddKnownItem</b> (library, item) <i>Adds the item to the specified library, if not yet added.</i>	
	<b>AddLicence</b> (faction, licencetype, licencefaction) <i>Adds the licence of the specified type of the licencefaction to the faction</i>	
	<b>AddLogbookEntry</b> (category, text [, interaction [, interactioncomponent]]) <i>Adds text as a logbook entry in the specified category.</i>	
	transferredamount = <b>AddMoney</b> (entityID, amount) <i>Adds or removes the specified amount to the entity account</i>	
	<b>AddPlayerDrone</b> (macro, amount) <i>Adds specified macro as drones to the playership.</i>	
	tradeID = <b>AddTradeToShipQueue</b> (tradeID, containerID, amount) <i>Adds the specified trade to the queue of the container. Returns ID of the trade on the queue.</i>	
changed in 4.00 RC 2	<b>AddUITriggeredEvent</b> (screen, control[, screenparam]) <i>Triggers MD/AI event condition &lt;event_ui_triggered /&gt;.</i>	optional screenparam parameter added in 4.00 RC 2

	<b>AddUnits</b> (defensibleID, unitMacro, amount, unavailable) <i>Adds or removes the specified amount of units to/from the defensible. Units can be flagged unavailable.</i>	
	string = <b>AdjustMultilineString</b> (string) <i>Helper function to fix unsupported line breaks.</i>	
	<b>AIRangeUpdated</b> (controllableID, entityID) <i>Triggers the object signal "range_setting_updated" on all subordinates of the controllable.</i>	
	AreAttackersInWarningRange	
	AreObserversInWarningRange	
	AreSliderValuesFixed	
	<b>AttackEnemySettingChanged</b> (defencenpcID) <i>Triggers the object signal "update config" on defence npc</i>	
	hullfraction = <b>CalculateBuildSlotHullFraction</b> (componentID, sequence, stage) <i>Calculates the fraction of hull for an individual build slot of the component.</i>	
	hullfraction = <b>CalculateTotalHullFraction</b> (componentID) <i>Calculates the fraction of hull of the component.</i>	
	bool = <b>CanBeSubordinateOf</b> (controllableID, othercontrollableID) <i>Checks whether the controllable can be subordinate of the othercontrollable.</i>	
	success = <b>CancelConversation</b> () <i>Cancels the current conversation, e.g. when closing a conversation menu.</i>	
	bool, error = <b>CanTradeWith</b> (tradeID, containerID, amount) <i>Checks if trade is possible for the specified container.</i>	It may be necessary to call <b>SetVirtualCargoMode</b> on containerID first, if the container has a shopping queue.
	bool = <b>CanViewLiveData</b> (objectID or zoneID) <i>Checks whether live data is available for the specified object or zone.</i>	
	bool = <b>CheckInputProfileRegression</b> () <i>Checks whether inputmap.xml version regressed in the past.</i>	
	bool = <b>CheckSuitableTransportType</b> (containerID, ware) <i>Checks whether the specified container can transport the ware.</i>	
added in 4.00 Beta 3	<b>ClearContainerStockLimitOverride</b> (containerID, ware) <i>Remove any defined stock limit for a particular ware at a ship/station.</i>	
	<b>ClearContainerWarePriceOverride</b> (destructibleID [, ware] [, buysellswitch]) <i>Removes custom prices from the container of the destructible. Optional only for the specified ware and only buy/sell prices.</i>	
deprecated in 4.31 Beta 1	<b>DEPRECATED ClearErrors</b> () <i>Clears the internal error log.</i>	 <b>DEPRECATED</b> - use FFI function ClearErrorLog() instead
	<b>ClearLogbook</b> ([keepnumentries [, category]]) <i>Removes all entries from the logbook. Optional a number of entries can be preserved and only affect a specified category.</i>	
	<b>ClearTradeQueue</b> (containerID) <i>Clear the trade queue of the specified container.</i>	Clearing the trade queue will not happen instantaneously, especially if a trade is of the queue is already in progress.
	CloseMenusUponMouseClicked	
	bool = <b>CompareJumpRoute</b> (numgates, numjumps, othernumgates, othernumjumps) <i>Compare the length of one jump route with another. Both are specified with the amount of gate transitions and jumps they include.</i>	

	64-bit int = <b>ConvertIDTo64Bit</b> (luaID) Converts a luaID to a 64-bit integer, e.g. to interface with an FFI function.	
	string = <b>ConvertIntegerString</b> (number [, separators [, accuracy [, notrailingspaces [, colorprefix]]]]) <i>Formats number into a string with a given accuracy, thousand separators, trailing spaces to offset SI prefixes and colored SI prefixes.</i>	
	string = <b>ConvertMissionLevelString</b> (difficulty) <i>Converts the difficulty of a mission into a string.</i>	
added in 4.00 Beta 1	string = <b>ConvertMissionTypeString</b> (missiontype) <i>Converts the type of a mission into a string.</i>	
	string = <b>ConvertMoneyString</b> (money [, showcents [, separators [, accuracy [, notrailingspaces [, colorprefix]]]]) <i>Formats a money value into a string with a given accuracy, thousand separators, credit cents, trailing spaces to offset SI prefixes and colored SI prefixes.</i>	
	64-bit int = <b>ConvertStringTo64Bit</b> (64BitStringID) Converts a string representing a 64-bit value to a 64-bit integer.	
	luaID = <b>ConvertStringToLuaID</b> (64BitStringID) Converts a string representing a 64-bit value to a luaID, e.g. to pass it to MD/AI as a component.	
changed in 4.00 Beta 1	string = <b>ConvertTimeString</b> (time [, format [, separators [, precision]]]) <i>Formats a time value in seconds into a string in the provided format.</i>  <i>Possible specifiers in the format string:</i> <ul style="list-style-type: none"> <li>• %s: All seconds</li> <li>• %.#s: All seconds with precision override (# = 1...9), e.g. %.3s</li> <li>• %S: Seconds (00-59)</li> <li>• %.#S: Seconds (00-59) with precision override (# = 1...9), e.g. %.3S</li> <li>• %m: All minutes</li> <li>• %M: Minutes (00-59)</li> <li>• %h: All hours</li> <li>• %H: Hours (00-23)</li> <li>• %d: All days</li> <li>• %%: A % sign</li> <li>• %T: Use time format from TextDB, showing days only if time &gt; 1d</li> <li>• %.#T: Like %T but with precision override (# = 1...9), e.g. %.3T</li> </ul> <i>Default format is "%T" if no format string is provided.</i>  <i>Separators: Use thousand separators, only applies to %s, %m, %h, %d</i>  <i>Precision: Number of fractional digits. Only applies to %s, %S, and %T. Default is 0. Use -1 for automatic display. A value of -1 is not compatible with separators. (Can be overridden by .# notation in format string)</i>	precision support for %s, %S, and %T were added in 4.00 Beta 1
	CreateButton	
added in 4.00 Beta 2	descriptor = <b>CreateCheckBox</b> (checkboxDescription)	
	CreateCutsceneDescriptor	
	CreateEditBox	
	CreateFontString	

added in 4.00 Beta 1	descriptor = <b>CreateGraph</b> (graphDescription)	
	CreateIcon	
deprecated in 4.31 Beta 1	<b>DEPRECATED</b> descriptor = <b>CreateInteractionDescriptor</b> (param1, param2) <i>Creates an interaction descriptor with the specified parameters.</i>	 <b>DEPRECATED</b> - use <b>CreateInteractionDescriptor2()</b> instead
added in 4.31 Beta 1	interactionDescriptorID = <b>CreateInteractionDescriptor2</b> (param1, param2) <i>Creates an interaction descriptor with the specified parameters and returns its ID.</i>	
	clusterID, objectID = <b>CreateObjectInPresentationCluster</b> (objectmacro [, backgroundmacro]) <i>Creates the specified object in a new cluster with the optional background.</i>	The cluster needs to be destroyed using <b>DestroyPresentationCluster</b> , when it is no longer needed.
	CreateProgressElement	
	CreateRenderTarget	
	CreateSink	
	CreateSlider	
	CreateSliderTableView	
	CreateSliderView	
	CreateSource	
changed in 4.21 Beta 1 (VR: 4.20)	CreateTable	optional argument alerteffectcolor added in 4.21 Beta 1 (VR: 4.20)
	CreateTableView	
	CreateThreeTableRenderTargetView	
	CreateThreeTableView	
	CreateTimer	
	CreateTwoTableRenderTargetView	
	CreateTwoTableSliderRenderTargetView	
	CreateTwoTableSliderView	
	CreateTwoTableView	
	DeactivateView	
	DebugError	
	<b>DebugProfileStart()</b> <i>Resets the profile timer. Call this function at the start of the section which you want to profile.</i>	
	timeInMs = <b>DebugProfileStop()</b> <i>Returns the time (in ms) since the last call to DebugProfileStart().</i>	
	<b>DestroyPresentationCluster</b> (clusterID) <i>Destroys a previously created cluster and everything inside it.</i>	
	DialogMenuHidden	
	DialogMenuShown	
	DisableMissionObjectiveBar	
	DisplayView	
	DumpAllMessages	

	DumpAllMessageSinks	
	DumpAllMessageSources	
	EnableMissionObjectiveBar	
deprecated in 4.31 Beta 1	<b>DEPRECATED</b> EventmonitorInteractionHidden	 <b>UNSUPPORTED.</b> Not designed to be used by mods.  <b>DEPRECATED</b> - use EventmonitorInteractionHidden2() instead
added in 4.31 Beta 1	EventmonitorInteractionHidden2	
deprecated in 4.31 Beta 1	<b>DEPRECATED</b> EventmonitorInteractionShown	 <b>UNSUPPORTED.</b> Not designed to be used by mods.  <b>DEPRECATED</b> - use EventmonitorInteractionShown2() instead
added in 4.31 Beta 1	EventmonitorInteractionShown2	
	ExecuteDebugCommand	
	text or nil = <b>ExistsText</b> (page, line) <i>Returns text if it exists, nil otherwise.</i>	
	numgates, numjumps = <b>FindJumpRoute</b> (fromsectorID, tosectorID) <i>Calculates the distance between two sectors in number of gate transitions and jumps needed.</i>	
	ForceAnarkUpdate	
	... = <b>GetAccountData</b> (entityID, ...) <i>Returns properties of the account of the specified entity. Possible properties are: "money", "minmoney", "maxmoney", "isdummy".</i>	
	GetActionDetails	
	GetActionInfo	
	componentID or nil = <b>GetActiveGuidanceMissionComponent</b> () <i>Returns the component with the active mission guidance if any.</i>	
	GetActiveMission	
	string = <b>GetAdapterOption</b> () <i>Returns the name of the current adapter.</i>	
	int = <b>GetAimAssistOption</b> () Get current setting of the aim assist option.	
	GetAlignment	
	componentIDArray = <b>GetAllCommanders</b> (controllableID) Returns the chain of commanders for the specified controllable.	

`extensionSettingsTable = GetAllExtensionSettings()`  
*Returns the settings for all registered extensions.*

#### **extensionSettingsTable**

> [Expand source](#)

```
extensionSettingsTable = {  
  [extensionid] = {  
    enabled = is extension enabled?,  
    sync = is extension synced?  
  },  
  ...  
}
```

`upgradeArray = GetAllMacroUpgrades(macro, sequence, stage, buildlimit)`  
*Returns information about all upgrades on a specified macro, with options to limit the information to parts of the buildplan.*

#### **upgradeArray**

> [Expand source](#)

```
upgradeArray = {  
  [1] = {  
    upgrade = upgrade type,  
    name = upgrade name,  
    total = total of available slots  
  },  
  [2] = { ... }  
}
```

`stringArray = GetAllStatIDs()`  
*Returns the ids of all statistic properties.*

`upgradeTable = GetAllUpgrades(destructibleID [, buildlimit] [, tag])`  
*Returns information about all upgrades of the specified destructible, with options to limit the information to the base buildplan and/or upgrades with the specified tag.*

### upgradeTable

[Expand source](#)

```
upgradeTable = {
  upgradetype = {
    name = upgrade name,
    ware = upgrade ware,
    tags = upgrade tags,
    total = total number of upgrades,
    free = number of free upgrades,
    operational = number of operational
upgrades,
    construction = number of upgrades in
construction
  },
  totaltotal = sum of total number of
upgrades,
  totalfree = sum of number of free upgrades,
  totaloperational = sum of the number of
operational upgrades,
  totalconstruction = sum of the number of
upgrades in construction,
  estimated = is data filtered?
}
```

weaponTable = **GetAllWeapons**(destructibleID)  
Returns information of all primary weapons and missiles of the destructible.

```
weaponTable > Expand source
weaponTable = {
  weapons = {
    [1] = {
      component = component id,
      macro = macro name,
      name = weapon name,
      range = bullet range,
      dps = hull and shield damage per sec
    },
    [2] = { ... }
  },
  missiles = {
    [1] = {
      name = missile name,
      speed = missile range,
      damage = explosion damage,
      macro = macro name,
      amount = amount number
    },
    [2] = { ... }
  }
}
```

wareKeyTable = **GetAmmoCountAfterShoppingList**(containerID)  
Returns information of ammo counts on the container after the trade queue is resolved.

```
wareKeyTable > Expand source
wareKeyTable = {
  [ware] = amount,
  ...
}
```

Call **SetVirtualCargoMode** on containerID first, otherwise result will be empty.

int = **GetAntiAliasModeOption**()  
Returns the current anti alias mode setting.

aaModesArray = **GetAntiAliasModes**()  
Returns all possible anti alias modes currently possible.

componentID or nil = **GetAutoPilotTarget**()  
Returns the target of the auto pilot, if any.

bool = **GetAutorollOption**()  
Returns current autoroll setting.

bool = **GetAutosaveOption**()  
Returns current autosave setting.

fontSize = **GetBestFontSize**(fontname, fontsize)  
Returns the best matching font size available for the specified font and size.



**bonusContentArray = GetBonusContentData()**  
*Returns information about all bonus content.*

**bonusContentArray**

[Expand source](#)

```
bonusContentArray = {  
  [1] = {  
    appid = appID,  
    name = "Name",  
    description = "Description",  
    path = "path\\to\\content",  
    optional = true|false,  
    owned = true|false,  
    installed = true|false,  
    changed = true|false, -- installation  
state already changed, can't be changed again  
  },  
  [2] = ...  
}
```

**bool = GetBoostToggleOption()**  
*Returns the current boost toggle setting.*

**... = GetBudgetData(entityID, ...)**  
*Returns properties of the budget of the specified entity. Possible properties are: "min", "max", "responsibility".*

**componentID or nil = GetBuildAnchor(containerID or buildmoduleID)**  
*Returns the build anchor of a container or buildmodule, e.g. the station a CV is building.*

**builderMacrosArray = GetBuilderMacros(containerID or buildmoduleID)**  
*Returns all information about all macros a container or buildmodule can build.*

**builderMacrosArray**

[Expand source](#)

```
builderMacrosArray = {  
  [1] = {  
    macro = macro,  
    name = name  
  },  
  [2] = { ... }  
}
```

**method = GetBuildProductionMethod(containerID or buildmoduleID, macro)**  
*Returns the production method used by the specified container or buildmodule to produce macro.*

**duration = GetBuildSlotDuration(macro or componentID, sequence, stage)**  
*Returns the time the construction of the specified build slot takes on average.*

`resourceTable = GetBuildSlotResources(containerID or buildmoduleID, macro or componentID, sequence, stage, upgradePlanArray)`  
Returns the needed resources to build the specified build slot with this container or buildmodule.

### resourceTable

[Expand source](#)

```
resourceTable = {  
  [1] = {  
    ware = ware id,  
    name = ware name,  
    cycle = amount,  
    price = price  
  },  
  [2] = { ... },  
  ...,  
  totalprice = totalprice  
}
```

### upgradePlanArray

[Expand source](#)

```
upgradePlanArray = {  
  [1] = {  
    [1] = upgradetype,  
    [2] = level  
  },  
  [2] = { ... }  
}
```

`moduleArray = GetBuildStageModules(macro or componentID, sequence, stage)`  
Returns all modules included in the specified buildslot.

### moduleArray

[Expand source](#)

```
moduleArray = {  
  [1] = {  
    macro = module macro,  
    name = module name,  
    library = moduletypes library  
  },  
  [2] = { ... }  
}
```

`upgradeTable = GetBuildStageUpgrades(containerID, sequence, stage [, buildlimit] [, tags])`  
*Returns information about all upgrades of the specified buildslot, with options to limit the information to the base buildplan and/or upgrades with the specified tag.*

### upgradeTable

[Expand source](#)

```
upgradeTable = {
  upgradetype = {
    name = upgrade name,
    tags = upgrade tags,
    total = total number of upgrades,
    operational = number of operational
upgrades,
    construction = number of upgrades in
construction
  },
  totaltotal = sum of total number of
upgrades,
  totaloperational = sum of the number of
operational upgrades,
  totalconstruction = sum of the number of
upgrades in construction
}
```

`buildTreeTable = GetBuildTree(macro or componentID)`  
*Returns information about the build tree of the specified macro or component.*

### buildTreeTable

[Expand source](#)

```
buildTreeTable = {
  name = "Station (base module) name for
display",
  [1] = {
    sequence = "sequencename"
    name = "Sequence name for display",
    currentstage = currentstagenumber or 0,
-- only if component supplied
    [1] = {
      stage = stagenumber,
      name = "Stage module name for display",
      primarycomponent = Primary component of
that stage, if it is already build (Note: At
the moment gives just the first of the primary
components)
    },
    [2] = ...
  },
  [2] = ...
}
```

GetButtonColor

GetButtonFont

	GetButtonHotkeyDetails	
	GetButtonIcon	
	GetButtonIcon2	
	GetButtonIcon2Color	
	GetButtonIcon2Offset	
	GetButtonIcon2Size	
	GetButtonIconColor	
	GetButtonIconOffset	
	GetButtonIconSize	
	GetButtonSwapIcon	
	GetButtonSwapIcon2	
	GetButtonText	
	GetButtonTextAlignment	
	GetButtonTextColor	
	GetButtonTextOffset	
	bool = <b>GetCaptureHQOption()</b> <i>Returns current capture hq setting.</i>	
	<p>wareKeyTable = <b>GetCargoAfterShoppingList</b>(containerID) <i>Returns information of ammo counts on the container after the trade queue is resolved.</i></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>wareKeyTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre>wareKeyTable = {   [ware] = amount,   ... }</pre> </div>	Call <b>SetVirtualCargoMode</b> on containerID first, otherwise result will be empty.
	GetCellContent	
	GetCellText	
	GetChildren	
	GetClosestAttackerDistance	
	GetClosestHighwaySection	
	GetClosestObserverDistance	
	componentIDArray = <b>GetClusters</b> ([showonmap]) <i>Returns an array of all player-known clusters.</i>	

changed in 4.10  
Beta 1

`collectableDataTable = GetCollectableData(collectableID)`  
Returns information on the content of the specified collectable.

```

collectableDataTable > Expand source

collectableDataTable = {
  type = "ammo" | "wares" | "shieldrestore",
  #if ammo:
    macro = ammo macro name,
    name = ammo name,
    icon = ammo icon,
    amount = ammo amount
  #elseif wares:
    wares = {
      [1] = {
        ware = ware,
        amount = amount
      },
      [2] = ...
    },
    money = contained money
  #elseif shieldrestore:
    restoretype = "duration" | "hp" | "percent",
    value = value (depending on restoretype)
  #end
}

```

shieldrestore value changed to return a floating point number (rather than an integer number) in 4.10 Beta 1

`bool = GetCollisionAvoidanceAssistOption()`  
Returns current collision avoidance assist option.

`GetColor`

`commanderID or nil = GetCommander(controllableID)`  
Returns commander of specified controllable, if any.

`entityID or nil = GetCommanderEntity(controllableID)`  
Returns commanding entity of specified controllable, if any.

changed in 3.60  
Beta 3, 4.00  
Beta 2, 4.00  
Beta 3 and 4.00  
Beta 5

`... = GetComponentData(componentID, ...)`  
Returns properties of the specified component. Possible properties are:

```

Possible properties > Expand source

- Any Component:
  "name", "uname", "description", "image",
  "owner", "ownername", "ownershortname",
  "ownericon", "size", "zone", "zoneid",
  "sector",
  "sectorid", "cluster", "clusterid",
  "galaxyid", "macro", "parent",
  "revealpercent", "isplayerowned", "isfriend",
  "isenemy", "iskill", "isdock"
  "wares", "caninitiatecomm", "sequence",
  "stage"

- Destructible:
  "hull", "hullmax", "hullpercent", "shield",
  "shieldmax", "shieldpercent",

```

Container "tradewares" property added in 3.60 Beta 3  
Entity "productionmoney" property added in 3.60 Beta 3  
Ship "nextjumptime" property added in 4.00 Beta 2  
Ship "isjumpdrivecharging" property added in 4.00 Beta 2  
Ship "isjumpdrivebusy" property added in 4.00 Beta 2  
Station "istradestation" property added in 4.00 Beta 3  
Entity "isitemtrader" property added in 4.00 Beta 3  
Ship "isdocked" property added in 4.00 Beta 5  
Ship "isdocking" property added in 4.00 Beta 5

```
"efficiencyfactor", "efficiencybonus",
"efficiencythreshold",
  "isfunctional"

- Controllable:
  "controlentity", "pilot", "defencenpc",
"tradenpc", "engineer", "boardingnpc",
"architect", "maxforwardspeed",
"maxradarrange"

- Defensible:
  "missilecapacity", "boardingresistance",
"boardingstrength"

- Object:
  "primarypurpose", "primarypurposename"

- Container:
  "engines", "shieldgenerators", "cargo",
"products", "availableproducts",
"allresources", "pureresources",
"intermediatewares", "buildingmodule",
"tradeoffers", "tradesubscription",
"numtrips", "isshipyard", "tradewares"

- Ship:
  "hasjumpdrive", "basejumpcost",
"issellable", "nextjumptime",
"isjumpdrivecharging", "isjumpdrivebusy",
"isdocked", "isdocking"

- Station:
  "istradestation"

- Entity:
  "typestring", "typeicon", "typename",
"skills", "combinedskill",
"experienceprogress", "neededexperience",
"skillsvisible",
  "repairqueue", "aicommandstack",
"aicommand", "aicommanddraw", "aicommandparam",
"aicommandparam2", "aicommandaction",
"aicommandactionparam",
  "iscontrolentity", "money", "commissions",
"discounts", "hasownaccount", "isspecialist",
"isremotecommmable", "wantedmoney",
"ismissionactor",
  "buildmodule", "productionmoney",
"isitentrader"

- Radar:
  "radarrange"



- CollectableWares/Crate:
  "money"

- ControlPanel
  "requireditems", "iscontrolpanelhacked",
"hackexpirationtime"
```


- Gate  
"destination"
- Highway, HighwayEntryGate, HighwayExitGate  
"nextdestname"
- Space  
"mapshortname"
- Dockingbay  
"populationtype"
- Production Module  
"isproducing"
- Zone  
"istemporaryzone", "hasjumpbeacon",  
"exitsector"

	<pre> - Lockbox   "numlocks" </pre>	
removed 3.50 Beta 3	<del>GetComponentFactionName</del>	
	<p>name = <b>GetComponentName</b>(componentID, fontname, fontsize, availablewidth[, omitfaction[, forcewidth]])  <i>Returns the name of the component. If the component is an object the name is formatted accordingly to the given parameters.</i></p>	
	<p>bool = <b>GetConfineMouseOption</b>()  <i>Returns the current confine mouse option.</i></p>	
	<p>objectIDArray = <b>GetContainedObjectsByOwner</b>(faction, [space])  <i>Returns an array of objects owned by the specified faction. Optionally results can be confined to a specified space.</i></p>	
	<p>shipIDArray = <b>GetContainedShips</b>([space], [shownonmap])  <i>Returns an array of ships in the specified space or the galaxy.</i></p>	
	<p>shipIDArray = <b>GetContainedShipsByOwner</b>(faction, [space])  <i>Returns an array of ships owned by the specified faction. Optionally results can be confined to a specified space.</i></p>	
	<p>stationIDArray = <b>GetContainedStations</b>([space], [shownonmap], [includeconstruction])  <i>Returns an array of stations in the specified space or the galaxy.</i></p>	
	<p>stationIDArray = <b>GetContainedStationsByOwner</b>(faction, [space], [includeconstruction])  <i>Returns an array of stations owned by the specified faction. Optionally results can be confined to a specified space.</i></p>	
	<p>price = <b>GetContainerWarePrice</b>(containerID or buildmoduleID, ware, buysellswitch)  <i>Returns the price the specified container buys/sells the ware.</i></p>	
	<p>contextID or nil = <b>GetContextByClass</b>(componentID, classname [, includeself])  <i>Returns the context of the specified class of the component.</i></p>	
	<p>entityID or nil = <b>GetControlEntity</b>(controllableID, commandertype)  <i>Returns the control entity of the specified controllable of the given type.</i></p>	
changed in 4.21 Beta 1 (VR: 4.20) and 4.30 RC2 (VR: 4.30 Hotfix 2)	<p>mode, angle = <b>GetControllerInfo</b>()  <i>Returns the current input mode ("mouseSteering" "mouseCursor" "gamepad" "touch" "joystick") and the joystick input angle (only reasonable in "touch", "joystick", or "gamepad" mode).</i></p> <p><i>The angle value will be -1, if the joystick is in its safe area. Otherwise it returns the angle in radian [0..2] (with 0 corresponding to the joystick pointing upwards - rotation is clockwise).</i></p>	4.21 Beta 1 (VR: 4.20): Added "touch" as new possible input mode. 4.30 RC 2 (VR: 4.30 Hotfix 2): Added "joystick" as new possible input mode.
	<p>bool = <b>GetCrashReportOption</b>()  <i>Returns the current crash report setting.</i></p>	
	<p>time = <b>GetCurRealTime</b>()  <i>Returns the real time since session start in seconds.</i></p>	



	<p>sequence, stage, progress, upgradePlanArray = <b>GetCurrentBuildSlot</b>(containerID)  <i>Returns information about the current build process of the specified container.  Returns nil if not currently building.</i></p> <div data-bbox="365 235 1070 596" style="border: 1px solid #ccc; padding: 10px;"> <p><b>upgradePlanArray</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre>upgradePlanArray = {   [1] = {     [1] = upgradetype,     [2] = level   },   [2] = { ... } }</pre> </div>	
	<p>resourceTable = <b>GetCurrentBuildSlotResources</b>(containerID or buildmoduleID)  <i>Returns the needed resources to build the current build slot under construction by this container or buildmodule.</i></p> <div data-bbox="365 737 1070 1220" style="border: 1px solid #ccc; padding: 10px;"> <p><b>resourceTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre>resourceTable = {   [1] = {     ware = ware id,     name = ware name,     cycle = amount,     price = price   },   [2] = { ... },   ...,   totalprice = totalprice }</pre> </div>	
<p>deprecated in 4.31 Beta 1</p>	<p><b>DEPRECATED</b> string = <b>GetCurrentModuleName</b>()  <i>Returns the name of the current game module.</i></p>	<p> <b>DEPRECATED</b> - use FFI function IsStartmenu() instead</p>
<p>deprecated in 4.22 Beta 2 (VR: 4.22)</p>	<p><b>DEPRECATED</b> time = <b>GetCurTime</b>()  <i>Returns the ingame time in seconds.</i></p>	<p> <b>DEPRECATED</b> - use FFI function GetCurrentGameTime() instead</p>

<p>changed in 4.30 RC 1 (VR: 4.30)</p>	<p>string table = <b>GetDate</b>([format [,time]])  <i>Returns a string or a table containing date and time, formatted according to the given string "format".</i>  <i>Refer to the Lua documentation about os.date() for details.</i></p>	<p>As of XR 4.30 RC 1 GetDate() supports only the following format specifiers:      ✓ <a href="#">List of format specifiers</a></p> <ul style="list-style-type: none"> <li>• %a</li> <li>• %A</li> <li>• %b</li> <li>• %B</li> <li>• %c</li> <li>• %d</li> <li>• %H</li> <li>• %I</li> <li>• %j</li> <li>• %m</li> <li>• %M</li> <li>• %p</li> <li>• %S</li> <li>• %U</li> <li>• %w</li> <li>• %W</li> <li>• %x</li> <li>• %X</li> <li>• %y</li> <li>• %Y</li> <li>• %Z</li> <li>• %%</li> </ul>
	<p>int = <b>GetDeadzoneOption</b>()  <i>Returns the current deadzone setting.</i></p>	
	<p>text, isSelectable, isImmediate, shortcutKey, RESERVED, mouseOverText = <b>GetDialogOption</b>(index)  <i>Returns information about the specified dialog option of the current active conversation.</i>  <i>text is the actual dialog menu button text</i>  <i>isSelectable indicates whether the option can be selected</i>  <i>isImmediate indicates whether the option is flagged as an immediate option</i>  <i>shortcutKey specifies the assigned shortcut key associated with the option</i>  <i>RESERVED is an internal value not to be used</i>  <i>mouseOverText is the text to display when mousing over the specified option</i></p>	
	<p>string = <b>GetDifficultyOption</b>()  <i>Returns the current difficulty setting.</i></p>	
	<p>bool = <b>GetDistortionOption</b>()  <i>Returns the current distortion setting.</i></p>	
	<p>dronecommanderID or nil = <b>GetDroneCommander</b>(drone)  <i>Returns the drone commander of the specified drone, if any.</i></p>	
	<p>dronesIDArray = <b>GetDrones</b>(shipID)  <i>Returns an array of all drones of that ship.</i></p>	<p>See also <b>GetUnitStorageData</b>.</p>
	<p>GetDroneWeapon</p>	
	<p>GetEditBoxCloseMenuOption</p>	
	<p>GetEditBoxColor</p>	
	<p>GetEditBoxFont</p>	
	<p>GetEditBoxHotkeyDetails</p>	
	<p>GetEditBoxText</p>	
	<p>GetEditBoxTextAlignment</p>	
	<p>GetEditBoxTextColor</p>	
	<p>number = <b>GetEffectDistanceOption</b>()  <i>Returns the current effect distance setting.</i></p>	

	<p>stringArray = <b>GetEfficiencyUpgrades</b>(destructible)  <i>Returns an array with the macro names of all efficiency upgrades connected to the specified destructible.</i></p>									
	<p><b>DEPRECATED</b> bool = <b>GetEMCutsceneOption</b>()  Returns false</p>	 <b>DEPRECATED</b> - do not use								
	<p>.. = <b>GetEntityTypeData</b>(entitytype, ...)  <i>Returns properties of the specified component. Possible properties are: "name", "icon"</i></p>									
	<p>GetError</p>									
<p>changed in 4.30  RC 1 (VR: 4.30)</p>	<p>severity = <b>GetErrorSeverity</b>(logMessageIndex)  <i>Returns the severity of the specified log message.</i></p> <table border="1" data-bbox="321 533 1112 726"> <thead> <tr> <th>severity</th> <th>level</th> </tr> </thead> <tbody> <tr> <td>info</td> <td>1</td> </tr> <tr> <td>error</td> <td>2</td> </tr> <tr> <td>panic</td> <td>5</td> </tr> </tbody> </table>	severity	level	info	1	error	2	panic	5	<p>4.30 RC 1 changed the severity levels and introduced the new info level.</p>
severity	level									
info	1									
error	2									
panic	5									
	<p>GetErrorTimestamp</p>									
	<p>GetEventMonitorDetailsBridge</p>									
<p>removed in 3.50  RC 1</p>	<p><del>GetEventMonitorName</del></p>									

extensionTable = **GetExtensionList()**  
Returns information about all extensions.

```

extensionTable > Expand source
extensionTable = {
  [1] = {
    id = extension id,
    index = extension index,
    name = extension name,
    desc = extension description,
    author = extension author,
    date = extension date,
    location = extension location,
    version = extension version,
    enabled = extension enabled,
    sync = extension sync,
    personal = extension personal,
    isworkshop = is extension in workshop,
    error = extension error ID, if any,
    errortext = extension error text, if any,
    warning = workshop update warning, if any,
    warningtext = workshop update warning, if
any,
    dependencies = {
      id = dependency id,
      name = dependency name,
      version = dependency version
    }
  }
}

```

string or nil = **GetExtensionUpdateWarningText**(extensionid, personal)  
*Returns the update warning text of a workshop extension, if any.*

removed 3.50  
Beta 3

~~GetFactionDetails~~

GetFont

GetFontHeight

number = **GetFOVOption()**  
*Returns current fov setting.*

GetFrameBackgroundID

GetFrameOverlayID

GetFramePosition

amount = **GetFreeCargoAfterShoppingList**(containerID, ware)  
*Returns amount of free cargo space for the specified ware after the trade queue will be resolved.*

Call **SetVirtualCargoMode** on containerID first, otherwise result will be empty.

amount = **GetFreeUnitStorageAfterShoppingList**(containerID)  
*Returns amount for free unit storage space for the specified container.*

Call **SetVirtualCargoMode** on containerID first, otherwise result will be empty.

bool = **GetFullscreenDetailmonitorOption()**  
*Returns the current fullscreen detailmonitor setting.*

	<p>fullscreen, borderless = <b>GetFullscreenOption()</b> Returns the current fullscreen and borderless setting.</p>	
	<p>int = <b>GetGamepadModeOption()</b> Returns the current gamepad mode setting.</p>	
	<p>number = <b>GetGammaOption()</b> Returns the current gamma setting.</p>	
	<p>gateIDArray = <b>GetGates</b>(space [, showonmap]) Returns an array of gates in the specified space.</p>	
	<p>int = <b>GetGfxQualityOption()</b> Returns the current gfx quality setting.</p>	
	<p>bool = <b>GetGlobalSyncSetting()</b> Returns the current global extension sync setting.</p>	
	<p>int = <b>GetGlowOption()</b> Returns the current glow setting.</p>	
	<p>GetHeader</p>	
	<p>licenceTable = <b>GetHeldLicences</b>(faction) Returns a table with information about all held licences of the specified faction.</p> <div data-bbox="363 779 1070 1293" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>licenceTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre> licenceTable = {   [1] = {     id = licence id,     type = licence type,     name = licence name,     icon = licence icon,     price = licence price,     minrelation = minimum relation required,     faction = factionid   },   [2] = { ... } }</pre> </div>	
	<p>x, y = <b>GetHintPosition</b>(position) Returns the defined x, y coordinates of the specified position in percent with (0, 0) being in the upper left corner. Hint positions can be defined in parameters.xml.</p>	
	<p>productionColor, buildColor, storageColor, radarColor, dronedockColor, efficiencyColor, defenceColor, playerColor, friendColor, enemyColor, missionColor = <b>GetHoloMapColors()</b> Returns the colors defined for various holomap items in parameters.xml.</p> <div data-bbox="363 1562 1070 1887" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>Color</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre> Color = {   r = red (0-255),   g = green (0-255),   b = blue (0-255),   a = alpha (0-100) }</pre> </div>	

GetIconDetails

wareKeyTable = **GetIncomingRemoteWares**(wareType)  
*Returns information about incoming remote transport wares.*

**wareKeyTable**

[Expand source](#)

```
wareKeyTable = {  
  [ware] = amount,  
  ...  
}
```

actionTable = **GetInputActionMap**()  
*Returns the current input actions mapped.*

**actionTable**

[Expand source](#)

```
actionTable = {  
  [ActionID] = {  
    [1] = {  
      [1] = InputID Source,  
      [2] = InputID Code,  
      [3] = InputID Signum  
    },  
    [2] = { ... }  
  },  
  [ActionID] = { ... }  
}
```

inputProfileTable = **GetInputProfiles**()  
*Returns information about the current input profiles.*

**inputProfileTable**

[Expand source](#)

```
inputProfileTable = {  
  [1] = {  
    id = profile ID,  
    version = profile version,  
    filename = filename,  
    personal = is file in personal folder?,  
    name = profile name,  
    customname = profile custom name  
  },  
  [2] = { ... }  
}
```

**rangeTable = GetInputRangeMap()**  
*Returns the current input ranges mapped.*

### rangeTable

> [Expand source](#)

```
rangeTable = {  
  [RangeID] = {  
    [1] = {  
      [1] = InputID Source,  
      [2] = InputID Code,  
      [3] = InputID Signum  
    },  
    [2] = { ... }  
  },  
  [RangeID] = { ... }  
}
```

**stateTable = GetInputStateMap()**  
*Returns the current input states mapped.*

### stateTable

> [Expand source](#)

```
stateTable = {  
  [StateID] = {  
    [1] = {  
      [1] = InputID Source,  
      [2] = InputID Code,  
      [3] = InputID Signum  
    },  
    [2] = { ... }  
  },  
  [StateID] = { ... }  
}
```

**GetInteractiveObject**

**wareTable = GetInventory(entityID)**  
*Returns information about the wares held in the inventory of specified entity.*

### wareTable

> [Expand source](#)

```
wareTable = {  
  [ware1] = {  
    name = ware name  
    amount = ware amount  
    price = ware price  
  },  
  [ware2] = { ... }  
}
```

	<p>bool = <b>GetInversionSetting</b>(rangeid)  <i>Returns the current inversion setting for the specified input range.</i></p>	
	<p>joystickTable = <b>GetJoysticksOption</b>()  <i>Returns information about currently connected joysticks.</i></p> <div data-bbox="365 296 1070 653" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>joystickTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre>joystickTable = {   [1] = {     name = joystick name,     guid = joystick guid   },   [2] = { ... } }</pre> </div>	
	<p>jumpbeaconIDArray = <b>GetJumpBeacons</b>(zoneID, [showonmap])  <i>Returns an array of jumpbeacons in the specified zone.</i></p>	
	<p>bool = <b>GetLegacyMainMenuOption</b>()  <i>Returns the current legacy main menu setting.</i></p>	
	<p>bool = <b>GetLegacyShadersOption</b>()  <i>Returns the current legacy shaders setting.</i></p>	
	<p>libraryTable = <b>GetLibrary</b>(library)  <i>Returns information about the items in the specified library.</i></p> <div data-bbox="365 995 1070 1444" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>libraryTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre>libraryTable = {   [1] = {     id = item id,     name = item name,     icon = item icon     [parent = item parent]     [relation = player relation]   },   [2] = { ... } }</pre> </div>	
<p>changed in 4.22  Beta 2 (VR:  4.22)</p>	<p>entryTable = <b>GetLibraryEntry</b>(library, itemid)  <i>Returns detailed information about the specified library entry.</i></p> <div data-bbox="365 1562 1070 1969" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>entryTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre>entryTable = {   name = info name,   description = info description,   icon = info icon,   image = info image,   video = info video   [race = npc race,   faction = npc faction]   [hull = object hull,   maxspeed = object maxspeed,</pre> </div>	<p>"storagenames" field added in 4.22  Beta 2 (VR: 4.22)</p>



```
shield = object shield,
jumpdrive = object has jumpdrive?,
storagecapacity = storage capacity,
storageetags = storage tags,
storagenames = storage names,
unitcapacity = unit capacity,
productions = {
  [1] = {
    [1] = {
      id = product id,
      name = product name,
      cycle = product cycle time,
      amount = product amount
    },
    [2] = { ... }
  },
  [2] = { ... }
},
weapons = {
  [1] = {
    class = weapon / missile,
    id = weapon id,
    name = weapon name
  },
  [2] = { ... }
},
upgrades = {
  [1] = {
    upgrade = upgrade type,
    name = upgrade name,
    total = upgrade total amount
  },
  [2] = { ... }
}]
[primaryrace = faction primaryrace]
[avgprice = ware average price,
volume = ware volume,
transporttype = ware transport type,
methods = {
  [1] = method,
  [2] = ...
}]
[range = weapon range,
dps = weapon dps]
[speed = missile speed,
acceleration = missile acceleration,
damage = missile damage]
[hull = shieldgen hull,
shield = shieldgen shield]
[hull = turret hull,
range = turret range,
```

	<pre>dps = turret dps, rotation = turret rotation speed] }</pre>	
	<p>int = <b>GetLibrarySize</b>(library)  <i>Returns the size of the specified library.</i></p>	
	<p>GetLiveDataBridge</p>	
	<p>GetLoadingInfo</p>	
<p>changed in 4.00 Beta 3</p>	<p>string = <b>GetLocalizedKeyName</b>(type, id)  <i>Returns the name of the key assigned to the specified input action, state or range.</i></p>	
	<p>string = <b>GetLocalizedRawKeyName</b>(keycode)  <i>Returns the name of the specified keycode.</i></p>	
<p>added in 4.00 Beta 2</p>	<p>mouseX, mouseY = <b>GetLocalMouseClickPosition</b>()  <i>Returns the current mouse click position (nil, if the mouse was not clicked on the current environment).</i></p>	
	<p>mouseX, mouseY = <b>GetLocalMousePosition</b>()  <i>Returns the current mouse position (nil, if the mouse is not over the current environment).</i></p>	
	<p>number = <b>GetLODOption</b>()  <i>Returns the current lod setting.</i></p>	
	<p>logbookTable = <b>GetLogbook</b>(start, count [, category])  <i>Returns logbook entries in the specified range, optionally filtered by category.</i></p> <div data-bbox="365 997 1071 1449" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid #ccc; padding-bottom: 5px;"> <span><b>logbookTable</b></span> <span style="color: #4a7ebb;">› Expand source</span> </div> <pre>logbookTable = {   [1] = {     text = log text,     time = timestamp,     category = log category,     interaction = log interaction type,     interactioncomponent = log interaction component, dependent on type   [2] = ... }</pre> </div>	

	<p>resourceTable = <b>GetMacroBuildSlotResources</b>(macro, sequence, stage)  <i>Returns the needed resources to build the specified macro build slot.</i></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p style="text-align: center;"><b>resourceTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre>resourceTable = {   [1] = {     ware = ware id,     name = ware name,     cycle = amount,     price = price   },   [2] = { ... },   ...,   totalprice = totalprice }</pre> </div>	
<p>changed in 4.00 RC 1  changed in 4.22 Beta 2 (VR: 4.22)</p>	<p>... = <b>GetMacroData</b>(macro, ...)  <i>Returns properties of the specified macro. Possible properties are: "name", "icon", "efficiencyfactor", "efficiencybonus", "ware", "primarypurpose", "primarypurpose", "storagecapacity", "storagetags", "storagenames", "scanlevel"</i></p>	<p>"scanlevel" property added in 4.00 RC 1  as of 4.00 RC 1 "storagetags" no longer returns "fuel", if the queried object contains another storage type as well  "storagenames" property added in 4.22 Beta 2 (VR: 4.22)</p>
	<p>int = <b>GetMacroUnitStorageCapacity</b>(macro, sequence, stage [, buildlimit])  <i>Returns the capacity of unit storage on the specified macro build slot.</i></p>	
	<p>joystickIndexArray = <b>GetMappedJoysticks</b>()  <i>Returns an array of joysticks currently mapped for input. Indices refer to the return value of <b>GetJoysticksOption</b>().</i></p>	
	<p>GetMaxRemoteControlRange</p>	
	<p>stringLength = <b>GetMaxTextLength</b>(text, fontname, fontsize, width)  <i>Returns the maximum length of the specified text which can fit in the given width in this font.</i></p>	
	<p>param, param2 = <b>GetMenuParameters</b>()  <i>Returns the parameters the current menu was opened with.</i></p>	
<p> <b>UNSUPPORTED</b></p>	<p>GetMessageScreenPosition</p>	
	<p>GetMiniGameCursorPosition</p>	
	<p>macroArray = <b>GetMiningUnitMacros</b>(macro)  <i>Returns all unit macros the specified macro can use to mine, if any.</i></p>	
<p>updated in 4.00 Beta 3  updated in 4.31 Beta 1</p>	<p>missionID, name, description, difficulty, maintype, subtype, faction, rewardmoney, rewardtext, briefingobjectives, activebriefingstep, opposingfaction, licence, timeleft, duration, abortable, guidancedisabled, associatedcomponent, hasObjective = <b>GetMissionDetails</b>(missionIndex, <del>fontname, fontsize, textwidth</del>)  <i>Returns details of the specified mission.</i></p>	<p>4.00 Beta 3: added hasObjective return value  4.31 Beta 1: removed fontname, fontsize, and textwidth parameters</p>
<p>updated in 4.31 Beta 1</p>	<p>objectiveText, objectiveIcon, timeout, progressName, curProgress, maxProgress, objectiveTargets = <b>GetMissionObjective</b>(missionIndex, <del>fontname, fontsize, textwidth</del>)  <i>Returns details of the specified mission's objective.</i></p>	<p>4.31 Beta 1: removed fontname, fontsize, and textwidth parameters</p>
	<p>GetMissionObjectiveBarInfo</p>	

	<p>name, desc, faction, type, level, reward, opposingfaction, licence, associatedcomponent = <b>GetMissionOfferAtConnection</b>(componentID, templateConnectionName)  <i>Returns details about the mission offer at the specified location.</i></p>	
<p>updated in 4.10 Beta 5  updated in 4.31 Beta 1</p>	<p>name, description, difficulty, maintype, subtype, faction, rewardmoney, rewardtext, briefingobjectives, activebriefingstep, opposingfaction, licence, timeleft, duration, abortable, guidancedisabled, associatedcomponent, offeractor, offercomponent = <b>GetMissionOfferDetails</b>(offerID, fontname, fontsize, textwidth)  <i>Returns details about the specified mission offer.</i></p>	<p>4.10 Beta 5: added offercomponent return value  4.31 Beta 1: removed fontname, fontsize, and textwidth parameters</p>
	<p>type = <b>GetModuleType</b>(destructibleID)  <i>Returns the module type of the specified destructible.</i></p>	
	<p>resourceTable = <b>GetNeededBuildSlotResources</b>(containerID or buildmoduleID)  <i>Returns the needed resources of the specified buildmodule.</i></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span><b>resourceTable</b></span> <span>&gt; <a href="#">Expand source</a></span> </div> <pre> resourceTable = {   [1] = {     ware = ware id,     name = ware name,     cycle = amount,     price = price   },   [2] = { ... },   ...,   totalprice = totalprice } </pre> </div>	
	<p>bool = <b>GetNonSquadShipsForTradeOffersOption</b>()  <i>Returns the setting whether non-squad ships should be included in the trade offers menu.</i></p>	
<p>updated in 4.31 Beta 1</p>	<p>notificationDetailsTable = <b>GetNotificationDetails</b>(notificationID)  <i>Returns details about the specified notification (or nil, if there's an error or if the notification doesn't exist).</i></p>	<p>4.31 Beta 1: added the interactionID to the returned table  4.31 Beta 1:  <b>DEPRECATED</b> the interactionDescriptor from the returned table</p>
	<p>GetNotificationPriority</p>	
	<p>componentIDArray = <b>GetNotUpgradesByClass</b>(destructibleID, class)  <i>Returns an array of all components included in the specified destructible of the specified class, which are not upgrades.</i></p>	
	<p>value or nil = <b>GetNPCBlackboard</b>(entityID, valueName)  <i>Return the blackboard variable "valueName" from the blackboard of the specified entity.</i></p>	
	<p>entityIDArray = <b>GetNPCs</b>(destructibleID)  <i>Returns an array of entities in the specified destructible.</i></p>	
	<p>entityIDArray = <b>GetNPCsInSectorOnStations</b>(sectorID, distance)  <i>Returns an array of entities on stations in the specified sector within the given distance to the player.</i></p>	
	<p>GetNumAttackers</p>	
	<p>GetNumErrors</p>	
	<p>int = <b>GetNumIncomingRemoteWares</b>(wareType)  <i>Returns the number of incoming remote transport wares for specified type.</i></p>	

	<p><b>int = GetNumLogbook</b>([category])  <i>Returns the number of logbook entries, optionally filtered by the given category.</i></p>	
	<p><b>int = GetNumMissions</b>()  <i>Returns the number of accepted missions.</i></p>	
	<p><b>GetNumObservers</b></p>	
	<p><b>subsystemsTable = GetObjectRepairStatus</b>(containerID)  <i>Returns information of all damaged subsystems of the specified container.</i></p> <div data-bbox="365 409 1071 1108" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid #ccc; padding-bottom: 5px;"> <span><b>subsystemsTable</b></span> <span>&gt; <a href="#">Expand source</a></span> </div> <pre> subsystemsTable = {   shieldgens = {     [1] = component id,     [2] = ...   },   engines = {     [1] = component id,     [2] = ...   },   weapons = {     [1] = component id,     [2] = ...   },   jumpdrives = {     [1] = component id,     [2] = ...   } } </pre> </div>	
	<p><b>GetOffset</b></p>	
	<p><b>GetOrigin</b></p>	
	<p><b>licenceTable = GetOwnLicences</b>(faction)  <i>Returns information of the own licences the specified faction offers.</i></p> <div data-bbox="365 1323 1071 1837" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid #ccc; padding-bottom: 5px;"> <span><b>licenceTable</b></span> <span>&gt; <a href="#">Expand source</a></span> </div> <pre> licenceTable = {   [1] = {     id = licence id,     type = licence type,     name = licence name,     icon = licence icon,     price = licence price,     minrelation = minimum relation required,     issellable = can the licence be bought   },   [2] = { ... } } </pre> </div>	
	<p><b>bool = GetPersonalizedCrashReportsOption</b>()  <i>Returns the setting to send personalized crash reports.</i></p>	

	dockingbayIDArray = <b>GetPlatforms</b> (containerID) <i>Returns an array of dockingbays that are platforms of the specified container.</i>	
	string = <b>GetPlayerActivity</b> () <i>Returns the current player mode.</i>	
	int = <b>GetPlayerAmmoCount</b> (macro) <i>Returns the ammo count of the specified macro on the player ship.</i>	
	contextID or nil = <b>GetPlayerContextByClass</b> (class) <i>Returns the context of the player with the specified class, if any.</i>	
	GetPlayerControlledHullShield	
	GetPlayerDroneDetails	
	int = <b>GetPlayerDroneSlots</b> () <i>Returns the number of drone slots on the player ship.</i>	
	GetPlayerDroneSpeed	
	playerDroneTable = <b>GetPlayerDroneStorageData</b> () <i>Returns information on the drones of the playership.</i>	
	<div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span><b>playerDroneTable</b></span> <span>&gt; <a href="#">Expand source</a></span> </div> <pre> playerDroneTable = {   stored = total stored amount,   [1] = {     name = "Unit Name",     amount = amountnumber,     macro = Macro Name   },   [2] = { ... } } </pre> </div>	
	GetPlayerDroneSystems	
	componentID = <b>GetPlayerEnvironmentObject</b> () <i>Returns the player's enviroment object.</i>	
	wareTable = <b>GetPlayerInventory</b> () <i>Returns information about the wares held in the inventory of specified entity.</i>	
	<div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span><b>wareTable</b></span> <span>&gt; <a href="#">Expand source</a></span> </div> <pre> wareTable = {   [ware1] = {     name = ware name     amount = ware amount     price = ware price   },   [ware2] = { ... } } </pre> </div>	
	money = <b>GetPlayerMoney</b> () <i>Returns the amount of money in the player account.</i>	

	<p>macro = <b>GetPlayerPrimaryShipCockpitMacro()</b>  <i>Returns the macro of the cockpit currently installed in the player ship.</i></p>	
	<p>shipID = <b>GetPlayerPrimaryShipID()</b>  <i>Returns the player primary ship.</i></p>	
	<p>macro = <b>GetPlayerPrimaryShipMacro()</b>  <i>Returns the macro of the player primary ship.</i></p>	
	<p>roomID = <b>GetPlayerRoom()</b>  <i>Returns the current player room.</i></p>	
	<p>GetPlayerShipHullShield</p>	
	<p>GetPlayerShipShieldDetails</p>	
	<p>GetPlayerSpeed</p>	
	<p>GetPlayerSteeringStrength</p>	
	<p>componentID = <b>GetPlayerTarget()</b>  <i>Returns the current player target.</i></p>	
	<p>adapterTable = <b>GetPossibleAdapters()</b>  <i>Returns information about all possible display adapters.</i></p> <div data-bbox="363 779 1070 1104" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>adapterTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre> adapterTable = {   [1] = {     name = adapter name,     ordinal = adapter ordinal   } } </pre> </div>	
	<p>productionInfoTable = <b>GetPossibleProducts(productionmoduleID)</b>  <i>Returns information about all possible products of the specified productionmodule.</i></p> <div data-bbox="363 1220 1070 1892" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>productionInfoTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre> productionInfoTable = {   [1] = {     ware = product id,     name = product name,     cycletime = cycle duration,     component = componentname     resources = {       [1] = {         ware = resource id,         name = resource name,         cycle = cycle amount       },       [2] = { ... }     }   },   [2] = { ... } } </pre> </div>	

`resolutionsTable = GetPossibleResolutions()`  
*Returns information about all possible resolutions.*

**resolutionsTable**

[Expand source](#)

```
resolutionsTable = {  
  [1] = {  
    height = resolution height in px,  
    width = resolution width in px  
  },  
  [2] = { ... }  
}
```

`GetPrimaryShipWeapon`

`entityIDArray = GetPrioritizedPlatformNPCs(dockingbayID)`  
*Returns an array of entities on the specified dockingbay.*



`productionDataTable = GetProductionModuleData(productionmoduleID)`  
Returns information about the specified productionmodule.

### productionDataTable

[Expand source](#)

```
if production queue empty: productionDataTable
= {
  state = "empty"
}
else: productionDataTable = {
  state = "ProductionState",
  cycletime = CycleTime (only if state ==
"producing", otherwise 0),
  cycleefficiency = cycle efficiency
percentage (100% default),
  remainingcycletime = Time remaining for this
cycle (only if state == "producing", otherwise
0),
  cycleprogress = percentage of current cycle
progress (only if state = "producing",
otherwise 0),
  remainingtime = time until out of resources
(does not take limited storage space into
account),
  products = {
    efficiency = product efficiency percentage
(100% default),
    [1] = {
      ware = "wareid",
      name = "Ware Name",
      amount = storageamount,
      cycle = cycleamount,
      component = componentname
    },
    [2] = { ... }
  },
  presources = { ... }, -- primary
resources, analogous to products table
  sresources = { ... }, -- secondary
resources, analogous to products table
  estimated = is data filtered?
}
```

`productionmoduleIDArray = GetProductionModules(destructible)`  
Returns an array with all productionmodules in the specified destructible.

`GetProgressElementDetails`

`string = GetRadarModuleName(objectID)`  
Returns the name of the macro with biggest radar range included in the specified object.

`int = GetRadarOption()`  
Returns the current radar setting.

`money or nil = GetReferenceProfit(container, ware, price, amount [, shoppinglistidx])`  
Returns the profit selling the specified ware from the given container at the given price will provide.

	<p>moduleArray = <b>GetRegisteredModules()</b>  <i>Returns information about all registered modules.</i></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>moduleArray</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre> moduleArray = {   id = module id,   name = module name,   description = gamestart description, if any,   image = gamestart image, if any } </pre> </div>	
removed 3.50 Beta 3	<del>GetRelationComponentStatus</del>	
removed 3.50 Beta 3	<del>GetRelationFactionStatus</del>	
added in 4.21 Beta 1 (VR: 4.20) changed in 4.31 Beta 1	<p>x, y, z = <b>GetRelativeMousePosition</b>(element[, sizedMousePoint])  <i>Returns the x/y/z-mouse cursor coordinates relative to the the elements own coordinate space. The return value range is [0, 1] if sizedMousePoint is not specified or false, and [0, maxExtends] if sizedMousePoint is set to true. 0/0/0 corresponding to the upper left front edge of the element and maxExtends corresponding to boxWidth/boxHeight/boxDepth of the specified elemnt.</i></p> <p><i>Note that the return value is undefined if the element is not pickable at all or wasn't hit by the mouse cursor in the current frame.</i></p>	4.31 Beta 1: added sizedMousePoint parameter
	GetRelativePlayerShipPosition	
	GetRemoteControlHUDIcons	
	<p>resourceTable = <b>GetRepairResources</b>(containerID or buildmoduleID, repairdestructibleID, sequence, stage, buildlimit, targethullfraction)  <i>Returns the resources needed to repair the specified repairdestructible by the given buildmodule to targethullfraction.</i></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>resourceTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre> resourceTable = {   [1] = {     ware = ware id,     name = ware name,     cycle = amount,     price = price   },   [2] = { ... },   ...,   totalprice = totalprice } </pre> </div>	
	<p>resolutionTable = <b>GetResolutionOption</b>([checkconfig])  <i>Returns the current resolution or the resolution set in config.xml.</i></p>	
	<p>number = <b>GetRumbleOption</b>()  <i>Returns the current controller rumble intensity setting.</i></p>	
	<p>productionmoduleID = <b>GetSalesRepProduction</b>(entityID)  <i>Returns the productionmodule in the same build slot as the specified entity.</i></p>	

savegameTable = **GetSaveList()**  
Returns information about existing savegames.

Performance critical, do not call unnecessarily.

### savegameTable

[Expand source](#)

```
savegameTable = {
  filename = save filename,
  name = savegame name,
  version = savegame version,
  time = formatted save date,
  rawtime = save date,
  playtime = played time,
  playername = player name,
  location = save location,
  money = player money,
  difficulty = save difficulty,
  mindifficulty = minimum difficulty the save
was ever set to,
  error = there was an error loading the
savegame (non-localized error message in name
property)
  [invalidgameid = the save is from an invalid
game],
  [invalidversion = save version is higher
than current game version],
  invalidpatches = {
    [1] = {
      id = patch id,
      name = patch name,
      state = patch state,
      requiredversion = version of this patch
required by the savegame,
      installedversion = version of this patch
that is installed
    },
    [2] = { ... }
  }
}
```

added in 4.10  
Beta 1

savegameTable = **GetSaveList2()**  
Returns available information about existing savegames.

*Note to trigger population of the list GetSaveList2() returns, call the FFI function ReloadSaveList() first (or make sure it was called before at some point). While this function returns the same data as GetSaveList(), it does only return the data which is available right now without blocking the game to wait for all savegame details having been processed. This allows for a more performant non-blocking approach than would be possible with GetSaveList().*


GetScreenAspects

getScreenInfo





GetSecondaryShipWeapon

sectorIDArray = **GetSectors**(clusterID)  
Returns an array of sectors in the specified cluster.

number = **GetSensitivitySetting**(rangeid)  
Returns the current sensitivity setting for the specified input range.

	<p><code>int = <b>GetShaderQualityOption()</b></code>  <i>Returns the current shader quality setting.</i></p>	
	<p><code>int = <b>GetShadowOption()</b></code>  <i>Returns the current shadow setting.</i></p>	
	<p><code>shoppingListTable = <b>GetShoppingList(containerID)</b></code>  <i>Returns information about the trade queue of the specified container.</i></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>shoppingListTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre>shoppingListTable = {   [1] = {     id = trade id,     name = ware name,     amount = trade amount,     minamount = trade minimum amount,     price = trade price,     isbuyoffer = is the container selling?,     isselloffer = is the container buying?,     averageprice = trade ware avg. price,     [station = trade partner,      stationname = trade partner's name,      stationzone = trade partner's zone name,      stationsectorid = trade partner's sector,]     iswareexchange = is it a ware exchange     trade?,     ispassive = is container passive?   },   [2] = { ... } }</pre> </div>	
	GetSize	
	GetSliderGranularity	
	GetSliderText	
	<b>DEPRECATED</b> GetSliderValues	 <b>DEPRECATED</b> - use GetSliderValues2() instead
added in 4.10 Beta 1	GetSliderValues2	
	GetSliderValueSuffix	
	<p><code>bool = <b>GetSoftShadowsOption()</b></code>  <i>Returns the current soft shadow setting.</i></p>	
	<p><code>bool = <b>GetSoundOption()</b></code>  <i>Returns the current global sound setting.</i></p>	
	<p><code>int = <b>GetSSAOOption()</b></code>  <i>Returns the current ssao setting.</i></p>	
	GetStandardButtons	
	<p><code>macroArray = <b>GetStandardUnitMacros(macro)</b></code>  <i>Returns an array of unit macros that are considered standard for the specified macro.</i></p>	

	<p>... = <b>GetStatData</b>(stat, ...)  <i>Returns properties of the specified statistic. Possible properties are: "exists", "value", "hidden", "displayname", "displayvalue"</i></p>	
	<p>bool = <b>GetSteeringNoteOption</b>()  <i>Returns the current steering note setting.</i></p>	
	<p>bool = <b>GetStopShipInMenuOption</b>()  <i>Returns the current setting whether the playership should stop while a menu is open.</i></p>	
	<p>storageDataTable = <b>GetStorageData</b>(destructibleID)  <i>Returns information about the storage capabilities in the specified destructible.</i></p> <div data-bbox="363 485 1071 1402" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid #ccc; padding-bottom: 5px;"> <span><b>storageDataTable</b></span> <span style="color: #4a90e2; font-size: 0.8em;">&gt; Expand source</span> </div> <pre style="font-family: monospace; font-size: 0.9em; margin-top: 5px;"> storageDataTable = {   modules = {     [1] = storagemodule,     [2] = ...   }   capacity = totalcapacityamount,   stored = totalstoredamount,   estimated = is data filtered?,   [1] = {     name = "cargobayname",     capacity = capacityamount,     stored = storedamount,     consumption = totalconsumption,     [1] = {       ware = "wareid",       name = "Ware Name",       amount = amountnumber,       volume = ware volume,       consumption = consumption and production of this ware,     },     [2] = ...   },   [2] = ... } </pre> </div>	
	<p>componentIDArray = <b>GetSubordinates</b>(controllableID [, commandertype])  <i>Returns an array with all subordinates of the specified controllable, optionally filtered by the commandertype.</i></p>	
	<p>string = <b>GetSubtitleOption</b>()  <i>Returns the current subtitle option.</i></p>	
	<p>GetTableColumnSpan</p>	
	<p>GetTableColumnWidths</p>	
	<p>GetTableFullHeight</p>	
	<p>GetTableInitialSelectedColumn</p>	
	<p>GetTableInitialSelectedRow</p>	
	<p>GetTableInitialTopRow</p>	
	<p>GetTableNumFixedRows</p>	

	GetTableNumRows	
	GetTableRowColor	
	GetTableRowHeight	
 <b>UNSUPPORTED</b>	GetTargetElementInfo	
	GetText	
deprecated in 4.31 Beta 1	<b>DEPRECATED</b> int = <b>GetTextHeightExact</b> (text, font, fontsize, width, wordwrap) <i>Returns the height in pixels of the specified text. Note that the width parameter specifies the maximum width (per row) to be displayed. The parameter is ignored, if wordwrap is false.</i>	 <b>DEPRECATED</b> - use FFI function GetTextHeight() instead
changed in 4.00 Beta 1	int, actualwidth = <b>GetTextNumLines</b> (text, font, fontsize, width) <i>Returns the number of lines of the specified text with the given font and width and the actual width of the wordwrapped text.</i>	4.00 Beta 1: added the second return value (actualwidth)
deprecated in 4.31 Beta 1	<b>DEPRECATED</b> int = <b>GetTextWidth</b> (text, fontname, fontsize) <i>Returns the width in pixels of the specified text.</i>	 <b>DEPRECATED</b> - use FFI function GetTextWidth() instead
	GetTimeLeft	
 <b>UNSUPPORTED</b>	GetTopTargetPriorityMessages	
	money = <b>GetTotalValue</b> (containerID [, sellprice] [, referenceContainerID]) <i>Returns the value of the specified container, optionally for selling and in reference to the price list of the reference container.</i>	

tradeData = **GetTradeData**(trade)  
Returns information about the specified trade.

### tradeDataTable

[Expand source](#)

```
tradeData = {
  id = trade id,
  ware = trade ware,
  name = ware name,
  amount = trade amount,
  desiredamount = trade desired amount,
  minamount = trade min amount,
  price = trade price,
  marketprice = price without
discounts/commissions,
  quantityfactor = market price / average
price,
  total price = price * amount,
  totalmarketprice = market price * amount,
  isbuyoffer = is it a buy offer?,
  isselloffer = is it a sell offer?,
  rebundle = rebundle units?,
  unbundle = unbundle units?
[station = trade container,
stationname = trade container name,
stationzone = trade container zone name,
stationzoneid = trade container zone,
stationsectorid = trade container sector,
isplayer = is trade container player owned,
pricemodifiers = {
  [1] = {
    name = modifier name,
    level = modifier level,
    amount = modifier amount,
    expire = modifier expire time
  },
  [2] = { ... }
},
faction = trade container faction,]
expire = trade expire time
}
```

changed in 4.00  
Beta 4

`tradeDataArray = GetTradeList([tradeship], [sortby], [indexbyid])`  
*Returns an array of player known trades, optionally indexed by id strings.*

Added "indexbyid" parameter in 4.00  
Beta 4

### tradeDataArray

[Expand source](#)

```
tradeDataArray = {
  [1 or IDString] = {
    id = trade id,
    ware = trade ware,
    name = ware name,
    amount = trade amount,
    desiredamount = trade desired amount,
    minamount = trade min amount,
    price = trade price,
    marketprice = price without
discounts/commissions,
    quantityfactor = market price / average
price,
    total price = price * amount,
    totalmarketprice = market price * amount,
    isbuyoffer = is it a buy offer?,
    isselloffer = is it a sell offer?,
    rebundle = rebundle units?,
    unbundle = unbundle units?
    [station = trade container,
    stationname = trade container name,
    stationzone = trade container zone name,
    stationzoneid = trade container zone,
    stationsectorid = trade container sector,
    isplayer = is trade container player
owned,
    pricemodifiers = {
      [1] = {
        name = modifier name,
        level = modifier level,
        amount = modifier amount,
        expire = modifier expire time
      },
      [2] = { ... }
    },
    faction = trade container faction,]
    expire = trade expire time
  },
  [2 or IDString] = { ... }
}
```



`restrictionsTable = GetTradeRestrictions(containerID)`  
*Returns information about trade restrictions for the specified container.*

### restrictionsTable

[Expand source](#)

```
restrictionsTable = {  
  faction = global restriction to trade with  
  other factions,  
  overrides = {  
    [ware] = bool,  
    ...  
  }  
}
```

`tradeArray = GetTradesAtConnection(componentID, templateConnectionName)`  
*Returns an array of trades at the specified connection.*

`tradeShipData = GetTradeShipData(shipID)`  
*Returns information about the specified ship related to trading.*

### tradeShipData

[Expand source](#)

```
tradeShipData = {  
  shipid = shipID,  
  name = ship name,  
  cargo = {  
    ware = ware,  
    name = ware name,  
    amount = ware amount,  
    volume = ware volume  
  },  
  queue = {  
    [1] = {  
      id = trade id,  
      name = ware name,  
      amount = trade amount,  
      minamount = trade min amount,  
      price = trade price,  
      isbuyoffer = is the ship selling?,  
      isselloffer = is the ship buying?,  
      averageprice = trade ware avg. price,  
      [stationname = trade partner name,  
      stationsectorid = trade partner sector]  
    },  
    [2]= { ... }  
  },  
  cargocurrent = currently used cargo space,  
  cargomax = cargo capacity,  
  cargofree = free cargo capacity,  
  numtrips = number of planned trips  
}
```

tradeShipdataArray = **GetTradeShipList()**  
Returns an array with information about ships related to trading.

### tradeShipdataArray

[Expand source](#)

```
tradeShipdataArray = {
  [1] = {
    shipid = shipID,
    name = ship name,
    cargo = {
      ware = ware,
      name = ware name,
      amount = ware amount,
      volume = ware volume
    },
    queue = {
      [1] = {
        id = trade id,
        name = ware name,
        amount = trade amount,
        minamount = trade min amount,
        price = trade price,
        isbuyoffer = is the ship selling?,
        isselloffer = is the ship buying?,
        averageprice = trade ware avg. price,
        [stationname = trade partner name,
        stationsectorid = trade partner
sector]
      },
      [2]= { ... }
    },
    cargocurrent = currently used cargo space,
    cargomax = cargo capacity,
    cargofree = free cargo capacity,
    numtrips = number of planned trips
  },
  [2] = { ... }
}
```

macroArray = **GetTransportUnitMacros**(macro)  
Returns an array of unit macros suitable for transport of cargo for the specified macro.

x2d, y2d, z, onscreen, width, height = **GetUIElementRectangleScreenPosition**(posID, minRecSize, maxRecSize)  
Retrieves the positioning information for the given target element in rectangle mode (only applicable in non-worldspace mode).  
width and height indicate the size of the rectangle and is ensured to always be a multiply of 2.  
For the other return values, refer to `GetUIElementScreenPosition()`.

x2d, y2d, z, onscreen = **GetUIElementScreenPosition**(posID, iconSizeOnScreen, iconSizeOffScreen)  
Retrieves the positioning information for the given target element (only applicable in non-worldspace mode).  
x2d and y2d are ranging from -viewWidth/2 (or -viewHeight/2) to +viewWidth/2 (or +viewHeight/2) with 0/0 being the center of the screen and -viewWidth/2-viewHeight/2 being the lower left corner of the screen.  
z is the position between the clipping planes ranging form 0 to 1.  
onscreen indicates whether the elemnt is actually on- or off-screen.

	<p>relation = <b>GetUIRelation</b>(faction) <i>Returns the relation of the player to the specified faction.</i></p>	
<p>changed in 4.00 Beta 6</p>	<p>unitTable = <b>GetUnitStorageData</b>(defensibleID [, unitcategory] [, virtualammo][, excluderestricted]) <i>Returns information about the units stored on the specified defensible. Optionally filtered by unit category.</i></p> <div data-bbox="365 336 1071 882" style="border: 1px solid #ccc; padding: 10px;"><p><b>unitTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p><pre>unitTable = {   capacity = capacityamount,   stored = storedamount,   [categorystored = storedamount of this category]   [1] = {     macro = "macroname",     name = "Unit Name",     amount = amountnumber,     unavailable = unavailablenumber,   },   [2] = { ... } }</pre></div>	<p>If you want to use the virtualammo flag, call <b>SetVirtualCargoMode</b> first.</p> <p>4.00 Beta 6: added excluderestricted parameter</p>

resourceTable = **GetUpgradesResources**(containerID or buildmoduleID, upgradeDestructibleID, sequence, stage, buildlimit, upgradePlanArray)  
Returns the needed resources to build the specified upgrades with this container or buildmodule.

```

resourceTable > Expand source
resourceTable = {
  [1] = {
    ware = ware id,
    name = ware name,
    cycle = amount,
    price = price
  },
  [2] = { ... },
  ...,
  totalprice = totalprice
}

```

```

upgradePlanArray > Expand source
upgradePlanArray = {
  [1] = {
    [1] = upgradetype,
    [2] = level
  },
  [2] = { ... }
}

```

string = <b>GetVersionString</b> () <i>Returns the current game version.</i>	
number = <b>GetViewDistanceOption</b> () <i>Returns the current view distance setting.</i>	
GetViewSize	
number = <b>GetVolumeOption</b> (volumetype) <i>Returns the current volume setting of the specified type.</i>	
bool = <b>GetVSyncOption</b> () <i>Returns the current vsync setting.</i>	
int = <b>GetWareCapacity</b> (containerID, ware [, ignoreexisting = true]) <i>Returns the capacity of the specified container for the given ware. By default existing cargo is ignored.</i>	

<p>changed in 4.00 Beta 1, 4.00 Beta 4, 4.00 Beta 5</p>	<p>... = <b>GetWareData</b>(ware, ...) Returns properties of the specified component. Possible properties are: <i>"name"</i>, <i>"shortname"</i>, <i>"description"</i>, <i>"icon"</i>, <i>"avgprice"</i>, <i>"minprice"</i>, <i>"maxprice"</i>, <i>"pricerange"</i>, <i>"tags"</i>, <i>"illegal"</i>, <i>"inventory"</i>, <i>"component"</i>, <i>"transport"</i>, <i>"image"</i>, <i>"video"</i>, <i>"hasproductionmethod"</i>, <i>"productionmethod"</i>, <i>"products"</i>, <i>"resources"</i>, <i>"buyprice"</i>, <i>"volatile"</i>, <i>"gift"</i>, <i>"allowdrop"</i>, <i>"groupID"</i>, <i>"groupName"</i>, <i>"groupIcon"</i>, <i>"isweaponmod"</i>, <i>"isammo"</i>, <i>"issinglecraft"</i></p>	<p>"allowdrop" property added in 4.00 Beta 1 "groupID" property added in 4.00 Beta 1 "groupName" property added in 4.00 Beta 1 "groupIcon" property added in 4.00 Beta 1 "isweaponmod" property added in 4.00 Beta 4 "isammo" property added in 4.00 Beta 4 "issinglecraft" property added in 4.00 Beta 5</p>
---	--	---

tradeDataArray = **GetWareExchangeTradeList**(tradingshipID, tradedcontainerID [, sortby])  
*Returns information about possible ware exchange trade between the two trade partners.*

Call **SetVirtualCargoMode** on tradingshipid and tradedcontainerid first, otherwise result will be empty.

```

tradeDataArray > Expand source
tradeDataArray = {
  [1] = {
    id = trade id,
    ware = trade ware,
    name = ware name,
    amount = trade amount,
    desiredamount = trade desired amount,
    minamount = trade min amount,
    price = trade price,
    marketprice = price without
discounts/commissions,
    quantityfactor = market price / average
price,
    total price = price * amount,
    totalmarketprice = market price * amount,
    isbuyoffer = is it a buy offer?,
    isselloffer = is it a sell offer?,
    rebundle = rebundle units?,
    unbundle = unbundle units?
    [station = trade container,
    stationname = trade container name,
    stationzone = trade container zone name,
    stationzoneid = trade container zone,
    stationsectorid = trade container sector,
    isplayer = is trade container player
owned,
    pricemodifiers = {
      [1] = {
        name = modifier name,
        level = modifier level,
        amount = modifier amount,
        expire = modifier expire time
      },
      [2] = { ... }
    },
    faction = trade container faction,]
    expire = trade expire time
  },
  [2] = { ... }
}

```

int = **GetWareProductionLimit**(containerID, ware)  
*Returns the production limit for the specified ware on the given container.*


GetWordWrap



zoneIDArray = **GetZones**(sectorID)  
*Returns an array of zones contained in the specified sector.*





	<p>yieldTable = <b>GetZoneYield</b>(zoneID [, skipmaxyields])  <i>Returns information about yields in the specified zone.</i></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>yieldTable</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre>yieldTable = {   [1] = {     ware = ware,     name = ware name,     amount = current yield,     max = max yield   },   [2] = { ... },   ...,   snapshottime = time when data was taken, 0   if live. }</pre> </div>	<p>Performance critical, set skipmaxyields = true, if possible.</p>
	<p>bool = <b>HasAllResourcesToCraft</b>(entityID, ware)  <i>Checks whether the specified entity has all ingredients to craft the specified ware.</i></p>	
<p>added in 4.00 Beta 3</p>	<p>bool = <b>HasContainerStockLimitOverride</b>(containerID, ware)  <i>Returns whether the ship/station has an override for the stock limit of a particular ware defined.</i></p>	
	<p>bool = <b>HasContainerWarePriceOverride</b>(containerID or buildmoduleID, ware, buysellswitch)  <i>Checks whether the specified ware has a price override on the given container.</i></p>	
	<p>HasDroneControl</p>	
	<p>HasFlightControl</p>	
	<p>bool = <b>HasLicence</b>(faction, licencetype, licencefaction)  <i>Checks whether the specified faction holds the given licence of the licence faction.</i></p>	
	<p>HasShipControl</p>	
	<p>bool = <b>HasShipyards</b>(spaceID)  <i>Checks whether the specified space includes a shipyard.</i></p>	
	<p>HasSliderTwoScales</p>	
	<p>bool = <b>HasTag</b>(componentID, templateConnectionName, tagname)  <i>Checks whether the specified connection has the given tag.</i></p>	
	<p><b>DEPRECATED</b> int = <b>HasToRestoreOptionsMenu</b>()  <i>Returns static variable to re-init the options menu after ui reload, e.g. on device lost.</i></p>	<p> <b>DEPRECATED</b> - This solution for static variables is deprecated. Use &lt;savedvariable /&gt; in ui.xml instead.</p>
	<p>bool = <b>HaveExtensionSettingsChanged</b>()  <i>Checks whether extension settings changed in this session.</i></p>	
	<p>HideMissionObjective</p>	
	<p>HidePresentation</p>	
	<p>HideView</p>	
	<p><b>IncStatValue</b>(stat[, addvalue])  <i>Increases the specified statistic by the given value.</i></p>	
	<p><b>InstallSteamDLC</b>(appid)  <i>Installs the specified steam dlc.</i></p>	

	<b>InterruptPlayerComputer()</b> Stops the current player ship computer line.	
	IsBorderEnabled	
	IsButtonActive	
	<b>IsCheatVersion()</b> <i>Checks whether the game has cheats enabled.</i>	
	bool <b>IsComponentClass</b> (componentID, classname) <i>Checks whether the specified component is of the given class.</i>	
	bool <b>IsComponentConstruction</b> (componentID) <i>Checks whether the specified component is under construction.</i>	
	bool = <b>IsComponentOperational</b> (componentID) <i>Checks whether the specified component is operational.</i>	
removed in 3.50 Beta 2	<del>IsComponentValid</del>	replaced by IsValidComponent()
	bool = <b>IsContainerOperationalRangeSufficient</b> (containerID, spaceID) <i>Checks whether the given operational range of the specified container is sufficient to operate normally.</i>	
	IsDialogActive	
	bool = <b>IsDockingPossible</b> (shipID, dockingbayID) <i>Checks whether the specified ship is allowed to dock at the given dockingbay.</i>	
	bool = <b>IsFactionKnown</b> (faction) <i>Checks whether the specified faction is known to the player.</i>	
	IsFadeEnabled	
	IsFirstPerson	
	bool = <b>IsGamepadActive</b> () <i>Check whether the gamepad is active.</i>	
	bool = <b>IsInfoUnlockedForPlayer</b> (componentID, info) <i>Check whether the specified information of the given component is available to the player.</i>	
	IsInteractive	
	bool = <b>IsKnownItem</b> (library, itemid) <i>Checks whether the specified item is known in the given library.</i>	
	bool = <b>IsMacroClass</b> (macro, classname) <i>Checks whether the specified macro is of the given class.</i>	
changed in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	bool = <b>IsObstructed</b> (posID, canBeObstructedByOwnComponent[, ignoreCockpitObstruction]) <i>Checks whether the specified target element (identified via the posID) is obstructed against the player view.</i> <i>If canBeObstructedByOwnComponent is set to true, then the target element can also be obstructed by elements on its own geometry (for example a target element on a capital ship would be obstructed by parts on the capital ship itself).</i> <i>If ignoreCockpitObstruction is set to true, then the player cockpit geometry will not cause the target element to be determined as being obstructed. (defaults to true - which equals the behavior pre 4.20 Hotfix 1).</i>	"ignoreCockpitObstruction" argument was added in 4.21 Beta 1 (VR:4.20 Hotfix 1)
	bool = <b>IsPlayerFirstPerson</b> () <i>Checks whether the player is in first person mode.</i>	
	bool = <b>IsSameComponent</b> (component1ID, component2ID) <i>Helper function to compare two luaIDs or 64-bit integer representing components.</i>	
	bool = <b>IsSameTrade</b> (trade1ID, trade2ID) <i>Helper function to compare two luaIDs representing trades.</i>	



	bool = <b>IsSavingPossible()</b> <i>Checks whether saving is currently possible.</i>	
	IsSelectable	
	IsSliderCenterValueDisplayed	
	IsSliderIndicatorInverted	
	IsSliderScaleInverted	
	IsSofttargetLocked	
	bool = <b>IsSteamworksEnabled()</b> <i>Checks whether steamworks are enabled.</i>	
	IsTableColumnWidthPercentage	
	IsTableRowSelectable	
	IsType	
added in 3.50 Beta 2	bool = <b>IsValidComponent</b> (componentID) <i>Checks whether the specified component has a valid context.</i>	
	IsValidWidgetElement	
	bool = <b>IsWareIllegalTo</b> (ware, licencefaction [, policefaction]) <i>Check whether the specified ware is illegal for the given licencefaction. Optionally checks against a specified police faction.</i>	
deprecated in 4.21 Beta 1 (VR: 4.20)	<b>DEPRECATED</b> <b>LeaveInteractMenu()</b> <i>Ends the interact menu interaction.</i>	 <b>DEPRECATED</b> - use FFI function <code>LeaveInteractMenu()</code> instead
	<b>ListenForInput()</b> <i>Raises a lua event for the next input instead of evaluating that input. Event names are: "keyboardInput", "mousebuttonsInput", "joyaxesInputPosSgn[1-8]", "joyaxesInputNegSgn[1-8]", "joybuttonsInput[1-8]"</i>	
	<b>LoadGame</b> (filename) <i>Loads the specified savegame.</i>	
	<b>LoadInputProfile</b> (filename, personal) <i>Load the specified input profile.</i>	
	LockPresentation	
	<b>MakeRepairPriority</b> (entityID, componentID) <i>Move the specified component to the top of the repair queue of the entity.</i>	
	<b>MovePlayerToZone</b> (zoneID) <i>Teleport the player to the specified zone.</i>	Needs cheats enabled.
	<b>NewGame</b> (modulename, difficulty) <i>Start a new game with the specified module and difficulty.</i>	
	NotifyOnActionPerformed	
	NotifyOnChangedEnvironmentObject	
	NotifyOnChangedZone	
	NotifyOnConversationFinished	
	NotifyOnCutsceneStopped	
	NotifyOnCycleWeapon	
	NotifyOnHackAborted	
	NotifyOnHackFinished	

	NotifyOnHackStarted	
added in 4.00 Beta 2	NotifyOnInventoryWaresAdded	
added in 4.00 Beta 3	NotifyOnJumpDriveCharging	
	NotifyOnMissionObjectiveBarUpdate	
added in 4.31 Beta 1	NotifyOnObservationStarted	
added in 4.31 Beta 1	NotifyOnObservationStopped	
	NotifyOnPlayerControlStarted	
	NotifyOnPlayerControlStopped	
	NotifyOnScanAborted	
	NotifyOnScanFinished	
	NotifyOnScanStarted	
	NotifyOnStartDialog	
	NotifyOnStopDialog	
deprecated in 4.31 Beta 1	<b>DEPRECATED</b> NotifyOnTargetElementDisplayed	 <b>DEPRECATED</b> - use NotifyTargetElementShown() instead
deprecated in 4.31 Beta 1	<b>DEPRECATED</b> NotifyOnTargetElementHidden	 <b>DEPRECATED</b> - use NotifyTargetElementHidden() instead
added in 4.31 Beta 1	NotifyTargetElementHidden	
added in 4.31 Beta 1	NotifyTargetElementShown	
	NotifyVoiceOutput	
	<b>OpenSteamOverlayStorePage</b> ([appid]) <i>Opens the steam store page in the steam overlay.</i>	
	<b>OpenSteamOverlayWebPage</b> (url) <i>Opens an url in the steam overlay.</i>	
	<b>OpenWorkshop</b> (id, personal) <i>Open the specified extension in the workshop overlay.</i>	
changed in 4.00 Beta 3	<b>Pause</b> ([explicit]) <i>Pause the game.</i> If the explicit flag is set to true, this will pause the game as if the pause key was hit.	Note: prior to 4.21 Beta 1 (VR: 4.20) a call to Pause() (or Pause(false) didn't display a Game Paused text. As of 4.21 Beta 1 (VR: 4.20) both cases will display this text now.
	PerformAction	
	PlayAmbientSound	
	<b>PlayCredits</b> () <i>Play the credits.</i>	
	bool = <b>PlayerPrimaryShipHasContents</b> (key) <i>Checks whether the specified key is part of the player ship. Possible keys are: "trademk1", "trademk2", "trademk3"</i>	
	PlaySound	

added in 4.00 Beta 6	PrepareIcon	 <b>UNSUPPORTED.</b> Not designed to be used by mods.
	success = <b>PrepareRenderTarget</b> (texturename) <i>(Optionally) Prepares the specified texture to be used as a rendertarget. Return value indicates success/failure.</i>	 <b>UNSUPPORTED.</b> Not designed to be used by mods.
	<b>ProceedFromMenu</b> (convsection[, choiceparam][, baseparam]) <i>Proceeds from a conversation menu to the specified conversation section in MD. Optionally passing parameters.</i>	
added in 4.21 Beta 1 (VR: 4.20)	<b>QuickMenuOpened</b> () <i>Notifies the system that the quick menu was closed.</i>	
	<b>QuitGame</b> () <i>Quit to desktop.</i>	
	<b>QuitModule</b> () <i>Quit the current module to the start menu.</i>	
	RaisePlayerInteractionEvent	
	text = <b>ReadText</b> (page, line) <i>Returns text if it exists, error message otherwise.</i>	
	<b>RegisterAddonBindings</b> (addonname [, groupid = "") <i>Registers addon key bindings using the &lt;bindinggroup /&gt; nodes in ui.xml. If no groupid is specified all groups will be registered.</i>	 <b>UNSUPPORTED.</b> Not designed to be used by mods.
	<b>RegisterMouseInteractions</b> (element) <i>Registers the specified element (and all its children) to enable mouse/pointer picking interactions.</i>	
	RegisterWidget	
	ReleaseCutsceneDescriptor	
	ReleaseDescriptor	
depreacted in 4.31 Beta 1	<b>DEPRECATED ReleaseInteractionDescriptor</b> (descriptor) <i>Releases the descriptor which was returned by CreateInteractionDescriptor().</i>	 <b>DEPRECATED</b> - use FFI: ReleaseInteractionDescriptor() instead
	ReleaseListener	
	ReleaseNotification	
	ReleaseViewDescriptor	
	<b>RemoveAmmo</b> (defensibleID, ammoMacro, amount) <i>Removes the specified amount of the ammo from the object.</i>	
	<b>RemoveCargo</b> (containerID, ware, amount) <i>Removes the specified amount of the ware from the object.</i>	
	<b>RemoveInventory</b> (entityID, ware, amount) <i>Removes the specified amount of the ware from the inventory of the entity.</i>	
	<b>RemoveLogbookEntry</b> (index [, category]) <i>Removes the logbook entry at index.</i>	
	<b>RemovePlayerDrone</b> (macroname, amount) <i>Removes specified macro from the playership.</i>	
	RemoveSofttarget	

	<p>requesterRemoved = <b>RemoveSofttargetLockRequest</b>(requesterID)</p> <p><i>Removes the lock-request for the specified requesterID.</i></p> <p><i>Note that the return value only indicates whether the requester was removed. It does not necessarily correspond to the target lock having been lifted (i.e. if there are still outstanding lock-requests).</i></p>	
	<p><b>RepairDestructibles</b>(destructible1ID, destructible2ID, ...)</p> <p><i>Repairs the specified destructibles.</i></p>	
	<p><b>ReplacePlayerCockpit</b>(cockpitmacro)</p> <p><i>Replace the player ship cockpit with the specified macro.</i></p>	
updated in 4.21 Beta 1 (VR: 4.20)	<p>success = <b>RequestSofttargetLock</b>(requesterID)</p> <p><i>Adds a lock-request (with the specified requesterID) of the current softtarget (preventing the current target from being changed).</i></p> <p><i>Prior to 4.20 the function could be used to prevent objects being targeted (i.e. calling RequestSofttargetLock() with no current target). As of 4.20 such a call will fail and indicate the failure via its return value.</i></p>	4.21 Beta 1 (VR: 4.20): added success return value
	<p><b>ResetAllExtensionSettings</b>()</p> <p><i>Resets all changes to extension settings from this session.</i></p>	
	<p><b>RestoreGameOptions</b>()</p> <p><i>Restores the game settings from the default config.</i></p>	
	<p><b>RestoreGraphicOptions</b>()</p> <p><i>Restores the gfx settings from the default config.</i></p>	
	<p><b>RestoreOptionsMenu</b>()</p> <p><i>Opens the options menu.</i></p>	
	<p><b>RestoreSoundOptions</b>()</p> <p><i>Restores the sound settings from the default config.</i></p>	
	<p><b>ReturnFromMenu</b>([returnparam])</p> <p><i>Returns from a conversation menu. Optionally passing a parameter.</i></p>	
	<p><b>SaveFullscreenOption</b>()</p> <p><i>Save the current fullscreen setting to the player config.</i></p>	
	<p><b>SaveGame</b>(filename, savename)</p> <p><i>Save the game to the specified filename.</i></p>	
	<p><b>SaveInputProfile</b>(filename, id, customname[, copyfromfile])</p> <p><i>Save the input profile.</i></p>	

**SaveInputSettings**(actionTable, stateTable, rangeTable)  
Save the input map.

#### actionTable

> [Expand source](#)

```
actionTable = {
  [ActionID] = {
    [1] = {
      [1] = InputID Source,
      [2] = InputID Code,
      [3] = InputID Signum
    },
    [2] = { ... }
  },
  [ActionID] = { ... }
}
```

#### stateTable

> [Expand source](#)

```
stateTable = {
  [StateID] = {
    [1] = {
      [1] = InputID Source,
      [2] = InputID Code,
      [3] = InputID Signum
    },
    [2] = { ... }
  },
  [StateID] = { ... }
}
```

#### rangeTable

> [Expand source](#)

```
rangeTable = {
  [RangeID] = {
    [1] = {
      [1] = InputID Source,
      [2] = InputID Code,
      [3] = InputID Signum
    },
    [2] = { ... }
  },
  [RangeID] = { ... }
}
```

#### **SaveResolutionOption()**

Save the current resolutions setting to the player config.

#### **ScheduleReloadUI()**

Schedule a ui reload.


#### SelectAction




	<p><b>bool = SelectBackOption()</b>  <i>Select the back in a conversation choice.</i></p>	
	<p>SelectDialogOption</p>	
	<p><b>SendEntityTransportDrone</b>(entityID, containerID or nil)  <i>Sends the specified entity to the given container or the player ship via remote transport drone.</i></p>	
<p>added in 3.50  RC 1</p>	<p><b>SignalObject</b>(componentID [, param1] [, param2] [, param3] [, delay])  <i>Sends a generic ObjectSignalledEvent on the specified component. These can be listened to by MD/AI script.</i></p>	
	<p><b>SendTransportDrone</b>(sendingentityID, receivingentityID, waretype, wareAmountArray)  <i>Sends the specified wareAmountList from the sendingentity to the receivingentity.</i></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>wareAmountArray</b> <span style="float: right;">&gt; <a href="#">Expand source</a></span></p> <pre> wareAmountArray = {   [1] = {     [1] = ware,     [2] = amount   },   [2] = { ... } } </pre> </div>	
	<p><b>SetActiveMission</b>(missionID)  <i>Sets the specified mission as active.</i></p>	
	<p><b>SetAdapterOption</b>(adapterOrdinal)  <i>Sets the display adapter to the specified ordinal.</i></p>	
	<p><b>SetAimAssistOption</b>(setting)  <i>Sets the aim assist setting.</i></p>	
	<p><b>SetAntiAliasModeOption</b>(antialiasmode)  <i>Sets the anti alias mode.</i></p>	
	<p><b>SetAutorollOption</b>()  <i>Toggles the autoroll setting.</i></p>	
	<p><b>SetAutosaveOption</b>()  <i>Toggles autosave setting.</i></p>	
	<p><b>SetBoostToggleOption</b>()  <i>Toggle the boost toggle setting.</i></p>	
	<p>responsibility = <b>SetBudgetResponsibility</b>(entityID, responsibility)  <i>Set the budget responsibility of the specified entity.</i></p>	
<p>added in 3.60  Beta 1</p>	<p><b>SetButtonColor</b>(buttonID, r, g, b, alpha)  <i>Sets the color for the specified button.</i></p>	
	<p>SetButtonText</p>	
	<p><b>SetCaptureHQOption</b>()  Toggle the capture hq setting.</p>	
	<p>SetCellContent</p>	
	<p><b>SetCollisionAvoidanceAssistOption</b>()  Toggle the collision avoidance assist setting.</p>	
	<p><b>SetComponentName</b>(componentID, name)  Set the name of the specified component.</p>	

	<b>SetConfineMouseOption()</b> <i>Toggle confine mouse setting.</i>	
added in 4.00 Beta 3	<b>SetContainerStockLimitOverride</b> (containerID, ware, amount) <i>Define a stock limit for a particular ware for this ship/station.</i>	
	<b>SetContainerWarePriceOverride</b> (containerID or buildmoduleID, ware, buyssellswitch, price) <i>Sets the price the specified container buys/sells the ware.</i>	
	<b>SetCrashReportOption()</b> <i>Toggles the crash report setting.</i>	
	<b>SetDeadzoneOption</b> (percentage) <i>Set the deadzone setting to the specified percentage.</i>	
	<b>SetDifficultyOption</b> (setting) <i>Sets the difficulty setting.</i>	
	SetDiffuseColor	
	<b>SetDistortionOption()</b> <i>Toggles the distortion setting.</i>	
	<b>SetEffectDistanceOption</b> (distance) <i>Sets the effect distance setting.</i>	
	<b>DEPRECATED SetEMCutsceneOption()</b> <i>Toggle the event monitor cutscene setting.</i>	 <b>DEPRECATED</b> - do not use
	<b>SetExtensionSettings</b> (id, personal, setting, value) <i>Sets the setting of the specified extension.</i>	
	<b>SetFOVOption</b> (fov) <i>Sets the fov setting.</i>	
	<b>SetFullscreenDetailmonitorOption()</b> <i>Toggles the fullscreen detailmonitor setting.</i>	
updated in 4.10 Beta 1	bool = <b>SetFullscreenOneToOne()</b> <i>Sets the render mode of the current UI environment to fullscreen 1:1 mapping. Return value indicates success/failure.</i>	4.10 Beta 1: added boolean return value.
added in 4.21 Beta 1 (VR: 4.20)	<b>SetFullscreenWorldSpace()</b> <i>Sets the render mode of the current UI environment to worldspace rendering mode.</i>	
	<b>SetFullscreenOption</b> (setting, [checksetting]) <i>Sets the fullscreen setting and optionally provide a prompt if the setting worked.</i>	
	<b>SetGamepadModeOption</b> (setting) <i>Sets the gamepad mode option to setting.</i>	
	<b>SetGammaOption</b> (gamma) <i>Sets the gamma setting.</i>	
	<b>SetGfxQualityOption</b> (setting) <i>Sets the gfx quality setting.</i>	
	<b>SetGlowOption</b> (setting) <i>Sets the glow setting.</i>	
	SetHeight	
	SetIcon	
	<b>SetInversionSetting</b> (rangeid, configname) <i>Toggles inversion for the specified range input.</i>	
	<b>SetJoysticksOption</b> (joystick, guid) <i>Sets the specified guid as the joystick.</i>	

	<b>SetLegacyMainMenuOption()</b> <i>Toggles the legacy main menu setting.</i>	
	<b>SetLegacyShadersOption()</b> <i>Toggles the legacy shaders setting.</i>	
	<b>SetLODOption(lodfactor)</b> <i>Sets the lod setting.</i>	
	<b>SetMainMissiontargetMessage(posID, messageID)</b>	 <b>UNSUPPORTED.</b> Not designed to be used by mods.
	amount = <b>SetMaxBudget</b> (entityID, amount) <i>Sets the max budget of the specified entity.</i>	
	amount = <b>SetMinBudget</b> (entityID, amount) <i>Sets the min budget of the specified entity.</i>	
deprecated in 4.21 Beta 1 (VR: 4.20)	<b>DEPRECATED SetMouseOverride</b> (element, cursorIcon) <i>Sets the mouse cursor override of the specified element (and all of its children) to the given cursor ID. If the mouse is hovered over the element, the cursor will display the specified cursor icon.</i>	 <b>DEPRECATED</b> - use <b>SetPointerOverride</b> () instead (simply pass the cursorIcon as pointerID)
	<b>SetMouseSleeping()</b> <i>Sets the mouse to sleep until moved again.</i>	
	<b>SetNonSquadShipsForTradeOffersOption()</b> <i>Toggles the setting whether non-squad ships should be included in the trade offers menu.</i>	
changed in 4.10 Beta 5	<b>SetNPCBlackboard</b> (entityID, valueName, value) <i>Sets the blackboard variable "valueName" from the blackboard of the specified entity to "value". If the "value" is nil, the blackboard variable is cleared.</i>	4.10 Beta 5: If the "value" is nil, the blackboard variable is cleared.
	<b>SetNPCSkill</b> (entityID, skill, value) <i>Set the specified skill of the given entity to value.</i>	
	<b>SetPersonalizedCrashReportsOption()</b> <i>Toggles the setting to send personalized crash reports.</i>	
added in 4.21 Beta 1 (VR: 4.20)	<b>SetPointerOverride</b> (element, pointerID) <i>Sets the pointer ID override of the specified element (and all of its children) to the given pointer ID. If the pointer (i.e. either the cursor or a pointing device) is hovered over the element, the pointer will display the specified pointer ID).</i>	
added in 4.31 Beta 1	<b>SetPriorityMissionTargetMessage</b> (posID, messageID)	 <b>UNSUPPORTED.</b> Not designed to be used by mods.
	<b>SetRadarOption</b> (setting) <i>Sets the radar setting.</i>	
added in 4.00 Beta 1	<b>SetRenderTargetOneToOne()</b> <i>Swaps the rendering mode to a preset rendertarget. It's intended for UI internal use. No current support to utilize this in a useful way for UI modding.</i>	
	<b>SetRenderTargetSize</b> (texturename, width, height) <i>Sets the size of the specified rendertarget (given as a relative path to the texturename).</i>	 <b>UNSUPPORTED.</b> Not designed to be used by mods.
	<b>SetResolutionOption</b> (width, height [, checksetting]) <i>Sets the resolution to the specified values. Optionally provide a prompt if the setting worked.</i>	
	<b>SetRumbleOption</b> (setting) <i>Sets the rumble setting.</i>	
	<b>SetSensitivitySetting</b> (rangeid, configname, sensitivity) <i>Sets the sensitivity setting of the specified input range.</i>	
	<b>SetShaderQualityOption</b> (setting) <i>Sets the shader quality setting.</i>	



	<b>SetShadowOption</b> (setting) <i>Sets the shadow setting.</i>	
	<b>SetSoftShadowsOption</b> () <i>Toggles the soft shadow setting.</i>	
	SetSofttarget	
	<b>SetSoundOption</b> () <i>Toggle the global sound setting.</i>	
	<b>SetSSAOOption</b> (setting) <i>Sets the ssao setting.</i>	
	<b>SetStatValue</b> (stat, value) <i>Sets the specified statistic to value.</i>	
	<b>SetSteeringNoteOption</b> () <i>Toggles the steering note setting.</i>	
	<b>SetStopShipInMenuOption</b> () <i>Toggles the setting whether the playership should stop while a menu is open.</i>	
	<b>SetSubtitleOption</b> (setting) <i>Sets the subtitle setting.</i>	
	SetText	
	SetTextColor	
	SetTextureColorMode	
	<b>SetViewDistanceOption</b> (distance) <i>Sets the view distance setting.</i>	
	<b>SetVirtualCargoMode</b> (containerID, onoff) <i>Sets the specified container to virtual cargo mode. Using its trade queue to predict future cargo and unit availability.</i>	Virtual cargo mode should be disabled again when leaving the current menu.
	<b>SetVolumeOption</b> (volumetype, volume) <i>Sets the volume setting of the specified type.</i>	
	<b>SetVSyncOption</b> () <i>Toggles the vsync setting.</i>	
	SetWidth	
deprecated in 4.21 Beta 1 (VR: 4.20)	<b>DEPRECATED ShowInteractMenu</b> () <i>Starts the interact menu interaction.</i>	 <b>DEPRECATED</b> - use FFI function EnterInteractMenu() instead
	ShowMissionObjective	
	ShowPresentation	
	<b>StartAutoPilot</b> (componentID) <i>Start the auto pilot on the specified component.</i>	
	<b>StartBriefing</b> (missionid, {texturename1, ...}) <i>Notifies the MD that a mission briefing was started.</i>	
added in 4.21 Beta 1 (VR: 4.20)	success = <b>StartConversationFromMenu</b> (conversation, actor, convparam) <i>Starts the specified new conversation in MD. This requires that no conversation is currently active. Conversation parameter is required but can be nil.</i>	
	StartCutscene	
	StartPlayingSound	
	success = <b>StartSubConversationFromMenu</b> (conversation, actor[, convparam][, baseparam][, view]) <i>Proceeds from a conversation menu to the specified new conversation in MD. Optionally passing parameters.</i>	

	<b>StopAutoPilot()</b> Stops the auto pilot.	
	<b>StopBriefing</b> (missionid) <i>Notify the MD that a mission briefing stopped.</i>	
	StopCutscene	
	StopPlayingSound	
	<b>StopRestoringOptionsMenu()</b> <i>Stop restoring the options menu after ui reloads.</i>	
	SwitchInteractiveObject	
	<b>ToggleFactionTradeRestriction</b> (containerID) Toggle faction trade restriction on the specified container.	
	<b>ToggleFactionTradeWareOverride</b> (containerID, ware) Toggle faction trade restriction override on the specified container for the given ware.	
	<b>ToggleMissionGuidance</b> (missionid) <i>Toggle the mission guidance for the specified mission.</i>	
	TraceBack	
	transferredAmount = <b>TransferInventoryToPlayer</b> (ware, amount, entityID) <i>Transfers the specified ware and amount from the given entity to the player.</i>	
	transferredMoney = <b>TransferMoneyToPlayer</b> (amount, entityID) <i>Transfers the specified amount from the given entity to the player.</i>	
	transferredAmount = <b>TransferPlayerInventoryTo</b> (ware, amount, entityID) <i>Transfers the specified ware and amount from the player to the given entity.</i>	
	transferredMoney = <b>TransferPlayerMoneyTo</b> (amount, entityID) <i>Transfers the specified amount from the player to the given entity.</i>	
	text = <b>TruncateText</b> (text, fontname, fontsize, width [, wordwrap, textheight]) <i>Truncates the given text with the specified limitations.</i>	
	<b>UninstallSteamDLC</b> (appid) <i>Uninstalls the specified steam dlc.</i>	
changed in 4.00 Beta 3	<b>Unpause</b> ([explicit]) <i>Unpause the game. If the explicit flag is set to true, this will unpause the game as if the pause key was hit.</i>	
	<b>UnregisterAddonBindings</b> (addonname, [, groupid = ""]) <i>Unregisters addon key bindings using the &lt;bindinggroup /&gt; nodes in ui.xml. If no groupid is specified all groups will be unregistered.</i>	 <b>UNSUPPORTED.</b> Not designed to be used by mods.
deprecated in 4.31 Beta 1	<b>DEPRECATED UnsetMapRenderTarget()</b> <i>Resets the render target for the radar/map display.</i>	 <b>DEPRECATED</b> - use FFI function UnsetRadarRenderTarget() instead
deprecated in 4.21 Beta 1 (VR: 4.20)	<b>DEPRECATED UnsetMouseOverride</b> (element) <i>Removes any specified mouse cursor override from the specified element or its children.</i>	 <b>DEPRECATED</b> - use UnsetPointerOverride() instead
added in 4.21 Beta 1 (VR: 4.20)	<b>UnsetPointerOverride</b> (element) <i>Removes any specified pointer override from the specified element or its children.</i>	
added in 4.31 Beta 1	<b>UnregisterMouseInteractions</b> (element) <i>Unregisters the specified element (and all its children) from mouse/pointer picking interactions.</i>	
	UnsuspendConversation	
	UpdateRegisteredWidget	

<b>UpgradeObject</b> (destructibleID, upgradetype, level) <i>Upgrade the specified destructible with the given upgrade level.</i>	
--	--

// @since 4.31 Beta 1: deprecated (use NotifyTargetElementShown)