Razor Enhanced

RE Team

Jun 17, 2021

CONTENTS

CHAPTER

ONE

PYTHON API

1.1 AutoLoot

1.1.1 Methods

`AutoLoot.`**`ChangeList`**`(`*listName*`)` → Void

- listName: `String` Name of an existing organizer list.

Change the Autoloot's active list.

`AutoLoot.`**`GetList`**`(`*lootListName*`)` → List[*AutoLoot.AutoLootItem*]

- lootListName: `String` Name of the AutoLoot list.

Given an AutoLoot list name, return a list of AutoLootItem associated.

`AutoLoot.`**`GetLootBag`**`()` → UInt32

Get current Autoloot destination container.

`AutoLoot.`**`ResetIgnore`**`()` → Void

Reset the Autoloot ignore list.

`AutoLoot.`**`RunOnce`**`(`*lootListName*, *millisec*, *filter*`)` → Void

- lootListName: `String` Name of the Autoloot listfilter for search on ground.
- millisec: `Int32` Delay in milliseconds. (unused)
- filter: *`Items.Filter`* Item filter

Start Autoloot with custom parameters.

`AutoLoot.`**`SetNoOpenCorpse`**`(`*noOpen*`)` → Boolean

- noOpen: `Boolean` True: "No Open Corpse" is active - False: otherwise

Toggle "No Open Corpse" on/off. The change doesn't persist if you reopen razor.

`AutoLoot.`**`Start`**`()` → Void

Start the Autoloot Agent on the currently active list.

`AutoLoot.`**`Status`**`()` → Boolean

Check Autoloot Agent status

`AutoLoot.`**`Stop`**`()` → Void

Stop the Autoloot Agent.

1.2 AutoLoot.AutoLootItem

1.2.1 Properties

- AutoLoot.AutoLootItem.Color `Int32`
- AutoLoot.AutoLootItem.Graphics `Int32`
- AutoLoot.AutoLootItem.List `String`
- AutoLoot.AutoLootItem.LootBagOverride `Int32`
- AutoLoot.AutoLootItem.Name `String`
- AutoLoot.AutoLootItem.Properties `List[AutoLoot.AutoLootItem.Property]`
- AutoLoot.AutoLootItem.Selected `Boolean`

1.3 BandageHeal

1.3.1 Methods

`BandageHeal.`**`Start`**`()` → Void

Start BandageHeal Agent.

`BandageHeal.`**`Status`**`()` → Boolean

Check BandageHeal Agent status, returns a bool value.

`BandageHeal.`**`Stop`**`()` → Void

Stop BandageHeal Agent.

1.4 BuyAgent

1.4.1 Methods

`BuyAgent.`**`ChangeList`**`(`*listName*`)` → Void

- listName: `String` Name of an existing buy list.

Change the BuyAgent's active list.

`BuyAgent.`**`Disable`**`()` → Void

Disable BuyAgent Agent.

`BuyAgent.`**`Enable`**`()` → Void

Enable BuyAgent on the currently active list.

`BuyAgent.`**`Status`**`()` → Boolean

Check BuyAgent Agent status

1.5 DPSMeter

1.5.1 Methods

`DPSMeter.`**`GetDamage`**`(`*serial*`)` → Int32

- serial: `Int32` Serial of the Mobile.

Get total damage per Mobile.

`DPSMeter.`**`Pause`**`()` → Void

Pause DPSMeter data recording.

`DPSMeter.`**`Start`**`()` → Void

Start DPSMeter engine.

`DPSMeter.`**`Status`**`()` → Boolean

Check DPSMeter Agent status, returns a bool value.

`DPSMeter.`**`Stop`**`()` → Void

Stop DPSMeter engine.

1.6 Dress

1.6.1 Methods

`Dress.`**`ChangeList`**`(`*dresslist*`)` → Void

- dresslist: `String` Name of the list of friend.

Change dress list, List must be exist in dress/undress Agent tab.

`Dress.`**`DressFStart`**`()` → Void

Start Dress engine.

`Dress.`**`DressFStop`**`()` → Void

Stop Dress engine.

`Dress.`**`DressStatus`**`()` → Boolean

Check Dress Agent status, returns a bool value.

`Dress.`**`UnDressFStart`**`()` → Void

Start UnDress engine.

`Dress.`**`UnDressFStop`**`()` → Void

Stop UnDress engine.

`Dress.`**`UnDressStatus`**`()` → Boolean

Check UnDress Agent status, returns a bool value.

1.7 Friend

1.7.1 Methods

`Friend.`**`AddFriendTarget`**`()` → Void

`Friend.`**`AddPlayer`**`(`*friendlist*, *name*, *serial*`)` → Void

- friendlist: `String` Name of the the Friend List. (See Agent tab)
- name: `String` Name of the Friend want to add.
- serial: `Int32` Serial of the Friend you want to add.

Add the player specified to the Friend list named in FriendListName parameter

`Friend.`**`ChangeList`**`(`*friendlist*`)` → Void

- friendlist: `String` Name of the list of friend.

Change friend list, List must be exist in friend list GUI configuration

`Friend.`**`GetList`**`(`*friendlist*`)` → List[Int32]

- friendlist: `String` Name of the list of friend.

Retrive list of serial in list, List must be exist in friend Agent tab.

`Friend.`**`IsFriend`**`(`*serial*`)` → Boolean

- serial: `Int32` Serial you want to check

Check if Player is in FriendList, returns a bool value.

1.8 Gumps

1.8.1 Methods

`Gumps.`**`CloseGump`**`(`*gumpid*`)` → Void

- gumpid: `UInt32` ID of the gump

Close a specific Gump.

`Gumps.`**`CurrentGump`**`()` → UInt32

Return the ID of most recent, still open Gump.

`Gumps.`**`HasGump`**`()` → Boolean

Get status if have a gump open or not.

`Gumps.`**`LastGumpGetLine`**`(`*line_num*`)` → String

- line_num: `Int32` Number of the line.

Get a specific line from the most recent and still open Gump. Filter by line number.

`Gumps.`**`LastGumpGetLineList`**`()` → List[String]

Get all text from the most recent and still open Gump.

`Gumps.`**`LastGumpRawData`**`()` → String

Get the Raw Data of the most recent and still open Gump.

`Gumps.`**`LastGumpRawText`**`()` → List[String]

Get the Raw Text of the most recent and still open Gump.

`Gumps.`**`LastGumpTextExist`**`(`*text*`)` → Boolean

- text: `String` Text to search.

Search for text inside the most recent and still open Gump.

`Gumps.`**`LastGumpTextExistByLine`**`(`*line_num*, *text*`)` → Boolean

- line_num: `Int32` Number of the line.
- text: `String` Text to search.

Search for text, in a spacific line of the most recent and still open Gump.

`Gumps.`**`LastGumpTile`**`()` → List[Int32]

Get the list of Gump Tile (! this documentation is a stub !)

`Gumps.`**`ResetGump`**`()` → Void

Clean current status of Gumps.

`Gumps.`**`SendAction`**`(`*gumpid*, *buttonid*`)` → Void

- gumpid: `UInt32` ID of the gump.
- buttonid: `Int32` ID of the Button to press.

Send a Gump response by gumpid and buttonid.

`Gumps.`**`SendAdvancedAction`**`(`*gumpid*, *buttonid*, *switchlist_id*, *textlist_id*, *textlist_str*`)` → Void

- gumpid: `UInt32`
- buttonid: `Int32`
- switchlist_id: `List[Int32]`
- textlist_id: `List[Int32]`
- textlist_str: `List[String]`

`Gumps.`**`SendAdvancedAction`**`(`*gumpid*, *buttonid*, *textlist_id*, *textlist_str*`)` → Void

- gumpid: `UInt32`
- buttonid: `Int32`
- textlist_id: `List[Int32]`
- textlist_str: `List[String]`

`Gumps.`**`SendAdvancedAction`**`(`*gumpid*, *buttonid*, *switchs*`)` → Void

- gumpid: `UInt32`
- buttonid: `Int32`
- switchs: `List[Int32]`

`Gumps.`**`WaitForGump`**`(`*gumpid*, *delay*`)` → Void

- gumpid: `UInt32` ID of the gump. (0: any)
- delay: `Int32` Maximum wait, in milliseconds.

Waits for a specific Gump to appear, for a maximum amount of time. If gumpid is 0 it will match any Gump.

1.9 HotKeyEvent

1.9.1 Properties

- HotKeyEvent.HotKey `Keys`
- HotKeyEvent.LastEvent *`HotKeyEvent`*
- HotKeyEvent.Timestamp `Double`

1.9.2 Methods

`HotKeyEvent.`**`AddEvent`**`(`*`k`*`)` → *HotKeyEvent*

- k: `Keys`

1.10 Item

1.10.1 Properties

- Item.Amount `Int32` Read amount from item type object.
- Item.Container `Int32` Serial of the container which contains the object.
- Item.Contains `List[Item]` Contains the list of Item inside a container.
- Item.Deleted `Boolean`
- Item.Direction `String` Item direction.
- Item.Durability `Int32` Get the current durability of an Item. (0: no durability)
- Item.GridNum `Byte` Returns the GridNum of the item. (need better documentation)
- Item.Hue `Int32`
- Item.Image `Bitmap` Get the in-game image on an Item as Bitmap object.

See MSDN: https://docs.microsoft.com/dotnet/api/system.drawing.bitmap

- Item.IsBagOfSending `Boolean` True: if the item is a bag of sending - False: otherwise.
- Item.IsContainer `Boolean` True: if the item is a container - False: otherwise.
- Item.IsCorpse `Boolean` True: if the item is a corpse - False: otherwise.
- Item.IsDoor `Boolean` True: if the item is a door - False: otherwise.
- Item.IsInBank `Boolean` True: if the item is in the Player's bank - False: otherwise.
- Item.IsLootable `Boolean` True: For regualar items - False: for hair, beards, etc.
- Item.IsPotion `Boolean` True: if the item is a potion - False: otherwise.
- Item.IsPouch `Boolean` True: if the item is a pouch - False: otherwise.
- Item.IsResource `Boolean` True: if the item is a resource (ore, sand, wood, stone, fish) - False: otherwise

- Item.IsTwoHanded `Boolean` True: if the item is a 2-handed weapon - False: otherwise.
- Item.IsVirtueShield `Boolean` True: if the item is a virtue shield - False: otherwise.
- Item.ItemID `Int32` Represents the type of Item, usually unique for the Item image. Sometime called ID or Graphics ID.
- Item.Layer `String` Gets the Layer, for werable items only. (need better documentation)
- Item.MaxDurability `Int32` Get the maximum durability of an Item. (0: no durability)
- Item.Movable `Boolean` Item is movable
- Item.Name `String` Item name
- Item.OnGround `Boolean` True: if the item is on the ground - False: otherwise.
- Item.Position *Point3D*
- Item.Properties `List[Property]` Get the list of Properties of an Item.
- Item.PropsUpdated `Boolean` True: if Properties are updated - False: otherwise.
- Item.RootContainer `Int32` Get serial of root container of item.
- Item.Serial `Int32`
- Item.Updated `Boolean` Check if the Item already have been updated with all the properties. (need better documentation)
- Item.Visible `Boolean` Item is Visible
- Item.Weight `Int32` Get the weight of a item. (0: no weight)

1.10.2 Methods

`Item.`**`DistanceTo`**`(`*itm*`)` → Int32

- itm: *Item* Target as Item

`Item.`**`DistanceTo`**`(`*mob*`)` → Int32

- mob: *Mobile* Target as Mobile

Return the distance in number of tiles, from Item to Mobile.

`Item.`**`GetWorldPosition`**`()` → *Point3D*

`Item.`**`IsChildOf`**`(`*container*`)` → Boolean

- container: *Item* Item as container.

Check if an Item is contained in a container. Can be a Item or a Mobile (wear by).

`Item.`**`IsChildOf`**`(`*container*`)` → Boolean

- container: *Mobile* Mobile as container.

`Item.`**`ToString`**`()` → String

1.11 Items

1.11.1 Methods

`Items.`**`ApplyFilter`**`(`*filter*`)` → List[*Item*]

- filter: *Items.Filter* A filter object.

Filter the global list of Items according to the options specified by the filter (see: Items.Filter).

`Items.`**`BackpackCount`**`(`*itemid*, *color*`)` → Int32

- itemid: `Int32` ItemID to search.
- color: `Int32` Color to search. (default -1: any color)

Count items in Player Backpack.

`Items.`**`ContainerCount`**`(`*serial*, *itemid*, *color*, *recursive*`)` → Int32

- serial: `Int32`
- itemid: `Int32`
- color: `Int32`
- recursive: `Boolean`

`Items.`**`ContainerCount`**`(`*container*, *itemid*, *color*, *recursive*`)` → Int32

- container: *Item* Serial or Item to search into.
- itemid: `Int32` ItemID of the item to search.
- color: `Int32` Color to match. (default: -1, any color)
- recursive: `Boolean` Search also in already open subcontainers.

Count items inside a container, summing also the amount in stacks.

`Items.`**`ContextExist`**`(`*i*, *name*`)` → Int32

- i: *Item*
- name: `String`

`Items.`**`ContextExist`**`(`*serial*, *name*`)` → Int32

- serial: `Int32` Serial or Item to check.
- name: `String` Name of the Context Manu entry

Check if Context Menu entry exists for an Item.

`Items.`**`DropFromHand`**`(`*item*, *container*`)` → Void

- item: *Item* Item object to drop.
- container: *Item* Target container.

Drop into a bag an Item currently held in-hand. (see: Items.Lift)

`Items.`**`DropItemGroundSelf`**`(`*serialitem*, *amount*`)` → Void

- serialitem: `Int32`
- amount: `Int32`

`Items.`**`DropItemGroundSelf`**`(`*item*, *amount*`)` → Void

- item: *Item* Item object to drop.
- amount: Int32 Amount to move. (default: 0, the whole stack)

Drop an Item on the ground, at the current Player position. NOTE: On some server is not allowed to drop Items on tiles occupied by Mobiles and the Player.

Items.**FindByID**(*itemid*, *color*, *container*, *range*, *considerIgnoreList*) → *Item*

- itemid: Int32
- color: Int32
- container: Int32
- range: Int32
- considerIgnoreList: Boolean

Items.**FindByID**(*itemid*, *color*, *container*, *recursive*, *considerIgnoreList*) → *Item*

- itemid: Int32 ItemID filter.
- color: Int32 Color filter. (-1: any, 0: natural)
- container: Int32 Serial of the container to search. (-1: any Item)
- **recursive: Boolean Search subcontainers.** True: all subcontainers False: only main 1,2,n: Maximum subcontainer depth
- considerIgnoreList: Boolean True: Ignore Items are excluded - False: any Item.

Find a single Item matching specific ItemID, Color and Container. Optionally can search in all subcontaners or to a maximum depth in subcontainers. Can use -1 on color for no chose color, can use -1 on container for search in all item in memory. The depth defaults to only the top but can search for # of sub containers.

Items.**FindBySerial**(*serial*) → *Item*

- serial: Int32 Serial of the Item.

Search for a specific Item by using it Serial

Items.**GetImage**(*itemID*, *hue*) → Bitmap

- itemID: Int32 ItemID to use.
- hue: Int32 Optional: Color to apply. (Default 0, natural)

Get the Image on an Item by specifing the ItemID. Optinally is possible to apply a color.

Items.**GetPropStringByIndex**(*serial*, *index*) → String

- serial: Int32 Serial or Item to read.
- index: Int32 Number of the Property line.

Get a Property line, by index. if not found returns and empty string.

Items.**GetPropStringByIndex**(*item*, *index*) → String

- item: *Item*
- index: Int32

Items.**GetPropStringList**(*serial*) → List[String]

- serial: Int32 Serial or Item to read.

Get string list of all Properties of an item, if item no props list is empty.

Items.**GetPropStringList**(*item*) → List[String]

- item: *Item*

Items.**GetPropValue**(*serial*, *name*) → Single

- serial: Int32 Serial or Item to read.
- name: String Name of the Propery.

Read the value of a Property.

Items.**GetPropValue**(*item*, *name*) → Single

- item: *Item*
- name: String

Items.**Hide**(*item*) → Void

- item: *Item*

Items.**Hide**(*serial*) → Void

- serial: Int32 Serial or Item to hide.

Hied an Item, affects only the player.

Items.**Lift**(*item*, *amount*) → Void

- item: *Item* Item object to Lift.
- amount: Int32 Amount to lift. (0: the whole stack)

Lift an Item and hold it in-hand. (see: Items.DropFromHand)

Items.**Message**(*serial*, *hue*, *message*) → Void

- serial: Int32
- hue: Int32
- message: String

Items.**Message**(*item*, *hue*, *message*) → Void

- item: *Item* Serial or Item to display text on.
- hue: Int32 Color of the message.
- message: String Message as

Display an in-game message on top of an Item, visibile only for the Player.

Items.**Move**(*source*, *destination*, *amount*) → Void

- source: Int32
- destination: *Item*
- amount: Int32

Items.**Move**(*source*, *destination*, *amount*, *x*, *y*) → Void

- source: *Item*
- destination: *Mobile*
- amount: Int32
- x: Int32

- y: Int32

Items.**Move**(*source*, *destination*, *amount*) → Void

- source: Int32
- destination: Int32
- amount: Int32

Items.**Move**(*source*, *destination*, *amount*) → Void

- source: *Item*
- destination: *Item*
- amount: Int32

Items.**Move**(*source*, *destination*, *amount*, *x*, *y*) → Void

- source: Int32
- destination: *Mobile*
- amount: Int32
- x: Int32
- y: Int32

Items.**Move**(*source*, *destination*, *amount*) → Void

- source: *Item*
- destination: Int32
- amount: Int32

Items.**Move**(*source*, *destination*, *amount*) → Void

- source: Int32
- destination: *Mobile*
- amount: Int32

Items.**Move**(*source*, *destination*, *amount*, *x*, *y*) → Void

- source: Int32 Serial or Item of the Item to move.
- destination: Int32 Serial, Mobile or Item as destination.
- amount: Int32 Amount to move (-1: the whole stack)
- x: Int32 Optional: X coordinate inside the container.
- y: Int32 Optional: Y coordinate inside the container.

Move an Item to a destination, which can be an Item or a Mobile.

Items.**Move**(*source*, *destination*, *amount*, *x*, *y*) → Void

- source: Int32
- destination: *Item*
- amount: Int32
- x: Int32
- y: Int32

Items.**Move**(*source*, *destination*, *amount*) → Void

- source: *Item*
- destination: *Mobile*
- amount: Int32

Items.**Move**(*source*, *destination*, *amount*, *x*, *y*) → Void

- source: *Item*
- destination: *Item*
- amount: Int32
- x: Int32
- y: Int32

Items.**Move**(*source*, *destination*, *amount*, *x*, *y*) → Void

- source: *Item*
- destination: Int32
- amount: Int32
- x: Int32
- y: Int32

Items.**MoveOnGround**(*source*, *amount*, *x*, *y*, *z*) → Void

- source: Int32 Serial or Item to move.
- amount: Int32 Amount of Items to move (0: the whole stack)
- x: Int32 X world coordinates.
- y: Int32 Y world coordinates.
- z: Int32 Z world coordinates.

Move an Item on the ground to a specific location.

Items.**MoveOnGround**(*source*, *amount*, *x*, *y*, *z*) → Void

- source: *Item*
- amount: Int32
- x: Int32
- y: Int32
- z: Int32

Items.**Select**(*items*, *selector*) → *Item*

- items: List[Item]
- selector: String

Items.**SetColor**(*serial*, *color*) → Void

- serial: Int32 Serial of the Item.
- color: Int32 Color as number. (default: -1, reset original color)

Change/override the Color of an Item, the change affects only Player client. The change is not persistent. If the color is -1 or unspecified, the color of the item is restored.

`Items.`**`SingleClick`**(*itemserial*) → Void

- itemserial: `Int32`

`Items.`**`SingleClick`**(*item*) → Void

- item: *Item* Serial or Item to click

Send a single click network event to the server.

`Items.`**`UseItem`**(*item*, *target*) → Void

- item: *Item*
- target: `Int32`

`Items.`**`UseItem`**(*item*, *target*) → Void

- item: *Item*
- target: `EnhancedEntity`

`Items.`**`UseItem`**(*item*) → Void

- item: *Item*

`Items.`**`UseItem`**(*itemSerial*, *targetSerial*) → Void

- itemSerial: `Int32`
- targetSerial: `Int32`

`Items.`**`UseItem`**(*item*, *target*) → Void

- item: `Int32`
- target: `EnhancedEntity`

`Items.`**`UseItem`**(*itemSerial*, *targetSerial*, *wait*) → Void

- itemSerial: `Int32` Serial or Item to use.
- targetSerial: `Int32` Optional: Serial of the Item or Mobile target.
- wait: `Boolean` Optional: Wait for confirmation by the server. (default: True)

Use an Item, optionally is possible to specify a Item or Mobile target. NOTE: The optional target may not work on some free shards. Use Target.Execute instead.

`Items.`**`UseItem`**(*itemserial*) → Void

- itemserial: `Int32`

`Items.`**`UseItemByID`**(*itemid*, *color*) → Boolean

- itemid: `Int32` ItemID to be used.
- color: `Int32` Color to be used. (default: -1, any)

Use any item of a specific type, matching Item.ItemID. Optionally also of a specific color, matching Item.Hue.

`Items.`**`WaitForContents`**(*bag*, *delay*) → Void

- bag: *Item* Container as Item object.
- delay: `Int32` Maximum wait, in milliseconds.

Open a container an wait for the Items to load, for a maximum amount of time.

`Items.`**`WaitForContents`**(*bag_serial*, *delay*) → Void

- bag_serial: `Int32` Container as Item serial.
- delay: `Int32`

`Items.`**`WaitForProps`**(*i*, *delay*) → Void

- i: *Item*
- delay: `Int32`

`Items.`**`WaitForProps`**(*itemserial*, *delay*) → Void

- itemserial: `Int32` Serial or Item read.
- delay: `Int32` Maximum waiting time, in milliseconds.

If not updated, request to the Properties of an Item, and wait for a maximum amount of time.

1.12 Items.Filter

1.12.1 Properties

- Items.Filter.CheckIgnoreObject `Boolean` Exclude from the search Items which are currently on the global Ignore List. (default: False, any Item)
- Items.Filter.Enabled `Boolean` True: The filter is used - False: Return all Item. (default: True, active)
- Items.Filter.Graphics `List[Int32]` Limit the search to a list of Grapichs ID (see: Item.ItemID)

Supports .Add() and .AddRange()

- Items.Filter.Hues `List[Int32]` Limit the search to a list of Colors.

Supports .Add() and .AddRange()

- Items.Filter.IsContainer `Int32` Limit the search to the Items which are also containers. (default: -1: any Item)
- Items.Filter.IsCorpse `Int32` Limit the search to the corpses on the ground. (default: -1, any Item)
- Items.Filter.IsDoor `Int32` Limit the search to the doors. (default: -1: any Item)
- Items.Filter.Layers `List[String]` Limit the search to the wearable Items by Layer.

Supports .Add() and .AddRange()

Layers: RightHand LeftHand Shoes Pants Shirt Head Gloves Ring Neck Waist InnerTorso Bracelet MiddleTorso Earrings Arms Cloak OuterTorso OuterLegs InnerLegs Talisman

- Items.Filter.Movable `Int32` Limit the search to only Movable Items. (default: -1, any Item)
- Items.Filter.Name `String` Limit the search by name of the Item.
- Items.Filter.OnGround `Int32` Limit the search to the Items on the ground. (default: -1, any Item)
- Items.Filter.RangeMax `Double` Limit the search by distance, to Items on the ground which are at most RangeMax tiles away from the Player. (default: -1, any Item)
- Items.Filter.RangeMin `Double` Limit the search by distance, to Items on the ground which are at least RangeMin tiles away from the Player. (default: -1, any Item)
- Items.Filter.Serials `List[Int32]` Limit the search to a list of Serials of Item to find. (ex: 0x0406EFCA)

Supports .Add() and .AddRange()

1.13 Journal

1.13.1 Methods

`Journal.`**`Clear`**`()` → Void

Removes all entry from the Jorunal. This operation is Global for all Razor.

`Journal.`**`GetJournalEntry`**`(`*afterTimestap*`)` → List[*Journal.JournalEntry*]

- afterTimestap: `Double` Timestap as UnixTime, the number of seconds elapsed since 01-Jan-1970. (default: -1, no filter)

Get a copy of all Journal lines as JournalEntry. The list can be filtered to include *only* most recent events.

`Journal.`**`GetJournalEntry`**`(`*afterJournalEntry*`)` → List[*Journal.JournalEntry*]

- afterJournalEntry: *`Journal.JournalEntry`* A JournalEntry object (default: null, no filter)

Get a copy of all Journal lines as JournalEntry. The list can be filtered to include *only* most recent events.

`Journal.`**`GetLineText`**`(`*text*, *addname*`)` → String

- text: `String` Text to search.
- addname: `Boolean` Prepend source name. (default: False)

Search and return the most recent line Journal containing the given text. (case sensitive)

`Journal.`**`GetSpeechName`**`()` → List[String]

Get list of speakers.

`Journal.`**`GetTextByColor`**`(`*color*, *addname*`)` → List[String]

- color: `Int32` Color of the soruce.
- addname: `Boolean` Prepend source name. (default: False)

Returns all the lines present in the Journal for a given color.

`Journal.`**`GetTextByName`**`(`*name*, *addname*`)` → List[String]

- name: `String` Name of the soruce.
- addname: `Boolean` Prepend source name. (default: False)

Returns all the lines present in the Journal for a given source name. (case sensitive)

`Journal.`**`GetTextBySerial`**`(`*serial*, *addname*`)` → List[String]

- serial: `Int32` Serial of the soruce.
- addname: `Boolean` Prepend source name. (default: False)

Returns all the lines present in the Journal for a given serial.

`Journal.`**`GetTextByType`**`(`*type*, *addname*`)` → List[String]

- type: `String` Regular

System Emote Label Focus Whisper Yell Spell Guild Alliance Party Encoded Special * addname: `Boolean` Prepend source name. (default: False)

Returns all the lines present in the Journal for a given type. (case sensitive)

`Journal.`**`Search`**`(`*text*`)` → Boolean

- text: `String` Text to search.

Search in the Journal for the occurrence of text. (case sensitive)

`Journal.`**`SearchByColor`**(*text*, *color*) → Boolean

- text: `String` Text to search.
- color: `Int32` Color of the message.

Search in the Journal for the occurrence of text, for a given color. (case sensitive)

`Journal.`**`SearchByName`**(*text*, *name*) → Boolean

- text: `String` Text to search.
- name: `String` Name of the source.

Search in the Journal for the occurrence of text, for a given soruce. (case sensitive)

`Journal.`**`SearchByType`**(*text*, *type*) → Boolean

- text: `String` Text to search.
- type: `String` Regular

System Emote Label Focus Whisper Yell Spell Guild Alliance Party Encoded Special

Search in the Journal for the occurrence of text, for a given type. (case sensitive)

`Journal.`**`WaitByName`**(*name*, *delay*) → Boolean

- name: `String` Name of the soruce.
- delay: `Int32` Maximum pause in milliseconds.

Pause script and wait for maximum amount of time, for a specific soruce to appear in Jorunal. (case sensitive)

`Journal.`**`WaitJournal`**(*msgs*, *delay*) → String

- msgs: `List[String]`
- delay: `Int32`

`Journal.`**`WaitJournal`**(*text*, *delay*) → Boolean

- text: `String` Text to search.
- delay: `Int32` Maximum pause in milliseconds.

Pause script and wait for maximum amount of time, for a specific text to appear in Journal. (case sensitive)

1.14 Journal.JournalEntry

1.14.1 Properties

- Journal.JournalEntry.Color `Int32` Color of the text.
- Journal.JournalEntry.Name `String` Name of the source, can be a Mobile or an Item.
- Journal.JournalEntry.Serial `Int32` Name of the source, can be a Mobile or an Item.
- Journal.JournalEntry.Text `String` Actual content of the Journal Line.
- Journal.JournalEntry.Timestamp `Double` Timestamp as UnixTime, the number of seconds elapsed since 01-Jan-1970.

- Journal.JournalEntry.Type `String` Regular

System Emote Label Focus Whisper Yell Spell Guild Alliance Party Encoded Special

1.14.2 Methods

`Journal.JournalEntry.`**`Copy`**`()` → *Journal.JournalEntry*

1.15 `Misc`

1.15.1 Methods

`Misc.`**`Beep`**`()` → Void

Play Beep system sound.

`Misc.`**`CancelPrompt`**`()` → Void

Cancel a prompt request.

`Misc.`**`CaptureNow`**`()` → Void

Creates a snapshot of the current UO window.

`Misc.`**`CheckIgnoreObject`**`(`*serial*`)` → Boolean

- serial: `Int32` Serial to check.

Check object from ignore list, return true if present. Can check Serial, Items or Mobiles

`Misc.`**`CheckIgnoreObject`**`(`*itm*`)` → Boolean

- itm: *`Item`* Item to check

`Misc.`**`CheckIgnoreObject`**`(`*mob*`)` → Boolean

- mob: *`Mobile`* Mobile to check

`Misc.`**`CheckSharedValue`**`(`*name*`)` → Boolean

- name: `String` Name of the value.

Check if a shared value exixts.

`Misc.`**`ClearDragQueue`**`()` → Void

Clear the Drag-n-Drop queue.

`Misc.`**`ClearIgnore`**`()` → Void

Clear ignore list from all object

`Misc.`**`CloseBackpack`**`()` → Void

Close the backpack. (OSI client only, no ClassicUO)

`Misc.`**`CloseMenu`**`()` → Void

Close opened Old Menu.

`Misc.`**`ContextReply`**`(`*serial*, *menu_name*`)` → Void

- serial: `Int32`

- menu_name: `String` Name of the Entry as wirtten in-game.

`Misc.`**`ContextReply`**(*serial*, *respone_num*) → Void

- serial: `Int32` Serial of the Entity
- respone_num: `Int32` Poition of the option in the menu. Starts from 0.

Respond to a context menu on mobile or item. Menu ID is base zero, or can use string of menu text.

`Misc.`**`ContextReply`**(*mob*, *menu_num*) → Void

- mob: *Mobile*
- menu_num: `Int32`

`Misc.`**`ContextReply`**(*mob*, *menu_name*) → Void

- mob: *Mobile*
- menu_name: `String`

`Misc.`**`ContextReply`**(*itm*, *menu_name*) → Void

- itm: *Item*
- menu_name: `String`

`Misc.`**`ContextReply`**(*itm*, *menu_num*) → Void

- itm: *Item*
- menu_num: `Int32`

`Misc.`**`CurrentScriptDirectory`**() → String

Get the path to the Scripts directory.

`Misc.`**`Disconnect`**() → Void

Force client to disconnect.

`Misc.`**`DistanceSqrt`**(*point_a*, *point_b*) → Double

- point_a: *Point3D* First coordinates.
- point_b: *Point3D* Second coordinates.

Compute the distance between 2 Point3D using pitagora's.

`Misc.`**`ExportPythonAPI`**(*path*, *pretty*) → Void

- path: `String` Name of the output file. (default: Config/AutoComplete.json)
- pretty: `Boolean` Print a readable JSON. (default: True)

Return a string containing list RE Python API list in JSON format.

`Misc.`**`FilterSeason`**(*enable*, *seasonFlag*) → Void

- enable: `Boolean` True: enable seasons filter
- seasonFlag: `UInt32` 0: Spring (default fallback)

1: Summer 2: Fall 3: Winter 4: Desolation

Enable or disable the Seasons filter forcing a specific season Season filter state will be saved on logout but not the season flag that will be recovered.

`Misc.`**`FocusUOWindow`**() → Void

Set UoClient window in focus or restore if minimized.

`Misc.`**`GetContPosition`**`()` → Point

Get the position of the currently active Gump/Container. (OSI client only, no ClassicUO)

`Misc.`**`GetMapInfo`**`(`*serial*`)` → *Misc.MapInfo*

- serial: `UInt32` Serial of the object.

Get MapInfo about a Mobile or Item using the serial

`Misc.`**`GetMenuTitle`**`()` → String

Get the title of title for open Old Menu.

`Misc.`**`HasMenu`**`()` → Boolean

Check if an Old Menu is open.

`Misc.`**`HasPrompt`**`()` → Boolean

Check if have a prompt request.

`Misc.`**`HasQueryString`**`()` → Boolean

Check if a have a query string menu opened, return true or false.

`Misc.`**`IgnoreObject`**`(`*serial*`)` → Void

- serial: `Int32` Serial to ignore.

Add an entiry to the ignore list. Can ignore Serial, Items or Mobiles.

`Misc.`**`IgnoreObject`**`(`*mob*`)` → Void

- mob: *`Mobile`* Mobile to ignore

`Misc.`**`IgnoreObject`**`(`*itm*`)` → Void

- itm: *`Item`* Item to ignore

`Misc.`**`Inspect`**`()` → Void

Prompt the user with a Target. Open the inspector for the selected target.

`Misc.`**`LastHotKey`**`()` → *HotKeyEvent*

Returns the latest HotKey recorded by razor as HotKeyEvent object.

`Misc.`**`MenuContain`**`(`*text*`)` → Boolean

- text: `String` Text to search.

Search in open Old Menu if contains a specific text.

`Misc.`**`MenuResponse`**`(`*text*`)` → Void

- text: `String` Name of subitem to respond.

Perform a menu response by subitem name. If item not exist close menu.

`Misc.`**`MouseLocation`**`()` → Point

Returns a point with the X and Y coordinates of the mouse relative to the UO Window

`Misc.`**`MouseMove`**`(`*posX*, *posY*`)` → Void

- posX: `Int32` X screen coordinate.
- posY: `Int32` Y screen coordinate.

Moves the mouse pointer to the position X,Y relative to the UO window

`Misc.NextContPosition(x, y) → Void`

- x: `Int32` X coordinate.
- y: `Int32` Y coordinate.

Return the X,Y of the next container, relative to the game window. (OSI client only, no ClassicUO)

`Misc.NoOperation() → Void`

Just do nothing and enjot the present moment.

`Misc.NoRunStealthStatus() → Boolean`

Get the status of "No Run When Stealth" via scripting.

`Misc.NoRunStealthToggle(enable) → Void`

- enable: `Boolean` True: enable the option.

Set "No Run When Stealth" via scripting. Changes via scripting are not persistents.

`Misc.OpenPaperdoll() → Void`

Open the backpack. (OSI client only, no ClassicUO)

`Misc.Pause(millisec) → Void`

- millisec: `Int32` Pause duration, in milliseconds.

Pause the script for a given amount of time.

`Misc.PetRename(mob, name) → Void`

- mob: *Mobile* Mobile object representing the pet.
- name: `String`

`Misc.PetRename(serial, name) → Void`

- serial: `Int32` Serial of the pet.
- name: `String` New name to set.

Rename a specific pet.

`Misc.QueryStringResponse(okcancel, response) → Void`

- okcancel: `Boolean` OK Button
- response: `String` Cancel Button

Perform a query string response by ok or cancel button and specific response string.

`Misc.ReadSharedValue(name) → Object`

- name: `String` Name of the value.

Get a Shared Value, if value not exist return null. Shared values are accessible by every script.

`Misc.RemoveSharedValue(name) → Void`

- name: `String` Name of the value.

Remove a Shared Value.

`Misc.ResetPrompt() → Void`

Reset a prompt response.

`Misc.`**`ResponsePrompt`**`(`*text*`)` → Void

- text: `String` Text of the response.

Response a prompt request. Often used to rename runes and similar.

`Misc.`**`Resync`**`()` → Void

Trigger a client ReSync.

`Misc.`**`ScriptRun`**`(`*scriptfile*`)` → Void

- scriptfile: `String` Name of the script.

Run a script by file name, Script must be present in script grid.

`Misc.`**`ScriptStatus`**`(`*scriptfile*`)` → Boolean

- scriptfile: `String`

Get status of script if running or not, Script must be present in script grid.

`Misc.`**`ScriptStop`**`(`*scriptfile*`)` → Void

- scriptfile: `String` Name of the script.

Stop a script by file name, Script must be present in script grid.

`Misc.`**`ScriptStopAll`**`()` → Void

Stop all script running.

`Misc.`**`SendMessage`**`(`*msg*, *wait*`)` → Void

- msg: `String`
- wait: `Boolean`

`Misc.`**`SendMessage`**`(`*msg*, *color*, *wait*`)` → Void

- msg: `String` The object to print.
- color: `Int32` Color of the message.
- wait: `Boolean` True: Wait for confimation. - False: Returns instatnly.

Send a message to the client.

`Misc.`**`SendMessage`**`(`*obj*, *color*`)` → Void

- obj: `Object`
- color: `Int32`

`Misc.`**`SendMessage`**`(`*num*`)` → Void

- num: `Int32`

`Misc.`**`SendMessage`**`(`*obj*`)` → Void

- obj: `Object`

`Misc.`**`SendMessage`**`(`*num*`)` → Void

- num: `UInt32`

`Misc.`**`SendMessage`**`(`*msg*`)` → Void

- msg: `Double`

`Misc.`**`SendMessage`**`(`*msg*, *color*`)` → Void

- msg: Double
- color: Int32

Misc.**SendMessage**(*msg*) → Void

- msg: Boolean

Misc.**SendMessage**(*num*) → Void

- num: Single

Misc.**SendMessage**(*num*, *color*) → Void

- num: Int32
- color: Int32

Misc.**SendMessage**(*num*, *color*) → Void

- num: UInt32
- color: Int32

Misc.**SendMessage**(*msg*, *color*) → Void

- msg: Boolean
- color: Int32

Misc.**SendToClient**(*keys*) → Void

- keys: String List of keys.

Send to the client a list of keystrokes. Can contain control characters: - Send Control+Key: ctrl+u: ^u - Send ENTER: {Enter} Note: some keys don't work with ClassicUO (es: {Enter})

Misc.**SetSharedValue**(*name*, *value*) → Void

- name: String Name of the value.
- value: Object Value to set.

Set a Shared Value by specific name, if value exist he repalce value. Shared values are accessible by every script.

Misc.**ShardName**() → String

Get the name of the shard.

Misc.**UnIgnoreObject**(*mob*) → Void

- mob: *Mobile* Item to unignore

Misc.**UnIgnoreObject**(*itm*) → Void

- itm: *Item* Item to unignore.

Misc.**UnIgnoreObject**(*serial*) → Void

- serial: Int32 Serial to unignore.

Remove object from ignore list. Can remove serial, items or mobiles

Misc.**WaitForContext**(*serial*, *delay*) → List[*Misc.Context*]

- serial: Int32 Serial of the entity.
- delay: Int32 Maximum wait.

Wait for Context Menu to appear, for a maximum amount of time. Usable on an Item or Mobile.

Misc.**WaitForContext**(*itm*, *delay*) → List[*Misc.Context*]

- itm: *Item* Entity as Item object.
- delay: Int32

Misc.**WaitForContext**(*mob*, *delay*) → List[*Misc.Context*]

- mob: *Mobile* Entity as Item object.
- delay: Int32

Misc.**WaitForMenu**(*delay*) → Boolean

- delay: Int32 Maximum wait, in milliseconds.

Pause script until server send an Old Menu, for a maximum amount of time.

Misc.**WaitForPrompt**(*delay*) → Boolean

- delay: Int32 Maximum wait time.

Wait for a prompt for a maximum amount of time.

Misc.**WaitForQueryString**(*delay*) → Boolean

- delay: Int32 Maximum wait, in milliseconds.

Pause script until server send query string request, for a maximum amount of time.

1.16 Misc.Context

1.16.1 Properties

- Misc.Context.Entry String
- Misc.Context.Response Int32

1.17 Misc.MapInfo

1.17.1 Properties

- Misc.MapInfo.Facet UInt16
- Misc.MapInfo.MapEnd *Point2D*
- Misc.MapInfo.MapOrigin *Point2D*
- Misc.MapInfo.PinPosition *Point2D*
- Misc.MapInfo.Serial UInt32

1.18 Mobile

1.18.1 Properties

- Mobile.Backpack *Item* Get the Item representing the backpack of a Mobile. Return null if it doesn't have one.
- Mobile.Body `Int32` Represents the type of Mobile, usually unique for the Mobile image. (Alias: Mobile.MobileID)
- Mobile.Color `Int32` Color of the mobile.
- Mobile.Contains `List[Item]` Returns the list of items present in the Paperdoll (or equivalent) of the Mobile.

Might not match the items found using via Layer.

- Mobile.Deleted `Boolean`
- Mobile.Direction `String` Returns the direction of the Mobile.
- Mobile.Female `Boolean` The Mobile is a female.
- Mobile.Flying `Boolean` The mobile is Flying (Gragoyle)
- Mobile.Hits `Int32` The current hit point of a Mobile. To be read as propotion over Mobile.HitsMax.
- Mobile.HitsMax `Int32` Maximum hitpoint of a Mobile.
- Mobile.Hue `Int32`
- Mobile.InParty `Boolean` True: if the Mobile is in your party. - False: otherwise.
- Mobile.IsGhost `Boolean` If is a Ghost

Match any MobileID in the list: 402, 403, 607, 608, 694, 695, 970

- Mobile.IsHuman `Boolean` Check is the Mobile has a human body.

Match any MobileID in the list: 183, 184, 185, 186, 400, 401, 402, 403, 605, 606, 607, 608, 666, 667, 694, 744, 745, 747, 748, 750, 751, 970, 695

- Mobile.Mana `Int32` The current mana of a Mobile. To be read as propotion over Mobile.ManaMax.
- Mobile.ManaMax `Int32` Maximum mana of a Mobile.
- Mobile.Map `Int32` Current map or facet.
- Mobile.MobileID `Int32` Represents the type of Mobile, usually unique for the Mobile image. (Alias: Mobile.Body)
- Mobile.Mount *Item* Returns the Item assigned to the "Mount" Layer.
- Mobile.Name `String` Name of the Mobile.
- Mobile.Notoriety `Int32` Get the notoriety of the Mobile.

Notorieties: 1: blue, innocent 2: green, friend 3: gray, neutral 4: gray, criminal 5: orange, enemy 6: red, hostile 6: yellow, invulnerable

- Mobile.Paralized `Boolean` The mobile is Paralized.
- Mobile.Poisoned `Boolean` The mobile is Poisoned.
- Mobile.Position *Point3D*
- Mobile.Properties `List[Property]` Get all properties of a Mobile as list of lines of the tooltip.
- Mobile.PropsUpdated `Boolean` True: Mobile.Propertires are updated - False: otherwise.

- Mobile.Quiver *Item* Get the Item representing the quiver of a Mobile. Return null if it doesn't have one.
- Mobile.Serial Int32
- Mobile.Stam Int32 The current stamina of a Mobile. To be read as propotion over Mobile.StamMax.
- Mobile.StamMax Int32 Maximum stamina of a Mobile.
- Mobile.Visible Boolean True: The Mobile is visible - Flase: The mobile is hidden.
- Mobile.WarMode Boolean Mobile is in War mode.
- Mobile.YellowHits Boolean The mobile healthbar is not blue, but yellow.

1.18.2 Methods

Mobile.**DistanceTo**(*other_mobile*) → Int32

- other_mobile: *Mobile* The other mobile.

Returns the distance between the current Mobile and another one.

Mobile.**GetItemOnLayer**(*layer*) → *Item*

- **layer: String Layers:** Layername RightHand LeftHand Shoes Pants Shirt Head Gloves Ring Neck Waist InnerTorso Bracelet MiddleTorso Earrings Arms Cloak OuterTorso OuterLegs InnerLegs

Returns the Item associated with a Mobile Layer.

1.19 Mobiles

1.19.1 Methods

Mobiles.**ApplyFilter**(*filter*) → List[*Mobile*]

- filter: *Mobiles.Filter*

Mobiles.**ContextExist**(*serial*, *name*) → Int32

- serial: Int32
- name: String

Mobiles.**ContextExist**(*mob*, *name*) → Int32

- mob: *Mobile*
- name: String

Mobiles.**FindBySerial**(*serial*) → *Mobile*

- serial: Int32 Serial of the Mobile.

Find the Mobile with a specific Serial.

Mobiles.**GetPropStringByIndex**(*serial*, *index*) → String

- serial: Int32
- index: Int32

Mobiles.**GetPropStringByIndex**(*mob*, *index*) → String

- mob: *Mobile*

- index: Int32

Mobiles.**GetPropStringList**(*mob*) → List[String]

- mob: *Mobile*

Mobiles.**GetPropStringList**(*serial*) → List[String]

- serial: Int32

Mobiles.**GetPropValue**(*mob*, *name*) → Single

- mob: *Mobile*
- name: String

Mobiles.**GetPropValue**(*serial*, *name*) → Single

- serial: Int32
- name: String

Mobiles.**GetTrackingInfo**() → *Mobiles.TrackingInfo*

Get the most updated information about tracking.

Mobiles.**Message**(*mobile*, *hue*, *message*, *wait*) → Void

- mobile: *Mobile*
- hue: Int32
- message: String
- wait: Boolean

Mobiles.**Message**(*serial*, *hue*, *message*, *wait*) → Void

- serial: Int32
- hue: Int32
- message: String
- wait: Boolean

Mobiles.**Select**(*mobiles*, *selector*) → *Mobile*

- mobiles: List[Mobile]
- selector: String

Mobiles.**SingleClick**(*mobileserial*) → Void

- mobileserial: Int32

Mobiles.**SingleClick**(*mobile*) → Void

- mobile: *Mobile*

Mobiles.**UseMobile**(*mobileserial*) → Void

- mobileserial: Int32

Mobiles.**UseMobile**(*mobile*) → Void

- mobile: *Mobile*

Mobiles.**WaitForProps**(*m*, *delay*) → Void

- m: *Mobile*

- delay: `Int32`

`Mobiles.`**`WaitForProps`**`(`*mobileserial*, *delay*`)` → Void

- mobileserial: `Int32`
- delay: `Int32`

`Mobiles.`**`WaitForStats`**`(`*mobileserial*, *delay*`)` → Void

- mobileserial: `Int32`
- delay: `Int32`

`Mobiles.`**`WaitForStats`**`(`*m*, *delay*`)` → Void

- m: *`Mobile`*
- delay: `Int32`

1.20 `Mobiles.Filter`

1.20.1 Properties

- Mobiles.Filter.Blessed `Int32` Limit the search to only Blessed Mobiles. (default: -1, any Mobile)
- Mobiles.Filter.Bodies `List[Int32]` Limit the search to a list of MobileID (see: Mobile.ItemID or Mobile.Body)

Supports .Add() and .AddRange()

- Mobiles.Filter.CheckIgnoreObject `Boolean` Exclude from the search Mobiles which are currently on the global Ignore List. (default: False, any Item)
- Mobiles.Filter.CheckLineOfSight `Boolean` Limit the search only to the Mobiles which are in line of sight. (default: false, any Mobile)
- Mobiles.Filter.Enabled `Boolean` True: The filter is used - False: Return all Mobile. (default: True, active)
- Mobiles.Filter.Female `Int32` Limit the search to female Mobile. (default: -1, any)
- Mobiles.Filter.Friend `Int32` Limit the search to friend Mobile. (default: -1, any)
- Mobiles.Filter.Hues `List[Int32]` Limit the search to a list of Colors.

Supports .Add() and .AddRange()

- Mobiles.Filter.IsGhost `Int32` Limit the search to Ghost only. (default: -1, any Mobile)

Match any MobileID in the list: 402, 403, 607, 608, 694, 695, 970

- Mobiles.Filter.IsHuman `Int32` Limit the search to Humans only. (default: -1, any Mobile)

Match any MobileID in the list: 183, 184, 185, 186, 400, 401, 402, 403, 605, 606, 607, 608, 666, 667, 694, 744, 745, 747, 748, 750, 751, 970, 695

- Mobiles.Filter.Name `String` Limit the search by name of the Mobile.
- Mobiles.Filter.Notorieties `List[Byte]` Limit the search to the Mobile by notoriety.

Supports .Add() and .AddRange()

Notorieties: 1: blue, innocent 2: green, friend 3: gray, neutral 4: gray, criminal 5: orange, enemy 6: red, hostile 6: yellow, invulnerable

- Mobiles.Filter.Paralized `Int32` Limit the search to paralized Mobile. (default: -1, any)
- Mobiles.Filter.Poisoned `Int32` Limit the search to only Poisoned Mobiles. (default: -1, any Mobile)
- Mobiles.Filter.RangeMax `Double` Limit the search by distance, to Mobiles which are at most RangeMax tiles away from the Player. (default: -1, any Mobile)
- Mobiles.Filter.RangeMin `Double` Limit the search by distance, to Mobiles which are at least RangeMin tiles away from the Player. (default: -1, any Mobile)
- Mobiles.Filter.Serials `List[Int32]` Limit the search to a list of Serials of Mobile to find. (ex: 0x0406EFCA)

Supports .Add() and .AddRange()

- Mobiles.Filter.Warmode `Int32` Limit the search to Mobile War mode. (default: -1, any Mobile)

  -1: any 0: peace 1: war

1.21 Mobiles.TrackingInfo

1.21.1 Properties

- Mobiles.TrackingInfo.lastUpdate `DateTime`
- Mobiles.TrackingInfo.serial `UInt32`
- Mobiles.TrackingInfo.x `UInt16`
- Mobiles.TrackingInfo.y `UInt16`

1.22 Organizer

1.22.1 Methods

`Organizer.`**`ChangeList`**`(`*listName*`)` → Void

- listName: `String` Name of an existing organizer list.

Change the Organizer's active list.

`Organizer.`**`FStart`**`()` → Void

Start the Organizer Agent on the currently active list.

`Organizer.`**`FStop`**`()` → Void

Stop the Organizer Agent.

`Organizer.`**`RunOnce`**`(`*organizerName*, *sourceBag*, *destBag*, *dragDelay*`)` → Void

- organizerName: `String`
- sourceBag: `Int32`
- destBag: `Int32`
- dragDelay: `Int32`

`Organizer.`**`Status`**`()` → Boolean

Check Organizer Agent status

1.23 PathFinding

1.23.1 Methods

PathFinding.**GetPath**(*dst_x*, *dst_y*, *ignoremob*) → List[*Tile*]

- dst_x: Int32 Destination X.
- dst_y: Int32 Destination Y.
- ignoremob: Boolean Ignores any mobiles with the path calculation.

Compute the path for the given destination and returns a list of Tile (coordinates).

PathFinding.**Go**(*r*) → Boolean

- r: *PathFinding.Route* A customized Route object.

Check if a destination is reachable.

PathFinding.**RunPath**(*path*, *timeout*, *debugMessage*, *useResync*) → Boolean

- path: List[Tile]
- timeout: Single
- debugMessage: Boolean
- useResync: Boolean

PathFinding.**Tile**(*x*, *y*) → *Tile*

- x: Int32 X coordinate
- y: Int32 Y coordinate

Create a Tile starting from X,Y coordinates (see PathFindig.RunPath)

1.24 PathFinding.Route

1.24.1 Properties

- PathFinding.Route.DebugMessage Boolean Outputs a debug message. (default: False)
- PathFinding.Route.IgnoreMobile Boolean Ignores any mobiles with the path calculation. (default: 0)
- PathFinding.Route.MaxRetry Int32 Number of attempts untill the path calculation is halted. (default: -1, no limit)
- PathFinding.Route.StopIfStuck Boolean Halts the pathfinding fail to walk the path. (default: 0)
- PathFinding.Route.Timeout Single Maximum amount of time to run the path. (default: -1, no limit)
- PathFinding.Route.UseResync Boolean ReSyncs the path calculation. (default: False)
- PathFinding.Route.X Int32 Sets the destination position X. (default: 0)
- PathFinding.Route.Y Int32 Sets the destination position Y. (default: 0)

1.25 Player

1.25.1 Properties

- Player.AR `Int32` Resistance to Phisical damage.
- Player.Backpack *`Item`* Player backpack, as Item object.
- Player.Bank *`Item`* Player bank chest, as Item object.
- Player.Body `Int32` Player Body or MobileID (see: Mobile.Body)
- Player.Buffs `List[String]` List of Player active buffs:

  Meditation Agility Animal Form Arcane Enpowerment Arcane Enpowerment (new) Arch Protection Armor Pierce Attunement Aura of Nausea Bleed Bless Block Bload Oath (caster) Bload Oath (curse) BloodWorm Anemia City Trade Deal Clumsy Confidence Corpse Skin Counter Attack Criminal Cunning Curse Curse Weapon Death Strike Defense Mastery Despair Despair (target) Disarm (new) Disguised Dismount Prevention Divine Fury Dragon Slasher Fear Enchant Enemy Of One Enemy Of One (new) Essence Of Wind Ethereal Voyage Evasion Evil Omen Faction Loss Fan Dancer Fan Fire Feeble Mind Feint Force Arrow Berserk Fly Gaze Despair Gift Of Life Gift Of Renewal Healing Heat Of Battle Hiding Hiryu Physical Malus Hit Dual Wield Hit Lower Attack Hit Lower Defense Honorable Execution Honored Horrific Beast Hawl Of Cacophony Immolating Weapon Incognito Inspire Invigorate Invisibility Lich Form Lighting Strike Magic Fish Magic Reflection Mana Phase Mass Curse Medusa Stone Mind Rot Momentum Strike Mortal Strike Night Sight NoRearm Orange Petals Pain Spike Paralyze Perfection Perseverance Poison Poison Resistance Polymorph Protection Psychic Attack Consecrate Weapon Rage Rage Focusing Rage Focusing (target) Reactive Armor Reaper Form Resilience Rose Of Trinsic Rotworm Blood Disease Rune Beetle Corruption Skill Use Delay Sleep Spell Focusing Spell Focusing (target) Spell Plague Splintering Effect Stone Form Strangle Strength Surge Swing Speed Talon Strike Vampiric Embrace Weaken Wraith Form
- Player.ColdResistance `Int32` Resistance to Cold damage.
- Player.DamageChanceIncrease `Int32` Get total Damage Chance Increase.
- Player.DefenseChanceIncrease `Int32` Get total Defense Chance Increase.
- Player.Dex `Int32` Stats value for Dexterity.
- Player.DexterityIncrease `Int32` Get total Dexterity Increase.
- Player.Direction `String` Player current direction, as text.
- Player.EnergyResistance `Int32` Resistance to Energy damage.
- Player.EnhancePotions `Int32` Get total Enhance Potions.
- Player.FasterCasting `Int32` Get total Faster Casting.
- Player.FasterCastRecovery `Int32` Get total Faster Cast Recovery.
- Player.Female `Boolean` Player is a female.
- Player.FireResistance `Int32` Resistance to Fire damage.
- Player.Followers `Int32` Player current amount of pet/followers.
- Player.FollowersMax `Int32` Player maximum amount of pet/followers.
- Player.Gold `Int32` Player total gold, in the backpack.
- Player.HasSpecial `Boolean` Player have a special abilities active.
- Player.HitPointsIncrease `Int32` Get total Hit Points Increase.

- Player.HitPointsRegeneration `Int32` Get total Hit Points Regeneration.
- Player.Hits `Int32` Current hit points.
- Player.HitsMax `Int32` Maximum hit points.
- Player.InParty `Boolean` Player is in praty.
- Player.Int `Int32` Stats value for Intelligence.
- Player.IntelligenceIncrease `Int32` Get total Intelligence Increase.
- Player.IsGhost `Boolean` Player is a Ghost
- Player.LowerManaCost `Int32` Get total Lower Mana Cost.
- Player.LowerReagentCost `Int32` Get total Lower Reagent Cost.
- Player.Luck `Int32` Player total luck.
- Player.Mana `Int32` Current mana.
- Player.ManaIncrease `Int32` Get total Mana Increase.
- Player.ManaMax `Int32` Maximum mana.
- Player.ManaRegeneration `Int32` Get total Mana Regeneration.
- Player.Map `Int32` Player current map, or facet.
- Player.MaximumHitPointsIncrease `Int32` Get total Maximum Hit Points Increase.
- Player.MaximumManaIncrease `Int32` Get total Maximum Mana Increase.
- Player.MaximumStaminaIncrease `Int32` Get total Maximum Stamina Increase.
- Player.MaxWeight `Int32` Player maximum weight.
- Player.MobileID `Int32` Player MobileID or Body (see: Mobile.MobileID)
- Player.Mount *Item* Player current Mount, as Item object.

NOTE: On some server the Serial return by this function doesn't match the mount serial.

- Player.Name `String` Player name.
- Player.Notoriety `Byte` Player notoriety

  1: blue, innocent 2: green, friend 3: gray, neutral 4: gray, criminal 5: orange, enemy 6: red, hostile 6: yellow, invulnerable

- Player.Paralized `Boolean` Player is Paralized. True also while frozen because of casting of spells.
- Player.Poisoned `Boolean` Player is Poisoned
- Player.PoisonResistance `Int32` Resistance to Poison damage.
- Player.Position *Point3D* Current Player position as Point3D object.
- Player.Quiver *Item* Player quiver, as Item object.
- Player.ReflectPhysicalDamage `Int32` Get total Reflect Physical Damage.
- Player.Serial `Int32` Player unique Serial.
- Player.SpellDamageIncrease `Int32` Get total Spell Damage Increase.
- Player.Stam `Int32` Current stamina.
- Player.StaminaIncrease `Int32` Get total Stamina Increase.

- Player.StaminaRegeneration `Int32` Get total Stamina Regeneration.
- Player.StamMax `Int32` Maximum stamina.
- Player.StatCap `Int32` Get the stats cap.
- Player.StaticMount `Int32` Retrieves serial of mount set in Filter/Mount GUI.
- Player.Str `Int32` Stats value for Strenght.
- Player.StrengthIncrease `Int32` Get total Strength Increase.
- Player.SwingSpeedIncrease `Int32` Get total Swing Speed Increase.
- Player.Visible `Boolean` Player is visible, false if hidden.
- Player.WarMode `Boolean` Player has war mode active.
- Player.Weight `Int32` Player current weight.
- Player.YellowHits `Boolean` Player HP bar is not blue, but yellow.

1.25.2 Methods

`Player.`**`Area`**`()` → String

Get the name of the area in which the Player is currently in. (Ex: Britain, Destard, Vesper, Moongate, etc) Regions are defined inside by Config/regions.json.

`Player.`**`Attack`**`(`*serial*`)` → Void

- serial: `Int32` Serial or Mobile to attack.

Attack a Mobile.

`Player.`**`Attack`**`(`*mobile*`)` → Void

- mobile: *Mobile*

`Player.`**`AttackLast`**`()` → Void

Attack last target.

`Player.`**`BuffsExist`**`(`*buffname*`)` → Boolean

- buffname: `String` Meditation

Agility Animal Form Arcane Enpowerment Arcane Enpowerment (new) Arch Protection Armor Pierce Attunement Aura of Nausea Bleed Bless Block Bload Oath (caster) Bload Oath (curse) BloodWorm Anemia City Trade Deal Clumsy Confidence Corpse Skin Counter Attack Criminal Cunning Curse Curse Weapon Death Strike Defense Mastery Despair Despair (target) Disarm (new) Disguised Dismount Prevention Divine Fury Dragon Slasher Fear Enchant Enemy Of One Enemy Of One (new) Essence Of Wind Ethereal Voyage Evasion Evil Omen Faction Loss Fan Dancer Fan Fire Feeble Mind Feint Force Arrow Berserk Fly Gaze Despair Gift Of Life Gift Of Renewal Healing Heat Of Battle Hiding Hiryu Physical Malus Hit Dual Wield Hit Lower Attack Hit Lower Defense Honorable Execution Honored Horrific Beast Hawl Of Cacophony Immolating Weapon Incognito Inspire Invigorate Invisibility Lich Form Lighting Strike Magic Fish Magic Reflection Mana Phase Mass Curse Medusa Stone Mind Rot Momentum Strike Mortal Strike Night Sight NoRearm Orange Petals Pain Spike Paralyze Perfection Perseverance Poison Poison Resistance Polymorph Protection Psychic Attack Consecrate Weapon Rage Rage Focusing Rage Focusing (target) Reactive Armor Reaper Form Resilience Rose Of Trinsic Rotworm Blood Disease Rune Beetle Corruption Skill Use Delay Sleep Spell Focusing Spell Focusing (target) Spell Plague Splintering Effect Stone Form Strangle Strength Surge Swing Speed Talon Strike Vampiric Embrace Weaken Wraith Form

Check if a buff is active, by buff name.

Player.**ChatAlliance**(*msg*) → Void

- msg: Int32

Player.**ChatAlliance**(*msg*) → Void

- msg: String Message to send.

Send message to the alliace chat.

Player.**ChatChannel**(*msg*) → Void

- msg: Int32

Player.**ChatChannel**(*msg*) → Void

- msg: String Message to send.

Send an chat channel message.

Player.**ChatEmote**(*color*, *msg*) → Void

- color: Int32 Color of the text
- msg: String Message to send.

Send an emote in game.

Player.**ChatEmote**(*color*, *msg*) → Void

- color: Int32
- msg: Int32

Player.**ChatGuild**(*msg*) → Void

- msg: Int32

Player.**ChatGuild**(*msg*) → Void

- msg: String Message to send.

Send message to the guild chat.

Player.**ChatParty**(*msg*, *recepient_serial*) → Void

- msg: String Text to send.
- recepient_serial: Int32 Optional: Target of private message.

Send message in arty chat. If a recepient_serial is specified, the message is private.

Player.**ChatSay**(*color*, *msg*) → Void

- color: Int32 Color of the text
- msg: String Message to send.

Send message in game.

Player.**ChatSay**(*color*, *msg*) → Void

- color: Int32
- msg: Int32

Player.**ChatWhisper**(*color*, *msg*) → Void

- color: Int32 Color of the text
- msg: String Message to send.

Send an wishper message.

`Player.ChatWhisper(color, msg) → Void`

- color: `Int32`
- msg: `Int32`

`Player.ChatYell(color, msg) → Void`

- color: `Int32`
- msg: `Int32`

`Player.ChatYell(color, msg) → Void`

- color: `Int32` Color of the text
- msg: `String` Message to send.

Send an yell message.

`Player.CheckLayer(layer) → Boolean`

- **layer: `String` Layers:** RightHand LeftHand Shoes Pants Shirt Head Gloves Ring Neck Hair Waist InnerTorso Bracelet FacialHair MiddleTorso Earrings Arms Cloak OuterTorso OuterLegs InnerLegs Talisman

Check if a Layer is equipped by the Item.

`Player.DistanceTo(target) → Int32`

- target: *`Mobile`* The other Mobile or Item

Returns the distance between the Player and a Mobile or an Item.

`Player.DistanceTo(target) → Int32`

- target: *`Item`*

`Player.EquipItem(item) → Void`

- item: *`Item`*

`Player.EquipItem(serial) → Void`

- serial: `Int32` Serial or Item to equip.

Equip an Item

`Player.EquipUO3D(serials) → Void`

- serials: `List[Int32]`

`Player.Fly(status) → Void`

- status: `Boolean` True: Gargoyle Fly ON - False: Gargoyle fly OFF

Enable or disable Gargoyle Flying.

`Player.GetItemOnLayer(layer) → Item`

- **layer: `String` Layers:** RightHand LeftHand Shoes Pants Shirt Head Gloves Ring Neck Hair Waist InnerTorso Bracelet FacialHair MiddleTorso Earrings Arms Cloak OuterTorso OuterLegs InnerLegs Talisman

Returns the Item associated with a Mobile Layer.

`Player.GetPropStringByIndex(index) → String`

- index: `Int32` Line number, start from 0.

Get a single line of Properties of the Player, from the tooltip, as text.

`Player.GetPropStringList()` → List[String]

Get the list of Properties of the Player, as list of lines of the tooltip.

`Player.GetPropValue(`*name*`)` → Int32

- name: `String` Name of the property.

Get the numeric value of a specific Player property, from the tooltip.

`Player.GetRealSkillValue(`*skillname*`)` → Double

- skillname: `String` Alchemy

Anatomy Animal Lore Item ID Arms Lore Parry Begging Blacksmith Fletching Peacemaking Camping Carpentry Cartography Cooking Detect Hidden Discordance EvalInt Healing Fishing Forensics Herding Hiding Provocation Inscribe Lockpicking Magery Magic Resist Mysticism Tactics Snooping Musicianship Poisoning Archery Spirit Speak Stealing Tailoring Animal Taming Taste ID Tinkering Tracking Veterinary Swords Macing Fencing Wrestling Lumberjacking Mining Meditation Stealth Remove Trap Necromancy Focus Chivalry Bushido Ninjitsu Spell Weaving Imbuing

Get the base/real value of the skill for the given the skill name.

`Player.GetSkillCap(`*skillname*`)` → Double

- skillname: `String` Alchemy

Anatomy Animal Lore Item ID Arms Lore Parry Begging Blacksmith Fletching Peacemaking Camping Carpentry Cartography Cooking Detect Hidden Discordance EvalInt Healing Fishing Forensics Herding Hiding Provocation Inscribe Lockpicking Magery Magic Resist Mysticism Tactics Snooping Musicianship Poisoning Archery Spirit Speak Stealing Tailoring Animal Taming Taste ID Tinkering Tracking Veterinary Swords Macing Fencing Wrestling Lumberjacking Mining Meditation Stealth Remove Trap Necromancy Focus Chivalry Bushido Ninjitsu Spell Weaving Imbuing

Get the skill cap for the given the skill name.

`Player.GetSkillStatus(`*skillname*`)` → Int32

- skillname: `String` Alchemy

Anatomy Animal Lore Item ID Arms Lore Parry Begging Blacksmith Fletching Peacemaking Camping Carpentry Cartography Cooking Detect Hidden Discordance EvalInt Healing Fishing Forensics Herding Hiding Provocation Inscribe Lockpicking Magery Magic Resist Mysticism Tactics Snooping Musicianship Poisoning Archery Spirit Speak Stealing Tailoring Animal Taming Taste ID Tinkering Tracking Veterinary Swords Macing Fencing Wrestling Lumberjacking Mining Meditation Stealth Remove Trap Necromancy Focus Chivalry Bushido Ninjitsu Spell Weaving Imbuing

Get lock status for a specific skill.

`Player.GetSkillValue(`*skillname*`)` → Double

- skillname: `String` Alchemy

Anatomy Animal Lore Item ID Arms Lore Parry Begging Blacksmith Fletching Peacemaking Camping Carpentry Cartography Cooking Detect Hidden Discordance EvalInt Healing Fishing Forensics Herding Hiding Provocation Inscribe Lockpicking Magery Magic Resist Mysticism Tactics Snooping Musicianship Poisoning Archery Spirit Speak Stealing Tailoring Animal Taming Taste ID Tinkering Tracking Veterinary Swords Macing Fencing Wrestling Lumberjacking Mining Meditation Stealth Remove Trap Necromancy Focus Chivalry Bushido Ninjitsu Spell Weaving Imbuing

Get the value of the skill, with modifiers, for the given the skill name.

`Player.GetStatStatus(`*statname*`)` → Int32

- statname: `String` Strength

Dexterity Intelligence

Get lock status for a specific stats.

Player.**GuildButton**() → Void

Press the Guild menu button in the paperdoll.

Player.**HeadMessage**(*color*, *msg*) → Void

- color: Int32
- msg: Int32

Player.**HeadMessage**(*color*, *msg*) → Void

- color: Int32 Color of the Text.
- msg: String Text of the message.

Display a message above the Player. Visible only by the Player.

Player.**InRangeItem**(*item*, *range*) → Boolean

- item: *Item*
- range: Int32

Player.**InRangeItem**(*item*, *range*) → Boolean

- item: Int32
- range: Int32

Player.**InRangeMobile**(*mobile*, *range*) → Boolean

- mobile: Int32
- range: Int32

Player.**InRangeMobile**(*mobile*, *range*) → Boolean

- mobile: *Mobile*
- range: Int32

Player.**InvokeVirtue**(*virtue*) → Void

- virtue: String Honor

Sacrifice Valor Compassion Honesty Humility Justice

Invoke a virtue by name.

Player.**KickMember**(*serial*) → Void

- serial: Int32 Serial of the Mobile to remove.

Kick a member from party by serial. Only for party leader

Player.**LeaveParty**(*force*) → Void

- force: Boolean True: Leave the party invite even you notin any party.

Leaves a party.

Player.**MapSay**(*msg*) → Void

- msg: String Message to send

Send message in the Map chat.

Player.**MapSay**(*msg*) → Void

- msg: Int32

Player.**PartyAccept**(*from_serial*, *force*) → Boolean

- from_serial: Int32 Optional: Serial to accept party from.(in case of multiple offers)
- force: Boolean True: Accept the party invite even you are already in a party.

Accept an incoming party offer. In case of multiple party oebnding invitation, from_serial is specified,

Player.**PartyCanLoot**(*CanLoot*) → Void

- CanLoot: Boolean

Set the Party loot permissions.

Player.**PartyInvite**() → Void

Invite a person to a party. Prompt for a in-game Target.

Player.**PathFindTo**(*x*, *y*, *z*) → Void

- x: Int32 X map coordinates or Point3D
- y: Int32 Y map coordinates
- z: Int32 Z map coordinates

Go to the given coordinates using Client-provided pathfinding.

Player.**PathFindTo**(*Location*) → Void

- Location: *Point3D*

Player.**QuestButton**() → Void

Press the Quest menu button in the paperdoll.

Player.**Run**(*direction*, *checkPosition*) → Boolean

- direction: String North

South East West Up Down Left Right * checkPosition: Boolean True: Wait until the server confirm the new Player.Position - False: Don't wait.

Run one step in the specified direction and wait for the confirmation of the new position by the server. If the character is not facing the direction, the first step only "turn" the Player in the required direction. Optional: When checkPosition is True allow for slower but safe walking, the new position confirmed at each step via return value. When checkPosition is Flase allow for faster walking/running, but requires custom delay and position checking. Info: Walking: 5 tiles/sec (~200ms between each step) Running: 10 tiles/sec (~100ms between each step)

Player.**SetSkillStatus**(*skillname*, *status*) → Void

- skillname: String Alchemy

Anatomy Animal Lore Item ID Arms Lore Parry Begging Blacksmith Fletching Peacemaking Camping Carpentry Cartography Cooking Detect Hidden Discordance EvalInt Healing Fishing Forensics Herding Hiding Provocation Inscribe Lockpicking Magery Magic Resist Mysticism Tactics Snooping Musicianship Poisoning Archery Spirit Speak Stealing Tailoring Animal Taming Taste ID Tinkering Tracking Veterinary Swords Macing Fencing Wrestling Lumberjacking Mining Meditation Stealth Remove Trap Necromancy Focus Chivalry Bushido Ninjitsu Spell Weaving Imbuing * status: Int32 Lock status:

0: Up 1: Down 2: Locked

Set lock status for a specific skill.

Player.**SetStatStatus**(*statname*, *status*) → Void

- statname: String

- **status: Int32 Lock status:** 0: Up 1: Down 2: Locked

Set lock status for a specific skill.

Player.**SetWarMode**(*warflag*) → Void

- warflag: Boolean True: War - False: Peace

Set war Mode on on/off.

Player.**SpellIsEnabled**(*spellname*) → Boolean

- spellname: String Name of the spell.

Check if spell is active using the spell name (for spells that have this function).

Player.**SumAttribute**(*attributename*) → Single

- attributename: String Name of the property.

Scan all the equipped Item, returns the total value of a specific property. (ex: Lower Reagent Cost) NOTE: This function is slow.

Player.**ToggleAlwaysRun**() → Void

Toggle on/off the awlays run flag. NOTE: Works only on OSI client.

Player.**UnEquipItemByLayer**(*layer*, *wait*) → Void

- **layer: String Layers:** RightHand LeftHand Shoes Pants Shirt Head Gloves Ring Neck Hair Waist InnerTorso Bracelet FacialHair MiddleTorso Earrings Arms Cloak OuterTorso OuterLegs InnerLegs Talisman
- wait: Boolean Wait for confirmation from the server.

Unequip the Item associated with a specific Layer.

Player.**UseSkill**(*skillname*, *target*, *wait*) → Void

- skillname: String
- target: *Item*
- wait: Boolean

Player.**UseSkill**(*skillname*, *wait*) → Void

- skillname: String
- wait: Boolean

Player.**UseSkill**(*skillname*, *target*, *wait*) → Void

- skillname: String Alchemy

Anatomy Animal Lore Item ID Arms Lore Parry Begging Blacksmith Fletching Peacemaking Camping Carpentry Cartography Cooking Detect Hidden Discordance EvalInt Healing Fishing Forensics Herding Hiding Provocation Inscribe Lockpicking Magery Magic Resist Mysticism Tactics Snooping Musicianship Poisoning Archery Spirit Speak Stealing Tailoring Animal Taming Taste ID Tinkering Tracking Veterinary Swords Macing Fencing Wrestling Lumberjacking Mining Meditation Stealth Remove Trap Necromancy Focus Chivalry Bushido Ninjitsu Spell Weaving Imbuing * target: Int32 Optional: Serial, Mobile or Item to target. (default: null) * wait: Boolean Optional: True: wait for confirmation from the server (default: False)

Use a specific skill, and optionally apply that skill to the target specified.

Player.**UseSkill**(*skillname*, *target*, *wait*) → Void

- skillname: String
- target: *Mobile*

- wait: `Boolean`

`Player.`**`UseSkill`**`(`*skillname*`)` → Void

- skillname: `String`

`Player.`**`UseSkillOnly`**`(`*skillname*, *wait*`)` → Void

- skillname: `String`
- wait: `Boolean`

`Player.`**`Walk`**`(`*direction*, *checkPosition*`)` → Boolean

- direction: `String` North

South East West Up Down Left Right * checkPosition: `Boolean` True: Wait until the server confirm the new Player.Position - False: Don't wait.

Walk one step in the specified direction and wait for the confirmation of the new position by the server. If the character is not facing the direction, the first step only "turn" the Player in the required direction. Optional: When checkPosition is True allow for slower but safe walking, the new position confirmed at each step via return value. When checkPosition is Flase allow for faster walking/running, but requires custom delay and position checking. Info: Walking: 5 tiles/sec (~200ms between each step) Running: 10 tiles/sec (~100ms between each step)

`Player.`**`WeaponClearSA`**`()` → Void

Disable any active Special Ability of the weapon.

`Player.`**`WeaponDisarmSA`**`()` → Void

Toggle Disarm Ability.

`Player.`**`WeaponPrimarySA`**`()` → Void

Toggle on/off the primary Special Ability of the weapon.

`Player.`**`WeaponSecondarySA`**`()` → Void

Toggle on/off the secondary Special Ability of the weapon.

`Player.`**`WeaponStunSA`**`()` → Void

Toggle Stun Ability.

`Player.`**`Zone`**`()` → String

Get the type of zone in which the Player is currently in. Regions are defined inside by Config/regions.json.

1.26 Point2D

1.26.1 Properties

- Point2D.X `Int32`
- Point2D.Y `Int32`

1.26.2 Methods

`Point2D.`**`ToString`**`()` → String

1.27 Point3D

1.27.1 Properties

- Point3D.X `Int32`
- Point3D.Y `Int32`
- Point3D.Z `Int32`

1.27.2 Methods

`Point3D.`**`ToString`**`()` → String

1.28 Property

1.28.1 Properties

- Property.Args `String`
- Property.Number `Int32`

1.28.2 Methods

`Property.`**`ToString`**`()` → String

1.29 Restock

1.29.1 Methods

`Restock.`**`ChangeList`**`(`*listName*`)` → Void

- listName: `String` Name of an existing restock list.

Change the Restock's active list.

`Restock.`**`FStart`**`()` → Void

Start the Restock Agent on the currently active list.

`Restock.`**`FStop`**`()` → Void

Stop the Restock Agent.

`Restock.`**`Status`**`()` → Boolean

Check Restock Agent status

1.30 Scavenger

1.30.1 Methods

`Scavenger.`**`ChangeList`**`(`*listName*`)` → Void

- listName: `String` Name of an existing organizer list.

Change the Scavenger's active list.

`Scavenger.`**`GetScavengerBag`**`()` → UInt32

Get current Scravenger destination container.

`Scavenger.`**`ResetIgnore`**`()` → Void

`Scavenger.`**`RunOnce`**`(`*scavengerList*, *millisec*, *filter*`)` → Void

- scavengerList: `List[Scavenger.ScavengerItem]`
- millisec: `Int32`
- filter: *`Items.Filter`*

`Scavenger.`**`Start`**`()` → Void

Start the Scavenger Agent on the currently active list.

`Scavenger.`**`Status`**`()` → Boolean

Check Scavenger Agent status

`Scavenger.`**`Stop`**`()` → Void

Stop the Scavenger Agent.

1.31 SellAgent

1.31.1 Methods

`SellAgent.`**`ChangeList`**`(`*listName*`)` → Void

- listName: `String`

`SellAgent.`**`Disable`**`()` → Void

`SellAgent.`**`Enable`**`()` → Void

`SellAgent.`**`Status`**`()` → Boolean

1.32 Spells

1.32.1 Methods

`Spells.`**`Cast`**`(`*SpellName*`)` → Void

- SpellName: `String`

`Spells.`**`Cast`**`(`*SpellName*, *mobile*, *wait*`)` → Void

- SpellName: `String`
- mobile: *Mobile*
- wait: `Boolean`

`Spells.`**`Cast`**`(`*SpellName*, *target*, *wait*`)` → Void

- SpellName: `String` Name of the spell to cast.
- target: `UInt32` Optional: Serial or Mobile to target (default: null)
- wait: `Boolean` Optional: Wait server to confirm. (default: True)

Cast spell using the spell name. See the skill-specific functions to get the full list of spell names. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

`Spells.`**`CastBushido`**`(`*SpellName*, *wait*`)` → Void

- SpellName: `String` Honorable Execution

Confidence Counter Attack Lightning Strike Evasion Momentum Strike * wait: `Boolean` Optional: Wait server to confirm. (default: True)

Cast a Bushido spell using the spell name.

`Spells.`**`CastChivalry`**`(`*SpellName*, *target*, *wait*`)` → Void

- SpellName: `String` Curse Weapon

Pain Spike Corpse Skin Evil Omen Blood Oath Wraith Form Mind Rot Summon Familiar Horrific Beast Animate Dead Poison Strike Wither Strangle Lich Form Exorcism Vengeful Spirit Vampiric Embrace * target: `UInt32` Optional: Serial or Mobile to target (default: null) * wait: `Boolean` Optional: Wait server to confirm. (default: True)

Cast a Chivalry spell using the spell name. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

`Spells.`**`CastChivalry`**`(`*SpellName*, *mobile*, *wait*`)` → Void

- SpellName: `String`
- mobile: *Mobile*
- wait: `Boolean`

`Spells.`**`CastChivalry`**`(`*SpellName*`)` → Void

- SpellName: `String`

`Spells.`**`CastCleric`**`(`*SpellName*, *target*, *wait*`)` → Void

- SpellName: `String` Bark Skin : Turns the druid's skin to bark, increasing physical, poison and energy resistence while reducing fire resistence.

Circle Of Thorns : Creates a ring of thorns preventing an enemy from moving. Deadly Spores : The enemy is afflicted by poisonous spores. Enchanted Grove : Causes a grove of magical trees to grow, hiding the player for a short time. Firefly : Summons a tiny firefly to light the Druid's path. The Firefly is a weak creature with little or no combat skills. Forest Kin : Summons from a list of woodland spirits that will fight for the druid and assist him in different ways. Grasping Roots : Summons roots from the ground to entangle a single target. Hibernate : Causes the target to go to sleep. Hollow Reed : Increases both the strength and the intelligence of the Druid. Hurricane : Calls forth a violent hurricane that damages any enemies within range. Lure Stone : Creates a magical stone that calls all nearby animals to it. Mana Spring : Creates a magical spring that restores mana to the druid and any party members within range. Mushroom Gateway : A magical circle of mushrooms opens, allowing the Druid to step through it to another location. Pack Of Beasts : Summons a pack of beasts to fight for the Druid. Spell length increases with skill. Restorative Soil : Saturates a patch of land with power, causing healing mud to seep through . The mud can restore the dead to life. Shield Of Earth : A quick-growing wall of foliage springs up at the bidding of the Druid. Spring Of Life : Creates a magical spring that heals the Druid and their party. Swarm Of Insects : Summons a swarm of insects that bite and sting the Druid's enemies. Treefellow : Summons a powerful woodland spirit to fight for the Druid. Volcanic Eruption : A blast of molten lava bursts from the ground, hitting every enemy nearby. * target: `UInt32` Optional: Serial or Mobile to target (default: null) * wait: `Boolean` Optional: Wait server to confirm. (default: True)

Cast a Cleric spell using the spell name. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

`Spells.`**`CastCleric`**`(`*SpellName*`)` → Void

- SpellName: `String`

`Spells.`**`CastCleric`**`(`*SpellName*, *mobile*, *wait*`)` → Void

- SpellName: `String`
- mobile: *`Mobile`*
- wait: `Boolean`

`Spells.`**`CastDruid`**`(`*SpellName*, *target*, *wait*`)` → Void

- SpellName: `String` Angelic Faith : Turns you into an angel, boosting your stats. At 100 Spirit Speak you get +20 Str/Dex/Int. Every 5 points of SS = +1 point to each stat, at a max of +24. Will also boost your Anatomy, Mace Fighting and Healing, following the same formula.

Banish Evil : Banishes Undead targets. Auto kills rotting corpses, lich lords, etc. Works well at Doom Champ. Does not produce a corpse however Dampen Spirit : Drains the stamina of your target, according to the description Divine Focus : Heal for more, but may be broken. Hammer of Faith : Summons a War Hammer with Undead Slayer on it for you Purge : Cleanses Poison. Better than Cure Restoration : Resurrection. Brings the target back with 100% HP/Mana Sacred Boon : A HoT, heal over time spell, that heals 10-15 every few seconds Sacrifice : Heals your party members when you take damage. Sort of like thorns, but it heals instead of hurts Smite : Causes energy damage Touch of Life : Heals even if Mortal Strike or poison are active on the target Trial by Fire : Attackers receive damage when they strike you, sort of like a temporary RPD buff * target: `UInt32` * wait: `Boolean`

Cast a Druid spell using the spell name. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

`Spells.`**`CastDruid`**`(`*SpellName*`)` → Void

- SpellName: `String`

`Spells.`**`CastDruid`**`(`*SpellName*, *mobile*, *wait*`)` → Void

- SpellName: `String`
- mobile: *`Mobile`*

- wait: Boolean

Spells.**CastLastSpell**(*target*, *wait*) → Void

- target: UInt32 Optional: Serial or Mobile to target (default: null)
- wait: Boolean Optional: Wait server to confirm. (default: True)

Cast again the last casted spell, on last target.

Spells.**CastLastSpell**(*m*, *wait*) → Void

- m: *Mobile*
- wait: Boolean

Spells.**CastLastSpell**(*wait*) → Void

- wait: Boolean

Spells.**CastLastSpellLastTarget**() → Void

Cast again the last casted spell, on last target.

Spells.**CastMagery**(*SpellName*) → Void

- SpellName: String

Spells.**CastMagery**(*SpellName*, *mobile*, *wait*) → Void

- SpellName: String
- mobile: *Mobile*
- wait: Boolean

Spells.**CastMagery**(*SpellName*, *target*, *wait*) → Void

- SpellName: String Clumsy

Create Food Feeblemind Heal Magic Arrow Night Sight Reactive Armor Weaken Agility Cunning Cure Harm Magic Trap Magic Untrap Protection Strength Bless Fireball Magic Lock Poison Telekinesis Teleport Unlock Wall of Stone Arch Cure Arch Protection Curse Fire Field Greater Heal Lightning Mana Drain Recall Blade Spirits Dispel Field Incognito Magic Reflection Mind Blast Paralyze Poison Field Summon Creature Dispel Energy Bolt Explosion Invisibility Mark Mass Curse Paralyze Field Reveal Chain Lightning Energy Field Flamestrike Gate Travel Mana Vampire Mass Dispel Meteor Swarm Polymorph Earthquake Energy Vortex Resurrection Summon Air Elemental Summon Daemon Summon Earth Elemental Summon Fire Elemental Summon Water Elemental * target: UInt32 Optional: Serial or Mobile to target (default: null) * wait: Boolean Optional: Wait server to confirm. (default: True)

Cast a Magery spell using the spell name. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

Spells.**CastMastery**(*SpellName*, *target*, *wait*) → Void

- SpellName: String Inspire

Invigorate Resilience Perseverance Tribulation Despair Death Ray Ethereal Blast Nether Blast Mystic Weapon Command Undead Conduit Mana Shield Summon Reaper Enchanted Summoning Anticipate Hit Warcry Intuition Rejuvenate Holy Fist Shadow White Tiger Form Flaming Shot Playing The Odds Thrust Pierce Stagger Toughness Onslaught Focused Eye Elemental Fury Called Shot Saving Throw Shield Bash Bodyguard Heighten Senses Tolerance Injected Strike Potency Rampage Fists Of Fury Knockout Whispering Combat Training Boarding * target: UInt32 Optional: Serial or Mobile to target (default: null) * wait: Boolean Optional: Wait server to confirm. (default: True)

Cast a Mastery spell using the spell name. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

`Spells.`**`CastMastery`**`(`*SpellName*, *mobile*, *wait*`)` → Void

- SpellName: `String`
- mobile: *`Mobile`*
- wait: `Boolean`

`Spells.`**`CastMastery`**`(`*SpellName*`)` → Void

- SpellName: `String`

`Spells.`**`CastMysticism`**`(`*SpellName*, *mobile*, *wait*`)` → Void

- SpellName: `String`
- mobile: *`Mobile`*
- wait: `Boolean`

`Spells.`**`CastMysticism`**`(`*SpellName*`)` → Void

- SpellName: `String`

`Spells.`**`CastMysticism`**`(`*SpellName*, *target*, *wait*`)` → Void

- SpellName: `String` Animated Weapon

Healing Stone Purge Enchant Sleep Eagle Strike Stone Form SpellTrigger Mass Sleep Cleansing Winds Bombard Spell Plague Hail Storm Nether Cyclone Rising Colossus * target: `UInt32` Optional: Serial or Mobile to target (default: null) * wait: `Boolean` Optional: Wait server to confirm. (default: True)

Cast a Mysticism spell using the spell name. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

`Spells.`**`CastNecro`**`(`*SpellName*, *mobile*, *wait*`)` → Void

- SpellName: `String`
- mobile: *`Mobile`*
- wait: `Boolean`

`Spells.`**`CastNecro`**`(`*SpellName*`)` → Void

- SpellName: `String`

`Spells.`**`CastNecro`**`(`*SpellName*, *target*, *wait*`)` → Void

- SpellName: `String` Curse Weapon

Pain Spike Corpse Skin Evil Omen Blood Oath Wraith Form Mind Rot Summon Familiar Horrific Beast Animate Dead Poison Strike Wither Strangle Lich Form Exorcism Vengeful Spirit Vampiric Embrace * target: `UInt32` Optional: Serial or Mobile to target (default: null) * wait: `Boolean` Optional: Wait server to confirm. (default: True)

Cast a Necromany spell using the spell name. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

`Spells.`**`CastNinjitsu`**`(`*SpellName*`)` → Void

- SpellName: `String`

`Spells.`**`CastNinjitsu`**`(`*SpellName*, *target*, *wait*`)` → Void

- SpellName: `String` Animal Form

Backstab Surprise Attack Mirror Image Shadow jump Focus Attack Ki Attack * target: `UInt32` Optional: Serial or Mobile to target (default: null) * wait: `Boolean` Optional: Wait server to confirm. (default: True)

Cast a Ninjitsu spell using the spell name. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

`Spells.`**`CastNinjitsu`**`(`*SpellName*, *mobile*, *wait*`)` → Void

- SpellName: `String`
- mobile: *`Mobile`*
- wait: `Boolean`

`Spells.`**`CastSpellweaving`**`(`*SpellName*, *mobile*, *wait*`)` → Void

- SpellName: `String`
- mobile: *`Mobile`*
- wait: `Boolean`

`Spells.`**`CastSpellweaving`**`(`*SpellName*`)` → Void

- SpellName: `String`

`Spells.`**`CastSpellweaving`**`(`*SpellName*, *target*, *wait*`)` → Void

- SpellName: `String` Arcane Circle

Gift Of Renewal Immolating Weapon Attune Weapon Thunderstorm Natures Fury Summon Fey Summoniend Reaper Form Wildfire Essence Of Wind Dryad Allure Ethereal Voyage Word Of Death Gift Of Life Arcane Empowerment * target: `UInt32` Optional: Serial or Mobile to target (default: null) * wait: `Boolean` Optional: Wait server to confirm. (default: True)

Cast a Spellweaving spell using the spell name. Optionally is possible to specify the Mobile or a Serial as target of the spell. Upon successful casting, the target will be executed automatiaclly by the server. NOTE: The "automatic" target is not supported by all shards, but you can restort to the Target class to handle it manually.

`Spells.`**`Interrupt`**`()` → Void

Interrupt the casting of a spell by performing an equip/unequip.

1.33 Statics

1.33.1 Methods

`Statics.`**`CheckDeedHouse`**`(`*x*, *y*`)` → Boolean

- x: `Int32` X coordinate.
- y: `Int32` Y coordinate.

Check if the given Tile is occupied by a private house. Need to be in-sight, on most servers the maximum distance is 18 tiles.

`Statics.`**`GetLandFlag`**`(`*itemid*, *flagname*`)` → Boolean

- itemid: `Int32`

- flagname: `String` None

Translucent Wall Damaging Impassable Surface Bridge Window NoShoot Foliage HoverOver Roof Door Wet

Land: Check Flag value of a given Land item.

`Statics.`**`GetLandID`**`(`*x*, *y*, *map*`)` → Int32

- x: `Int32` X coordinate.
- y: `Int32` Y coordinate.
- **map: `Int32` 0 = Felucca** 1 = Trammel

2 = Ilshenar 3 = Malas

4 = Tokuno 5 = TerMur

Land: Return the StaticID of the Land item, give the coordinates and map.

`Statics.`**`GetLandName`**`(`*StaticID*`)` → String

- StaticID: `Int32` Land item StaticID.

Land: Get the name of a Land item given the StaticID.

`Statics.`**`GetLandZ`**`(`*x*, *y*, *map*`)` → Int32

- x: `Int32` X coordinate.
- y: `Int32` Y coordinate.
- **map: `Int32` 0 = Felucca** 1 = Trammel

2 = Ilshenar 3 = Malas

4 = Tokuno 5 = TerMur

Land: Return the Z coordinate (height) of the Land item, give the coordinates and map.

`Statics.`**`GetStaticsLandInfo`**`(`*x*, *y*, *map*`)` → *Statics.TileInfo*

- x: `Int32` X coordinate.
- y: `Int32` Y coordinate.
- **map: `Int32` 0 = Felucca** 1 = Trammel

2 = Ilshenar 3 = Malas

4 = Tokuno 5 = TerMur

Land: Return a TileInfo representing the Land item for a given X,Y, map.

`Statics.`**`GetStaticsTileInfo`**`(`*x*, *y*, *map*`)` → List[*Statics.TileInfo*]

- x: `Int32` X coordinate.
- y: `Int32` Y coordinate.
- **map: `Int32` 0 = Felucca** 1 = Trammel

2 = Ilshenar 3 = Malas

4 = Tokuno 5 = TerMur

Tile: Return a list of TileInfo representing the Tile items for a given X,Y, map.

`Statics.`**`GetTileFlag`**`(`*StaticID*, *flagname*`)` → Boolean

- StaticID: `Int32` StaticID of a Tile item.

- flagname: `String` None

Translucent Wall Damaging Impassable Surface Bridge Window NoShoot Foliage HoverOver Roof Door Wet

Tile: Check Flag value of a given Tile item.

`Statics.`**`GetTileHeight`**`(`*StaticID*`)` → Int32

- StaticID: `Int32` Tile item StaticID.

Tile: Get hight of a Tile item, in Z coordinate reference.

`Statics.`**`GetTileName`**`(`*StaticID*`)` → String

- StaticID: `Int32` Tile item StaticID.

Tile: Get the name of a Tile item given the StaticID.

1.34 Statics.TileInfo

1.34.1 Properties

- Statics.TileInfo.StaticHue `Int32`
- Statics.TileInfo.StaticID `Int32`
- Statics.TileInfo.StaticZ `Int32`

1.35 Target

1.35.1 Methods

`Target.`**`AttackTargetFromList`**`(`*target_name*`)` → Void

- target_name: `String`

Attack Target from gui filter selector, in Targetting tab.

`Target.`**`Cancel`**`()` → Void

Cancel the current target.

`Target.`**`ClearLast`**`()` → Void

Clear the last target.

`Target.`**`ClearLastandQueue`**`()` → Void

Clear last target and target queue.

`Target.`**`ClearQueue`**`()` → Void

Clear Queue Target.

`Target.`**`GetLast`**`()` → Int32

Get serial number of last target

`Target.`**`GetLastAttack`**`()` → Int32

Get serial number of last attack target

`Target.`**`GetTargetFromList`**`(`*target_name*`)` → *Mobile*

- target_name: `String` Name of the target filter.

Get Mobile object from GUI filter selector, in Targetting tab.

`Target.`**`HasTarget`**`()` → Boolean

Get status if have in-game cursor has target shape.

`Target.`**`Last`**`()` → Void

Execute the target on the last Item or Mobile targeted.

`Target.`**`LastQueued`**`()` → Void

Enqueue the next target on the last Item or Mobile targeted.

`Target.`**`PerformTargetFromList`**`(`*target_name*`)` → Void

- target_name: `String` Name of the target filter.

Execute Target from GUI filter selector, in Targetting tab.

`Target.`**`PromptGroundTarget`**`(`*message*, *color*`)` → *Point3D*

- message: `String` Hint on what to select.
- color: `Int32` Color of the message. (default: 945, gray)

Prompt a target in-game, wait for the Player to select the ground. Can also specific a text message for prompt.

`Target.`**`PromptTarget`**`(`*message*, *color*`)` → Int32

- message: `String` Hint on what to select.
- color: `Int32` Color of the message. (default: 945, gray)

Prompt a target in-game, wait for the Player to select an Item or a Mobile. Can also specific a text message for prompt.

`Target.`**`Self`**`()` → Void

Execute the target on the Player.

`Target.`**`SelfQueued`**`()` → Void

Enqueue the next target on the Player.

`Target.`**`SetLast`**`(`*serial*, *wait*`)` → Void

- serial: `Int32` Serial of the Mobile.
- wait: `Boolean` Wait confirmation from the server.

Set the last target to specific mobile, using the serial.

`Target.`**`SetLast`**`(`*mob*`)` → Void

- mob: *Mobile*

`Target.`**`SetLastTargetFromList`**`(`*target_name*`)` → Void

- target_name: `String` Name of the target filter.

Set Last Target from GUI filter selector, in Targetting tab.

`Target.`**`TargetExecute`**`(`*item*`)` → Void

- item: *Item* Item object to Target.

Target.**TargetExecute**(*serial*) → Void

- serial: Int32 Serial of the Target

Target.**TargetExecute**(*x*, *y*, *z*) → Void

- x: Int32
- y: Int32
- z: Int32

Target.**TargetExecute**(*mobile*) → Void

- mobile: *Mobile* Mobile object to Target.

Target.**TargetExecute**(*x*, *y*, *z*, *StaticID*) → Void

- x: Int32 X coordinate.
- y: Int32 Y coordinate.
- z: Int32 Z coordinate.
- StaticID: Int32 ID of Land/Tile

Execute target on specific serial, item, mobile, X Y Z point.

Target.**TargetExecuteRelative**(*serial*, *offset*) → Void

- serial: Int32 Serial of the mobile
- offset: Int32

Target.**TargetExecuteRelative**(*mobile*, *offset*) → Void

- mobile: *Mobile* Mobile object to target.
- offset: Int32 Distance from the target.

Execute target on specific land point with offset distance from Mobile. Distance is calculated by target Mobile.Direction.

Target.**TargetResource**(*item_serial*, *resource_number*) → Void

- item_serial: Int32 Item object to use.
- **resource_number: Int32 Resource as standard name or custom number** 0: ore 1: sand 2: wood 3: graves 4: red_mushrooms n: custom

Find and target a resource using the specified item.

Target.**TargetResource**(*item*, *resouce_name*) → Void

- item: *Item* Item object to use.
- resouce_name: String

Target.**TargetResource**(*item*, *resoruce_number*) → Void

- item: *Item*
- resoruce_number: Int32

Target.**TargetResource**(*item_serial*, *resource_name*) → Void

- item_serial: Int32
- resource_name: String

Target.**WaitForTarget**(*delay*, *noshow*) → Boolean

- delay: `Int32` Maximum amount to wait, in milliseconds
- noshow: `Boolean` Pevent the cursor to display the target.

Wait for the cursor to show the target, pause the script for a maximum amount of time. and optional flag True or False. True Not show cursor, false show it

1.36 Tile

1.36.1 Properties

- Tile.X `Int32` Coordinate X.
- Tile.Y `Int32` Coordinate Y.

1.36.2 Methods

`Tile.`**`Equals`**`(`*obj*`)` → Boolean

- obj: `Object`

`Tile.`**`GetHashCode`**`()` → Int32

`Tile.`**`ToString`**`()` → String

1.37 Timer

1.37.1 Methods

`Timer.`**`Check`**`(`*name*`)` → Boolean

- name: `String`

Check if a timer object is expired or not.

`Timer.`**`Create`**`(`*name*, *delay*, *message*`)` → Void

- name: `String` Timer name.
- delay: `Int32` Delay in milliseconds.
- message: `String` Message displayed at timeouit.

Create a timer with the provided name that will expire in ms_timer time (in milliseconds)

`Timer.`**`Create`**`(`*name*, *delay*`)` → Void

- name: `String`
- delay: `Int32`

`Timer.`**`Remaining`**`(`*name*`)` → Int32

- name: `String` Timer name

Get remaining time for a named timer

1.38 Vendor

1.38.1 Methods

Vendor.**BuyList**(*vendorSerial*) → List[*Vendor.BuyItem*]

- vendorSerial: Int32 Serial of the Vendor (default: -1 - most recent trade)

Get the list of items purchased in the last trade, with a specific Vendor.

1.39 Vendor.BuyItem

1.39.1 Properties

- Vendor.BuyItem.Amount Int32
- Vendor.BuyItem.ItemID Int32
- Vendor.BuyItem.Name String
- Vendor.BuyItem.Price Int32
- Vendor.BuyItem.Serial Int32

CHAPTER

TWO

TUTORIALS

CHAPTER

THREE

EXAMPLES

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX

H

I

J

M

O

P

R

S

T

V