HERMES
AIRLIFT SERVICES

VER. 1.96

Gunter Severloh & Rydygier

INTRODUCTION

Hermes Airlift Services is an easy to set up yet versatile and customizable mod that enables helicopter airlift of troops or supplies in Arma 3 scenarios for player controlled characters.

Addon (module) version is recommended for users, that want to implement HAS features as simply, as possible, even without writing a single line of code.

Script version is recommended for people, that prefer maximal control over HAS implementation and want to avoid HAS addon dependency for their projects.

Features:

- Airlifting for units to a chosen destination or back to base.
- Module or script version.
- Mouse actions or 0-8 supports variants.
- Task initialized by radio call or by entering a helicopter at base.
- Map click to choose a destination and customize flight path.
- Helicopter will land and give the action to go to a new destination or RTB.
- Variety of customization options through the user config or module settings.
- Ability to cancel pending airlift.
- The ability to point to an alternative LZ if landing is not possible at the chosen coordinates.
- Daylight (default smoke), and night signals (default flares/chemlights) to mark the chosen extraction LZ.
- Text radio chat communications or audio & text from the vanilla airlift support module.
- Customizable call limits.
- Possibility of adding new units and helicopters on the fly.
- Logic handles special situations (not enough cargo space, helicopter destroyed/unable to fly etc.).
- Optional auto embark*/disembark, *except for player controlled units and player lead groups.
- On team switch all playable units defined in the user config will get the "call helicopter" action provided one being switched to is the player.
- Fastroping option.
- Supply Drop - (Slingload or Paradrop).
- Option to map click to a supply drop.
- Changing flight ceiling on-the-fly.
- Option to add virtual Arsenal to supply drop.
- MP/dedi/client side compatibility.
- Respawn compatibility.
- Possibility to deliver as cargo prepared by the user or spawned with procedural content boxes, container or other objects.
- Various ways to determine, which helicopter will take the task.
- Custom CAS

INSTALLATION

1. **HAS addon version** is installed by placing folder "@RYD_HAS" in the same place, where are situated other addonfolders.

2. **HAS script version** is installed by placing contents of “Script version” folder in the mission folder. If mission folder already includes provided files, like *init.sqf* or *description.ext*, their content need to be merged together with HAS counterparts into single files.

USAGE

1. **HAS addon version:** launch the game with HAS addon, place on map some helicopters (recommended flat, empty and safe location at the helipads objects) and units, that will be using these helicopters, then place the HAS module near the helicopters. Lastly, synchronize helicopters and units with the module. Optionally adjust module settings.
   Video tutorial by **Gunter Severloh**

2. **HAS script version**:as above, but instead of placing module, place at helicopters ***RYD_HAS_Base*** Game Logic object and synchronize helicopters and units with it, then place following in the init.sqf:

```
if (isServer) then
    {
    call compile preprocessFile "HAS\userConfig.sqf";
    call compile preprocessFile "HAS\HAS_fnc.sqf";
    [] call RYD_HAS_Init;
    };
```

   As an alternative for syncing, give a name to every chosen helicopter and unit in EDEN. Next open userConfig.sqf file and put these helicopters names into ***RYD_HAS_allChoppers*** array and units names into ***RYD_HAS_STT*** array.
   Video tutorial by **Gunter Severloh**

See included demo missions as reference for proper basic set up. *Important note: caller unit **need to have map item assigned**, without it HAS can’t work properly (map clicks required).* **That’s all for basic usage.**

OPTIONAL CONFIGURATION

Customization via any of listed below configuration variables except mentioned in the **USAGE** section is purely optional. For the script version, config variables can be edited in the userConfig.sqf file. For the module version, all of them may be applied via scripting as well, but also all of them except variables of array kind have own counterpart in the module settings.

Detailed explanation of every user config variable:

RYD_HAS_allChoppers = []; - an array holding names of all helicopters appointed to serve for HAS. If left empty, while one helicopter is named RYD_HAS_Chopper, this helicopter becomes the one and only used.In the module version, when not defined, replaced by helicopters synchronized with the module.

RYD_HAS_STT = []; - an array holding names of all units, that will get possibility to call HAS support. In the module version, when not defined, replaced by units and crews of non-helicopter vehicles synchronized with the module.

RYD_HAS_Base - variable name restricted for Base game logic object used only in scripted version (replaced by the module object itself in module version; module doesn't require any name). Helicopters after completing the task/when RTB will search for suitable landing zone at this object's location. This object also will become a source of all communications tagged "HQ".

RYD_HAS_Limit = 4; - this variable holds an integer. Defines amount of calls available. If set with negative value – calls are unlimited. A call is considered as used as soon it is accepted by the HQ even, if cancelled later. Value can be changed by user on the fly.

RYD_HAS_RadioReq = ""; - a string. Unit need to have an item, which class name includes this string to get access to the HAS calls. Counted are items in the "assigned" item slots, uniform, vest, backpack and weaponry. If empty – no item restriction. Value can be changed on the fly.

RYD_HAS_VoiceChat = false; - boolean. If true, communications between HQ, caller and the pilot will be presented with voice and text via kbTell sentences from Arma 3 vanilla supports module, where possible. If there's no proper voice sentence or if this variable is set to false, all communications will be presented via radio channel text chat with pings***. Note: voice chat IS NOT MP COMPATIBLE***. Keep it "false" for multiplayer gameplay with client-side players.

RYD_HAS_whenInside = true; - boolean. When true, troop transport task will begin starting from choosing destination as soon any of chosen units get in any of the chosen helicopters. First of such units will become a caller (requestor communications sender and unit deciding about the destination, flight ceiling and cancelling the task, but cancelling not available at this point of the call) for this task.

RYD_HAS_CallerName = ""; - a string. Name of the caller to be displayed in the text communications. When left empty, used name includes the rank and name of the caller unit. For players characters by default the name equals to player's profile name.

RYD_HAS_AddHints = true; - boolean. If true, additional hint information about task status etc. will be given. May be useful for filling communication gaps of voice chat for troops transport, mostly redundant if voice chat not used.

RYD_HAS_SignalSearchRadius = 150; - a 2D radius in meters around initial caller's position to search for defined signal object. Sent helicopter needs a signal to know, where to land to pick up awaiting units (or drop the supplies, if map click variant not used). Any object of defined kind will suffice, no matter, who and how placed it. Hint: smoke grenade lifetime seems to be around 60 seconds, may be less or more for various light sources. The search may be CPU-heavy, avoid values above few hundred meters. Note: the caller, after the call, should not move from the initial position farther, than this value, or placing a signal within this radius around call position may become impossible and will stay undetected. Can be changed on the fly.

RYD_HAS_signalTimeLimit = 600; - how long in seconds pilot should loiter around caller's initial position awaiting for the signal before considering the call fail and RTB. Set 0 for unlimited time of waiting. Can be changed on the fly.

RYD_HAS_AutoInOut = true; - boolean. If true, chosen units will be automatically sent to embark the helicopter, and will automatically disembark at destination. Note: fast roping and alternative landing mode enforce auto disembark anyway, except RTB transport, where none of these affect helicopter's landing procedure. May be necessary, when the player is commanded by the AI team leader. Can be change on the fly.

RYD_HAS_RemoveAtLimit = true; - boolean. If true, call actions will be permanently removed from the units after the last call of the limit. Otherwise the actions will only stay invisible until limit is renewed. Can be changed on the fly.

RYD_HAS_ForceDescent = false; - boolean. If true, safe code is used in attempt do counter vanilla AI issue, when helicopter lifts up prematurely at disembark, before all units are out. Does nothing where alternative landing is used. Can be changed on the fly.

RYD_HAS_AlternativeLanding = true; - boolean. If true, alternative landing code is used for troops transport. Makes helicopter touchdown faster, also prevents premature lift up issue. Should be considered as experimental and used with caution. Possibly can make landing helicopter more exposed to collisions with objects like power lines – have this in mind when choosing LZ positions. Does nothing, where fast roping is used and during landings at the base. Can be changed on the fly.

RYD_HAS_UseSupportsMenu = true; - boolean. If true, 0-8 supports menu is used for HAS call actions except destination choice. Otherwise, all actions are available via mouse actions menu. In the 0-8 variant each time caller chooses manually, which helicopter will take the task. In the mouse actions variant each time is chosen the closest, able helicopter.

RYD_HAS_IgnoreAbilityCheck = true; - boolean. Only for mouse actions variant. When false, code will consider, if given helicopter is able to lift supply object before accept this helicopter for the supply drop task. That however may block delivery of the boxes of certain types, like simple ammo boxes, technically light enough to be lifted. If true, helicopter will be accepted no matter, if according to the script command, is able to lift given object. That enables delivery of said kinds of objects (for the sling load – using alternative delivery code), but will also make the helicopter to try lift objects genuinely too heavy for lift, which results with the helicopter permanently attached to the "anchor", futile trying to lift it up. Can be changed on the fly.

RYD_HAS_SignalClassesDay = []; - an array. Holds class names for daytime signals, that mark for the helicopter pick up position or supply delivery position, when map click option not used. Placed signal has to be an object of the given class or any class inheriting from it. For example "SmokeShell" will make any kind of vanilla smoke grenades work as daylight signal. It may be any object. If left empty, no daylight signal required. Can be changed on the fly. Not present in the module settings.

RYD_HAS_SignalClassesNight = []; - an array. Holds class names for nighttime signals, that mark for the helicopter pick up position or supply delivery position, when map click option not used. Placed signal has to be an object of the given class or any class inheriting from it. For example "FlareBase" will make any kind of vanilla flares and chemlights work as daylight signal. If it is a night, script determines automatically, for border hours helpful will be pilot's text communication or one of the additional hints, that points, which kind of signal is requested. If left empty, no nighttime signal required. Can be changed on the fly.Not present in the module settings.

RYD_HAS_SignReqDst = 1000; - 2D distance in meters between the helicopter and the call position, at which pilot will request the signal. Note, if call position is closer, than this value to the helicopter's position at base, request may appear immediately after the call. If the signal is placed before that point, it will be detected too, and no signal request will be sent by the pilot. Otherwise, after request, helicopter will loiter around the area awaiting for the signal until defined waiting time is up. Can be changed on the fly.

RYD_HAS_ChopperLvl = 30; - default desired flight ceiling for HAS tasks in meters. At certain moments, like attaching supply box, fast roping or landings, it will be ignored. Most of the times ignored also will be values considered by the AI as too low (probably < around

30). Can be changed on the fly, will be changed by “change flight ceiling” actions when available.

RYD_HAS_SafetyMargin = 10; - number. The part of the radius for flat and empty area search for safe landing. Bigger - safer. Lower - closer to the chosen position. 10 - reasonable minimum (default). 8 - slightly risky. Big, negative number (like -100) - maximal accuracy and collision risk. Can be changed on the fly.

RYD_HAS_FastRopingLvl = 15; - target hovering altitude during fast roping. Length of deployed ropes will be adjusted accordingly. Final altitude may become higher, if at destination are objects tall enough, to make collision risk on set altitude. Can be changed on the fly, but change will be ignored for pending fast roping.

RYD_HAS_FastRopingOSrad = 0; - radius in meters to search for obstacles of building kind around fast roping point. If any found – new point is tested near instead till one is found. Note, code calculates distance from the center point of the building, not the edges! Objects of other, non-building type are ignored. If set to 0, pointed spot will be accepted regardless of any buildings there. That, for example, will allow rooftop fast roping (not recommended for AI units though). Can be changed on the fly.

RYD_HAS_CruisingSpeed = 400; - maximal cruising speed for helicopters during HAS tasks. Helicopter shouldn’t try to fly faster, although may and usually will fly slower, depending on exact situation or path. Maximal speed, if not exceeding native typical helicopter speed, is likely mostly during long, straight flight path stages. Especially, helicopter will be slowed down at turnings to improve flight path accuracy. If set with value lower than 10, speed becomes unlimited and also helicopter will no longer slowed down at turning. Can be changed on the fly.

RYD_HAS_ContourFlightMode = false; - if enabled, new, lowest flight ceiling is added: "contour flight", where heli will try to fly as low, as possible "hugging" to the terrain profile and "hopping over" map objects. It’s experimental, advanced feature that doesn’t guarantee collisionless flight in 100% – use on your own risk.

RYD_HAS_RespawnHelis = 0; - if positive, destroyed/immobilized HAS helicopter will be after 60 seconds replaced by new one, placed, where destroyed was placed when was added to HAS. Old one will be deleted along with the crew. Value represents number of respawns. 0 - disabled, -1 - unlimited.

RYD_HAS_FastReady = false; - boolean. If true, helicopter will be available for another call at begining of RTB, not at the end of it. This way the heli can be call back right after troops/cargo/CAS delivery, when it is still returning to base. Experimental, use on own risk.

RYD_HAS_Switch_Taxi = true; - boolean. Global switch for troops transportation calls availability. Requires additional step in multiplayer each change – to public this variable over the network, example:

```
RYD_HAS_Switch_Taxi = false;
publicVariable "RYD_HAS_Switch_Taxi";
```

RYD_HAS_Switch_Supply = true; - boolean. Global switch for supply drops calls availability. Requires additional step in multiplayer each change – to public this variable over the network, example:

```
RYD_HAS_Switch_Supply = false;
publicVariable "RYD_HAS_Switch_Supply";
```

RYD_HAS_Switch_CAS = true; - boolean. Global switch for close air support calls availability. Requires additional step in multiplayer – each change to public this variable over the network, example:

```
RYD_HAS_Switch_CAS = false;
publicVariable "RYD_HAS_Switch_CAS";
```

RYD_HAS_addJIPs = true; - boolean. If true, players joining in progress should automatically get HAS calls access.

RYD_HAS_DoorsToOpen = []; - an array. Defines animation sources for given helicopter to be used at fast roping as “door animations”. If empty or no animations found, There will be no door animation. Syntax of each entry: [heliName, [animSource1, animSource2, etc.]]. Exemplary content:

```
[heli1,["door_l","door_r"]],//Ghosthawk
[heli2,["door_l","door_r"]]//Ghosthawk
```

Animation sources vary for each type of helicopter and may be found for given class via config viewer in the EDEN editor. Not present in module settings.

RYD_HAS_RopeAttachPoints = []; - an array. Defines ropes attachment points during fast roping in model space coordinates ([x,y,z], where [0,0,0] means center of the vehicle) per each helicopter. Syntax of each entry: [heliName, [coords1, coords2, etc.]] Exemplary content, that will define two attachment points, 1 meter left from vehicle center and 1 meter right from the center for heli1 and heli2:

```
[heli1,[[-1,0,0],[1,0,0]]],
[heli2,[[-1,0,0],[1,0,0]]]
```

Same defined points will be used with all helicopters of all types. If left empty, code will define attachment points procedurally (if no better point found, will be used single spot: [0,0,0]). Not present in module settings.

RYD_HAS_SupplyDrop_SlingLoad = false; - boolean. Supply delivery mode. If true – helicopter will attempt to sling load defined supply object. If false – supplies will be dropped on parachute. The latter may be considered less vulnerable to issues. Can be changed on the fly.

RYD_HAS_SupplyDrop_onMapClick = true; - boolean. If true, instead of usual signal request, pilot will request a map click to set final deliver coordinates (map will be opened automatically, if accidentally closed, just open the map again). If false, used are same signal rules, as for troops transport. Can be changed on the fly.

RYD_HAS_SupplyDrop_attachFlareAndSmoke = true; - boolean. If true, after delivered supply object touch the ground, smoke (grenade, 60 seconds lifetime) and light (chemlight, 10 minutes lifetime) sources are attached to it for easier spotting. Note: for usual signal class settings these sources may become a proper signal for further calls. Can be changed on the fly.

RYD_HAS_SupplyDrop_Arsenal = false; - boolean. If true, to the supply object at the delivery will be attached an extra mouse action that gives access to the Arma 3 Arsenal feature. Can be changed on the fly.

RYD_HAS_SupplyDrop_Prepared = []; - an array. Variable holds names of supply objects manually prepared by the user to be delivered before default, spawned objects will be used. In the module version, if this variable not defined, used will be all synchronized with the module objects of"ReammoBox_F" kind or any other synchronized objects, which init fields includes this code:

```
this setVariable ["RYD_HAS_ThisIsBox",true];
```

This way delivered as supply may be any object, including vehicles, if helicopter is actually able to lift its weight.

RYD_HAS_SupplyCall_ContainerClass = "B_supplyCrate_F"; - string. Class name of the spawned supply object that will be delivered. Can be changed on the fly.

RYD_HAS_SupplyCall_ContainerPositions = []; - an array. Defines positions on the ground at the landed helicopters, in model space coordinates, where positioned will be supply object for delivery in slingload mode. The position need to be at least 60-80 meters from the helicopter – if too close, helicopter has serious issues with hooking it. Also the positions need to be accessible from above, optimally without any tall objects near. Exemplary content, that will define these positions for both helicopters 80 meters ahead at their initial facing direction:

```
        [heli1,[0,80,0]],
        [heli2,[0,80,0]]
```

If left empty, code will define these spots automatically, same, as in the above example. Final position may vary, as there's performed a check, if defined position is safe and valid indeed. Not present in module settings.

RYD_HAS_SupplyCall_MagCount = -12; - integer. By default inside dropped supply container will be present every type of magazine compatible with each piece of weaponry used by each of defined STT units. This value defines amount of every type per each weapon piece using it. When integer is negative – value is randomized from 0 to positive opposition of this value per each weapon piece.

RYD_HAS_CAS_Distance = 600; - positive number (meters). Helicopter performing CAS task will take firing position at waypoint located that far from CAS coordinates towards helicopter's current position. The lesser value, the better accuracy with not guided weaponry, but also bigger risk of taking returning fire. Final distance may be lower down to 60% of this value, if objects draw distance is lower, than set value. Recommended values are 600-1000 meters.

RYD_HAS_CAS_Range = 150; - positive number (meters). During CAS fire helicopter seeks for hostile targets around CAS cords to engage them. This value defines the radius of this search.

RYD_HAS_CAS_OnlyKnownTargets = false; - boolean. If true, engaged will be only those of hostile targets found at CAS cords, that are actually known to the helicopter's crew or caller unit.

RYD_HAS_CAS_reammo = true; - boolean. If true, helicopter designated for CAS task will have restored full ammunition at the beginning.

RYD_HAS_SupplyCall_AddBackPack= []; - an array. Additional backpack classes added to each of spawned supply containers. Syntax of each entry: [class (string),amount (integer),if randomize from 0 to amount (boolean)]. Not present in module settings.

RYD_HAS_SupplyCall_AddWeapon = []; - ditto, for weapons.

RYD_HAS_SupplyCall_AddItem= []; - ditto, for items.

RYD_HAS_SupplyCall_AddMagazine = []; - ditto, for magazines.

RYD_HAS_SupplyCall_MagBlacklist = []; - an array. Magazine classes (**all lower-case!**) included in this array will be not automatically added to the supply container. Not present in module settings. Doesn't affect **RYD_HAS_SupplyCall_AddMagazine** magazines.

RYD_HAS_sent_Request = []; - an array. Includes strings – sentences, from which one will be each time randomly selected at troops transport call, by the caller, when voice communication not used. In this strings ***%1*** refer to the caller's name (see also **RYD_HAS_CallerName**). Exemplary sentence:

```
"HQ, this is %1. We need heli transport, how copy?"
```

Not present in module settings.

RYD_HAS_sent_Request_sc = []; - Ditto, for supply drop call.

RYD_HAS_sent_Request_cas = []; - Ditto, for CAS call.

RYD_HAS_sent_Confirmation = []; - similar, troops transport assignment confirmation, by HQ.

RYD_HAS_sent_Request_sc = []; - Ditto, for supply drop call.

RYD_HAS_sent_Request_cas = []; - Ditto, for CAS call, coords request from HQ.

RYD_HAS_sent_SignalReq = []; - request for a signal during troops transport task, by pilot.

RYD_HAS_sent_SignalReq_sc= []; - Ditto, for supply drop call.

RYD_HAS_sent_noLZ = []; - request for alternative LZ via map click, when suitable LZ not found at signaled position, by pilot.

RYD_HAS_sent_SignalSpot = []; - when helicopter spot the signal, by pilot.

RYD_HAS_sent_SignalSpot_sc = []; - Ditto, for supply drop call.

RYD_HAS_sent_CoordsConfirmed_cas = []; - pilot confirms CAS coords.

RYD_HAS_sent_AltCoordsConfirmed= []; - acceptance of alternative LZ coordinates, by pilot.

RYD_HAS_sent_AltCoordsConfirmed_cas = []; - pilot confirms corrected CAS coords.

RYD_HAS_sent_CASImminent_cas = []; - pilot informs about reaching CAS firing position.

RYD_HAS_sent_CASBegin_cas = []; - pilot informs, helicopter is about to open fire.

RYD_HAS_sent_CASEnd_cas = []; - pilot informs, CAS is completed (RTB).

RYD_HAS_sent_AirliftCancelled = []; - airlift cancelling, by caller to pilot (***%1*** – caller's name).

RYD_HAS_sent_AirliftCancelled_sc = []; - Ditto, for supply drop call.

RYD_HAS_sent_CASCancelled_sc = []; - Ditto, for CAS call.

RYD_HAS_sent_AirliftCancelledConf = []; - airlift cancel confirmed, by pilot to caller.

RYD_HAS_sent_AirliftCancelledConf_sc= []; - Ditto, for supply drop call.

RYD_HAS_sent_CASCancelledConf_cas = []; - Ditto, for CAS call.

RYD_HAS_sent_Welcome = []; - unit welcomed in the helicopter, destination request, by pilot.

RYD_HAS_sent_LZSelected = []; - at destination chosen, by pilot.

RYD_HAS_sent_Accomplished= []; - at destination reached/ disembark, by pilot.

RYD_HAS_sent_Accomplished_sc= []; - after supply drop, by pilot.

RYD_HAS_sent_Destroyed= []; - at helicopter destroyed or unable to fly, by HQ.

RYD_HAS_sent_Destroyed_sc = []; - Ditto, for supply drop call.

Additional advanced settings:

In order to add new manned helicopter into HAS in the middle of the gameplay, user can utilize this code:

```
HeliName call RYD_HAS_NewChopper;
```

Where *HeliName* is either name of the helicopter object set in EDEN either code reference to helicopter object.

In similar way user can add during the game new units into HAS:

```
[newUnit1,newUnit2,newUnit3(etc.)] call RYD_HAS_NewUnits;
```

In the mouse actions variant used may be two variables defined per each helicopter name space separately: ***"RYD_HAS_TransportPriority"*** and ***"RYD_HAS_SupplyPriority"***. Allow to determine, which helicopter should be used for given task in the first place, regardless of the distance, and also which helicopter shouldn't be used for given task at all. The higher value, the higher priority of usage. Application for example via helicopter's init field, pasting there:

```
this setVariable ["RYD_HAS_TransportPriority",1];//the heli gets priority 1 for troops
transport (will be used before any heli of 0 priority
```

or:

```
this setVariable ["RYD_HAS_SupplyPriority",12];//priority 12 for supply transport (will be
used before any heli with priority lesser, than 12)
```

or:

```
this setVariable ["RYD_HAS_SupplyPriority",-1];//when negative number, heli is excluded
completely from given tasks
```

etc.

By default both values are for all helicopters set to 0. In such case, used is normal procedure - closest is chosen. But if there's at least one able helicopter with priority set by user >0, distance will matter only amongst helicopters of the same, highest priority, so if the highest priority for given type of tasks is 10, and two helicopters has it, the closer of these two will be used, no matter, how close are all other helicopters.

This way user can decide, which helicopters will be used for which tasks and in what exact order.

Since 1.5 in analogous way user can adjust priorities for CAS, using ***"RYD_HAS_CASPriority"***.

User can also manually turn off and on at any time ability to use HAS actions for any of the units chosen to be users of the HAS. To do so, apply to unit namespace ***"RYD_HAS_canCall"*** boolean variable. Example:

```
Unit1 setVariable ["RYD_HAS_canCall",false];//this will hide HAS actions for Unit1
```

or:

```
Unit1 setVariable ["RYD_HAS_canCall",true];//this will reveal HAS actions for Unit1
```

Also it is possible to switch on/off only chosen type of calls availability per unit:

```
unitName setVariable ["RYD_HAS_canCall_Taxi",false,true];//disabling troops transportation
support availability for 'unitName' unit

unitName setVariable ["RYD_HAS_canCall_Taxi",true,true];//enabling troops transportation
support availability for 'unitName' unit
```

similar for the other types of calls:

```
unitName setVariable ["RYD_HAS_canCall_Supply",false,true];
unitName setVariable ["RYD_HAS_canCall_CAS",false,true];
```

HOW IT WORKS

Once the mission initiates, all defined units will get the **"Call helicopter"**, **"Call Supply Drop"** and **"Call CAS"** added to their action menu or 0-8 support menu giving them the ability to call an airlift or close air support.

It is visible only to the player-controlled characters (making a call require player action). The action represents a radio call for airlift.

When used one of two airlifit calls, if all is OK, the number of calls left will be decreased by 1 and after a short background radio chat, the closest helicopter able to performthe task will start, for slingload supply drop will hook the supply, and fly towards the caller's position. When approaching, the pilot will search for presence of any proper signalputting the extraction LZ/supply drop spot around caller's position (of any origin, not only placed by the player!).

The signal is typically a smoke grenade or flare for troops transport and map click for supply drop. If there was no signal given before the helicopter arrives, the pilot will request a signalbased on if its day or night and start to loiter over the area for set amount of time waiting for signal to be given.

If the signal is not given in that time, or airlift cancelled, the helicopter will RTB (supply is simply detached and will fall to the ground).

If the signal is seen thenfor the supply call, supplies are delivered at the spot and helicopter is RTB, which ends the task.

For the troops transport, the pilot will scan the area for a safe LZ spot.

If not found, the player will be informed and asked to point to an alternate extraction coordinate via map click.

This will be repeated until a suitable spot is found. When the LZ is found, the helicopter will land allowing units to embark. Till this point the callermay cancel the airlift via another action.

The caller will get three actions:**transport to another location (touchdown), transport to another location (fast rope) – both pointed via map click** and**RTB.**

After the destination and way of extraction is chosen, the helicopter will fly there, land allowing disembark, wait until all units are out, then RTB.

Except for RTB, during the fly towards chosen coordinates, caller may change destination and way of extraction by choosing again one of two “another location” actions. Click on the water or out of the map boundaries will cancel the choosing. It will be also auto-cancelled, if meanwhile helicopter begin landing stage at current destination.

After the helicopter lands back at base a new call can be made.

If units are closer than 100 meters to the closest of appointed helicopters, radio call is not available.

In this case, or if player chooses, the airlift may be initialized simply by all units embarking in the appointed helicopter (optional).

Once inside, the actions for choosing a destination will become available, returning back to base will not.

The task will immediately end if:

- There's no more appointed helicopters alive for airlifting units.
- Chosen helicopteris destroyed or deleted.
- Pilot of the chosen helicopteris killed or deleted.
- Chosen helicopteris unable to fly due to damage or lack of fuel.
- Airlift is cancelled by the player.

CAS task can be performed only by attack helicopters (gunship) amongst all appointed for HAS use helicopters. As such will be considered those equipped with any missile or rocket based weaponry. Before actual radio call, the caller chooses CAS pattern using mouse actions: HOVER or SAD. Note, in the HOVER pattern it is assumed, CAS helicopter will shoot straight forward, side or back weaponry not supported properly in this pattern. When this call is used, helicopter for the task and the CAS pattern is successfully chosen, HQ will immediately ask for CAS target coordinates and map will be opened. Coordinates are pointed via mouse click within map boundaries. Blue circle marker will appear at clicked coordinates, accompanied with triangle arrow marker showing current expected CAS approach azimuth and firing position. Radius of the circle marker corresponds to ***RYD_HAS_CAS_Radius*** variable value. Assigned attack helicopter will fly towards firing position located between initial position and CAS coords. Distance is defined by ***RYD_HAS_CAS_Distance*** variable. As long helicopter is en route, caller can change CAS coordinates by clicking new position on the map and change CAS pattern. When on place, markers becomeorange, CAS coordinates final/fixed. In the HOVER pattern helicopter will try to assume altitude allowing to get not blocked line of sight on the CAS coordinates and detected hostile vehicles within the attack radius, but “within reason”. When ready, markers become red and helicopter will open fire. Exact order of engagement depends on available weaponry. If AT homing missiles are available, helicopter will start from attacking with them hostile armored vehicles, aerial assets and soft vehicles. Next will be used not guided weaponry to engage soft targets, including infantry and vehicles crew. In this case adaptive targeting code is applied to improve fire accuracy – helicopter will dynamically adjust its aim, if probe shots was missing the target in average. Each target, if not destroyed, gets certain amount of “attention” then fire is moved to another in the queue. This pattern is looped. Targets hidden behind terrain features or map objects will be ignored. Targets inside/just behind buildings may be engaged though. If there’s no suitable targets to engage, helicopter will instead shortly fire with not guided weaponry at CAS coordinates directly. In the SAD pattern gunship will instead perform “Seek and Destroy” sweep as normal, under control of vanilla SAD waypoint.

Caller can cancel CAS at any point. Task will be concluded when helicopter got too much damage in the process, is destroyed, unable to fly or fire, also due to killed crew members. Other than that CAS will be ended when all chosen targets are neutralized or pilot decide, further fire is futile/not justifies the risk (semi-randomized, the longer fire is sustained, the bigger chance for it, if amount of targets doesn't decrease and the less targets was left. Pilot may be prone to make such consideration sooner, if helicopter is damaged).

Depending on used weaponry, kind and number of targets CAS fire may take from below one minute to few minutes.

At appropriate moments user can customize helicopter's rough flight path: between the call and HQ confirmation for all tasks, between choosing destination spot and reaching it for troops transport and for CAS tasks before helicopter reach CAS position (whenever map is automatically opened with related instructions on screen).

It is done via map clicks, which add, confirm or remove navigation points. To add new navpoint: shift+LMB. To clear whole path: alt+LMB. To confirm/execute flight plan: LMB. If you want to set custom path rather than straight flight from the base to destination, choose navpoints before LMB click, which will in the same time confirm the path and destination. During the CAS tasks, before helicopter reach CAS position, and during troops transport, between choosing destination and reaching it (in other words, whenever user is allowed to change destination dynamically), user can same way modify the flight path on the fly, at any moment, and helicopter should immediately follow the new path. New navpoints will be added after the last existing navpoint, but before destination waypoint. To confirm new path without changing final destination, click LMB out of valid destination area: on the sea or out of map boundaries for troops transport and out of map boundaries for CAS.

Whenever helicopter is RTB, his returning flight path will mimic in reversed order all navpoints already reached. New navpoints are marker on the map with blue triangle, reached navpoints are marked with grey color, while grey navpoints reached again during RTB are removed from the map.

If the flight path includes turning points, helicopter, that is closing in to one of them or whenever is facing other direction, than towards next navpoint, will be slowed down to keep its actual flight as close to planned path, as possible. This behavior can be disabled via setting ***RYD_HAS_CruisingSpeed*** variable with value lower, than 10.