

FFI function overview




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

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

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


Version	Structure	Note
added in 4.00 Beta 4	<code>typedef uint64_t TradeID;</code>	
added in 3.50 Beta 2	<code>typedef uint64_t UniverseID;</code>	
added in 4.00 Beta 6	<code>typedef struct { const char* macro; const char* ware; uint32_t amount; uint32_t capacity; } AmmoData;</code>	
added in 4.21 Beta 1 (VR: 4.20)	<code>typedef struct { float angle; bool inside; } ArrowDetails;</code>	
	<code>typedef struct { const char* name; float hull; float shield; bool hasShield; } ComponentDetails;</code>	
added in 4.31 Beta 1	<code>typedef struct { uint32_t messageID; bool obstructed; } CrosshairMessage;</code>	
added in 4.00 Beta 1	<code>typedef struct { uint32_t width; uint32_t height; uint32_t xHotspot; uint32_t yHotspot; } CursorInfo;</code>	
added in 4.00 Beta 1	<code>typedef struct { uint32_t red; uint32_t green; uint32_t blue; uint32_t alpha; } Color;</code>	
renamed in 3.50 Beta 3	<code>typedef struct { const char* factionName; const char* factionIcon; } Owner FactionDetails;</code>	
added in 4.00 Beta 1	<code>typedef struct { const char* name; uint32_t size; } Font;</code>	


added in 4.21 Beta 1 (VR: 4.20)	typedef struct { double fps; double moveTime; double renderTime; double gpuTime; } FPSDetails ;	<i>moveTime</i> , <i>renderTime</i> , and <i>gpuTime</i> always return 0
	typedef struct { int major; int minor; } GameVersion ;	
added in 4.00 Beta 1	typedef struct { double x; double y; } GraphDataPoint ;	
added in 4.00 Beta 1 / changed in 4.00 Beta 5	typedef struct { uint32_t MarkerType; uint32_t MarkerSize; Color MarkerColor; uint32_t LineType; uint32_t LineWidth; Color LineColor; size_t NumData; bool Highlighted; const char* MouseOverText; } GraphDataRecord ;	4.00 Beta 5: added <i>MouseOverText</i> field
added in 4.00 Beta 1	typedef struct { size_t DataRecordIdx; size_t DataIdx; const char* IconID; const char* MouseOverText; } GraphIcon ;	
added in 4.00 Beta 1	typedef struct { const char* text; Font font; Color color; } GraphTextInfo ;	
added in 4.00 Beta 1	typedef struct { GraphTextInfo label; double startvalue; double endvalue; double granularity; bool grid; Color color; Color gridcolor; } GraphAxisInfo ;	
added in 4.00 Beta 2	typedef struct { bool charging; bool jumping; float charge; } JumpDriveDetails ;	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	typedef struct { int32_t id; const char* name; const char* warning; const char* font; bool voice; } LanguageInfo ;	only available in XR VR
added in 3.50 Beta 2	typedef struct { const char* HintText; long HintID; } LoadingHint ;	

	<pre>typedef struct { uint64_t poiID; UniverseID componentID; const char* messageType; const char* connectionName; bool isCollectable; bool isAssociative; bool showIndicator; } MessageDetails;</pre>	
deprecated in 4.00 Beta 3	<p>DEPRECATED</p> <pre>typedef struct { const char* missionName; const char* missionDescription; int difficulty; const char* mainType; const char* subType; const char* faction; int64_t reward; const char* rewardText; size_t numBriefingObjectives; int activeBriefingStep; const char* opposingFaction; const char* license; float timeLeft; double duration; bool abortable; UniverseID associatedComponent; } MissionDetails;</pre>	<p> DEPRECATED - use MissionDetails2 instead</p>
added in 4.00 Beta 3	<pre>typedef struct { const char* missionName; const char* missionDescription; int difficulty; const char* mainType; const char* subType; const char* faction; int64_t reward; const char* rewardText; size_t numBriefingObjectives; int activeBriefingStep; const char* opposingFaction; const char* license; float timeLeft; double duration; bool abortable; bool hasObjective; UniverseID associatedComponent; } MissionDetails2;</pre>	
	<pre>typedef struct { const char* objectiveText; float timeout; size_t numTargets; } MissionObjective;</pre>	
	<pre>typedef struct { const char* text; int step; } MissionObjectiveStep;</pre>	
	<pre>typedef struct { const char* POIName; const char* POIType; } POIDetails;</pre>	
	<pre>typedef struct { int x; int y; } Position2D;</pre>	

added in 4.21 Beta 1 (VR: 4.20)	typedef struct { float x; float y; float z; float yaw; float pitch; float roll; } PosRot ;	Note: yaw/pitch/roll are given in radian.
added in 3.50 Beta 3	typedef struct { int relationStatus; int relationValue; int relationLEDValue; bool isBoostedValue; } RelationDetails ;	
added in 4.10 Beta 1	typedef struct { int x; int y; } ResolutionInfo ;	
added in 4.10 Beta 1	typedef struct { float yaw; float pitch; float roll; } Rotation ;	Note: yaw/pitch/roll are given in radian.
	typedef struct { UniverseID softtargetID; const char* softtargetName; const char* softtargetConnectionName; } SofttargetDetails ;	
added in 4.00 Beta 6	typedef struct { int64_t trade; int64_t defence; int64_t missile; } SupplyBudget ;	
added in 4.00 Beta 6	typedef struct { const char* macro; int amount; } SupplyOverride ;	
added in 4.00 Beta 6	typedef struct { double time; int64_t money; } UIAccountStatData ;	
added in 4.00 Beta 5	typedef struct { double time; uint64_t amount; } UICargoStatData ;	
added in 4.00 Beta 5	typedef struct { const char* wareid; UICargoStatData* data; uint32_t numdata; } UICargoStat ;	
added in 4.00 Beta 1	typedef struct { float x; float y; float z; float yaw; float pitch; float roll; } UIPosRot ;	


added in 4.00 Beta 2	typedef struct { double time; int64_t price; int amount; int limit; } UITradeOfferStatData ;	
added in 4.00 Beta 2	typedef struct { const char* wareid; bool isSellOffer; UITradeOfferStatData* data; uint32_t numdata; } UITradeOfferStat ;	
added in 4.00 Beta 4	typedef struct { const char* wareid; uint32_t amount; } UIWareAmount ;	
added in 4.00 Beta 3	typedef struct { uint32_t ID; const char* Name; const char* RawName; const char* WeaponMacro; const char* Ware; float DamageFactor; float CoolingFactor; float ReloadFactor; float SpeedFactor; float LifeTimeFactor; float MiningFactor; float StickTimeFactor; float ChargeTimeFactor; float BeamLengthFactor; uint32_t AddedAmount; } UIWeaponMod ;	
	typedef struct { const char* ClusterName; const char* SectorName; const char* ZoneName; bool isLocalHighway; bool isSuperHighway; } ZoneDetails ;	



Version	Function	Note
	void AbortCurrentNotification ()	
	void ActivateDirectInput ()	
added in 4.22 Beta 2 (VR: 4.22)	void ActivatePlayerControls ()	
	UniverseID AddHoloMap (const char* texturename, float x0, float x1, float y0, float y1) <i>Adds a holomap on the specified rendertarget.</i>	
added in 4.00 Beta 3	void AddTradeWare (UniverseID containerid, const char* wareid) <i>Add wareid as a tradeware to container.</i>	
added in 4.00 Beta 5	bool AllowPersonalizedData ()	 Designed for internal use only.
added in 4.00 RC 2	bool AreShipSystemsEnabled () Indicates whether the (player) ship systems are enabled.	Note: This is used in the vanilla game only at the early state during the campaign before the ship systems are enabled. In all other cases the ship systems are always enabled.

	bool AreAssertsEnabled()	
	bool AreMandatoryAssertsEnabled()	
	bool AreOptionalAssertEnabled()	
added in 4.00 Beta 3	bool CanActivateSeta() <i>Indicates whether we can activate SETA atm.</i>	
	bool CanActivateSmartDefense() <i>Indicates whether smart defense can be activated.</i>	
	bool CanCancelRecallDrones() <i>Indicates whether recalling drones can be canceled atm.</i>	
	void CancelRecallDrones() <i>Cancels recalling drones.</i>	
added in 4.00 Beta 5	bool CanOpenWebBrowser()	 Designed for internal use only.
	bool CanRecallDrones() <i>Indicates whether drones can be recalled atm.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void CircularMenuClosed() <i>Notifies the system that the circular menu was closed.</i>	
	void ClearErrorLog() <i>Clears the internal error log.</i>	
	void ClearHighlightMapComponent (UniverseID holomapid) <i>Clear the highlighted component on the specified holomap.</i>	
	bool CopyToClipboard (const char*const text) <i>Copies the specified text to the clipboard (so it can be pasted outside the game for instance).</i>	
added in 4.21 Beta 1 (VR: 4.20)	bool CycleNextPrimaryWeapon() <i>Cycles the player's primary weapons (to the next one). The return value indicates whether weapon cycling failed (note: it's considered a success if the function is called without any weapon or only a single one).</i>	
added in 4.21 Beta 1 (VR: 4.20)	bool CycleNextSecondaryWeapon() <i>Cycles the player's secondary weapons (to the next one). The return value indicates whether weapon cycling failed (note: it's considered a success if the function is called without any weapon or only a single one).</i>	
added in 4.21 Beta 1 (VR: 4.20)	bool CyclePrevPrimaryWeapon() <i>Cycles the player's primary weapons (to the previous one). The return value indicates whether weapon cycling failed (note: it's considered a success if the function is called without any weapon or only a single one).</i>	
added in 4.21 Beta 1 (VR: 4.20)	bool CyclePrevSecondaryWeapon() <i>Cycles the player's secondary weapons (to the previous one). The return value indicates whether weapon cycling failed (note: it's considered a success if the function is called without any weapon or only a single one).</i>	
	void DeactivateDirectInput()	
added in 4.22 Beta 2 (VR: 4.22)	void DeactivatePlayerControls()	
	void DisableAsserts()	

added in 4.00 RC 2	void DisableMapPicking (UniverseID holomapid)	
	void DisableOptionalAsserts ()	
added in 4.00 Beta 3	bool DismantleWeaponMod (uint32_t weaponmodid)	
	void EnableAsserts ()	
added in 4.00 RC 2	void EnableMapPicking (UniverseID holomapid)	
	void EnableOptionalAsserts ()	
added in 4.21 Beta 1 (VR: 4.20)	void EnterInteractMenu () <i>Starts the interact menu interaction mode.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void FadeScreen (float fadeouttime, float fadeintime) <i>Starts fading out and back in the screen with the given timings.</i>	
	UniverseID GetActiveMissionID () <i>Returns the active mission id.</i>	
added in 4.21 Beta 1 (VR: 4.20)	float GetAdaptiveSamplingOption ()	
added in 4.10 RC 1	uint32_t GetAllMissiles (AmmoData* result, uint32_t resultlen, UniverseID defensibleid) <i>Stores the missile types an object can use in result. Returns the number of results.</i>	
added in 4.00 Beta 6	uint32_t GetAmmoStorage (AmmoData* result, uint32_t resultlen, UniverseID defensibleid, const char* ammotype); <i>Stores the ammo storage data in result. Returns the number of results.</i>	
	const char* GetBuildSourceSequence (UniverseID componentid) <i>Returns the build sequence of the specified component.</i>	
added in 4.10 Beta 1	Rotation GetCameraRotation () <i>Returns the player's current camera rotation.</i>	
added in 3.50 Beta 3	int GetCargoSpaceUsedAfterShoppingList (UniverseID containerid) <i>Returns the cargo space used after the trade queue.</i>	SetVirtualCargoMode () must be called prior to calling GetCargoSpaceUsedAfterShoppingList ()
added in 4.00 Beta 5	uint32_t GetCargoStatistics (UICargoStat* result, uint32_t resultlen, size_t numdatapoints) <i>Stores the cargo statistics in result. Returns the number of results.</i>	
added in 4.00 Beta 2	bool GetCheckBoxColor (const int checkboxid, Color* color)	
	const char* GetComponentClass (UniverseID componentid) <i>Returns the class of the specified component.</i>	
	ComponentDetails GetComponentDetails (const UniverseID componentid)	
	const char* GetComponentName (UniverseID componentid) <i>Returns the name of the specified component.</i>	
	int GetConfigSetting (const char*const setting)	




added in 4.00 Beta 3	bool GetContainerWarelsBuyable (UniverseID containerid, const char* wareid) <i>Returns whether the specified ware is buyable by the specified container.</i>	
added in 4.00 Beta 3	bool GetContainerWarelsSellable (UniverseID containerid, const char* wareid) <i>Returns whether the specified ware is sellable by the specified container.</i>	
	UniverseID GetContextByClass (UniverseID componentid, const char* classname, bool includeself) <i>Returns the context of the specified component with the given class.</i>	
	int64_t GetCreditsDueFromPlayerTrades () <i>Returns the amount of money expected from outstanding trade queues.</i>	
added in 4.10 Beta 1	ArrowDetails GetCrosshairArrowDetails (int posid, float radius) <i>Returns the crosshair arrow details (ArrowDetails.angle in degree - upper position 0° - clock-wise; ArrowDetails.inside indicating whether the element is within the specified radius; ArrowDetails.valid indicates whether the return value is valid). posid corresponds to the target element and radius is an arbitrary number specifying a radius around the center of the crosshair. If the target element is within that radius, ArrowDetails.inside will be true.</i>	
deprecated in 4.31 Beta 1	DEPRECATED uint32_t GetCurrentCrosshairMessage ()	 DEPRECATED - use GetCurrentCrosshairMessage2() instead
added in 4.31 Beta 1	CrosshairMessage GetCurrentCrosshairMessage2 ()	
added in 4.00 Beta 1	CursorInfo GetCurrentCursorInfo ()	
added in 4.22 Beta 2 (VR: 4.22)	double GetCurrentGameTime () <i>Returns the current game time.</i>	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	int32_t GetCurrentLanguage () <i>Returns the ID of the currently used language.</i>	only functional in X Rebirth VR Edition
	UniverseID GetCurrentlyScannedObject ()	
added in 4.00 Beta 1	uint32_t GetCurrentMissionOffers (uint64_t* result, uint32_t resultlen, bool showninbbs) <i>Stores the missionIDs of current mission offers in result, returns the number of items in result.</i>	See also GetNumCurrentMissionOffers ()
added in 4.00 Beta 5	float GetDefensibleDPS (UniverseID defensibleid, bool primary, bool secondary, bool turrets) <i>Get the damage per second value for the given defensible, using the flagged weapon systems.</i>	
added in 3.50 RC 1	float GetDistanceBetween (UniverseID component1id, UniverseID component2id) <i>Returns the distance between the specified components.</i>	
	float GetDistanceToPlayerShip ()	
	const char* GetEntityActiveIcon (const UniverseID componentid)	

	const char* GetEntityInactiveIcon (const UniverseID componentid)	
	const char* GetEntitySelectedIcon (const UniverseID componentid)	
	UniverseID GetEnvironmentObject ()	
added in 4.00 Beta 2	UniverseID GetExternalTargetViewComponent () <i>Retrieves the component the external target view is set to (only valid in external target view).</i>	
added in 3.50 Beta 3	DEPRECATED const char* GetFactionNameForEventMonitorHack (UniverseID componentid)	 DEPRECATED - use GetOwnerDetails() instead
added in 3.50 Beta 3	const char* GetFactionName (const UniverseID componentid)	
added in 3.50 Beta 3	RelationDetails GetFactionRelationStatus (const char* factionid)	
added in 4.10 Beta 1	bool GetForceShootingAtCursorOption () <i>Returns whether shooting is forced to always aim towards the cursor position (rather than also towards the crosshair).</i>	
added in 4.21 Beta 1 (VR: 4.20)	FPSDetails GetFPS () <i>Returns the current frames per second.</i>	
	uint32_t GetFXAAOption () <i>Returns the current FXAA setting.</i>	
	GameVersion GetGameVersion ()	
added in 4.00 Beta 1	bool GetGraphBackgroundColor (const int graphid, Color* color)	
added in 4.00 Beta 1	uint32_t GetGraphData (GraphDataPoint* result, uint32_t resultlen, const int graphid, const size_t datarecordidx)	
added in 4.00 Beta 1	uint32_t GetGraphDataRecords (GraphDataRecord* result, uint32_t resultlen, int graphid)	
added in 4.00 Beta 1	uint32_t GetGraphIcons (GraphIcon* result, uint32_t resultlen, int graphid)	
added in 4.00 Beta 1	bool GetGraphTitle (const int graphid, GraphTextInfo* title)	
added in 4.00 Beta 1	uint32_t GetGraphType (const int graphid)	
added in 4.00 Beta 1	bool GetGraphXAxis (const int graphid, GraphAxisInfo* axis)	
added in 4.00 Beta 1	bool GetGraphYAxis (const int graphid, GraphAxisInfo* axis)	
added in 4.00 Beta 3	bool GetInstalledWeaponMod (UniverseID weaponid, UIWeaponMod* weaponmod)	

added in 4.21 Beta 1 (VR: 4.20)	const char* GetItemTraderTypeByWare (const char* wareid) <i>Allows to reverse lookup the correct item trader entitytype for a certain ware.</i>	
added in 4.00 Beta 2	JumpDriveDetails GetJumpDriveDetails () <i>Retrieves details about the player's jumpdrive. Only valid, if the player has a jumpdrive atm.</i>	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	uint32_t GetLanguages (LanguageInfo* result, uint32_t resultlen) Returns information about available languages.	only functional in X Rebirth VR Edition
added in 4.00 RC 2	bool GetLargeHUDMenusOption ()	
added in 3.50 Beta 2	LoadingHint GetLoadingHint (const long excludeid)	
	const char* GetLocalizedInteractiveNotificationKey ()	
	const char* GetLocalizedText (const long pageid, const long textid, const char*const defaultvalue)	
	int GetMainMissiontargetPOSID ()	 UNSUPPORTED. Not designed to be used by mods.
	UniverseID GetMapComponentBelowCamera (UniverseID holomapid) <i>Returns the component in the center of the specified holomap.</i>	
added in 4.00 Beta 1	bool GetMapPositionOnEcliptic (UniverseID holomapid, UIPosRot* position, bool showposition) <i>Returns the position on the ecliptic that the mouse cursor is currently pointing to.</i>	
	const char* GetMapShortName (UniverseID componentid) <i>Returns the short name of the specified space.</i>	
	MessageDetails GetMessageDetails (const uint32_t messageid)	
deprecated in 4.00 Beta 3	DEPRECATED MissionDetails GetMissionIDDetails (uint64_t missionid) <i>Returns details of the specified mission.</i>	 DEPRECATED - use GetMissionIDDetails2() instead
added in 4.00 Beta 3	MissionDetails2 GetMissionIDDetails2 (uint64_t missionid) <i>Returns details of the specified mission.</i>	Note: In addition to the details returned by GetMissionIDDetails() this function returns whether the mission has an objective or not.
deprecated in 4.31 Beta 1	DEPRECATED MissionObjective GetMissionIDObjective (uint64_t missionid, const char* fontname, uint32_t fontsize, uint32_t textwidth) <i>Returns the mission objective of the specified mission.</i>	 DEPRECATED - use GetMissionIDObjective2() instead
added in 4.31 Beta 1	MissionObjective GetMissionIDObjective2 (uint64_t missionid) <i>Returns the mission objective of the specified mission.</i>	
	UniverseID GetMissionIDObjectiveTarget (uint64_t missionid, size_t targetIndex) <i>Returns the target of the current objective of the specified mission.</i>	
deprecated in 4.31 Beta 1	DEPRECATED MissionObjectiveStep GetMissionObjectiveStep (uint64_t missionid, size_t objectiveIndex, const char* fontname, uint32_t fontsize, uint32_t textwidth) <i>Returns the objective of the specified mission and objectiveindex.</i>	 DEPRECATED - use GetMissionObjectiveStep2() instead
added in 4.31 Beta 1	MissionObjectiveStep GetMissionObjectiveStep2 (uint64_t missionid, size_t objectiveIndex) <i>Returns the objective of the specified mission and objectiveindex.</i>	

added in 4.00 Beta 1	const char* GetMouseOverText (const int widgetid)	
added in 4.00 Beta 1	bool GetMouseOverTextOption ()	
added in 4.22 Beta 2 (VR: 4.22)	float GetMouseVRSensitivityPitch ()	only available in XR VR
added in 4.22 Beta 2 (VR: 4.22)	float GetMouseVRSensitivityYaw ()	only available in XR VR
	int GetNotificationID (const size_t num)	
added in 4.00 Beta 6	uint32_t GetNPCAccountStatistics (UIAccountStatData* result, size_t resultlen, UniverseID entityid, double starttime, double endtime) <i>Stores the npc account statistics in result. Returns the number of entries in result.</i>	
added in 4.10 RC 1	uint32_t GetNumAllMissiles (UniverseID defensibleid) <i>Returns the number of missile types an object can use.</i>	
added in 4.00 Beta 6	uint32_t GetNumAmmoStorage (UniverseID defensibleid, const char* ammotype) <i>Returns the number of entries in the ammo storage.</i>	
added in 4.00 Beta 5	uint32_t GetNumCargoStatistics (UniverseID containerorspaceid, double starttime, double endtime, size_t numdatapoints) <i>Returns the number of entries in the cargo statistics.</i>	
added in 4.00 Beta 5	uint32_t GetNumContainedKnownUnreadSpaces (UniverseID spaceid) <i>Returns the number of unread known spaces contained in the specified space.</i>	
added in 4.00 Beta 1	uint32_t GetNumCurrentMissionOffers (bool showninbbs) <i>Returns the number of current mission offers.</i>	See also GetCurrentMissionOffers ()
added in 4.00 Beta 1	uint32_t GetNumGraphDataRecords (int graphid);	
added in 4.00 Beta 1	uint32_t GetNumGraphIcons (int graphid)	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	uint32_t GetNumLanguages () <i>Returns the number of available languages.</i>	only functional in X Rebirth VR Edition
	size_t GetNumNotifications () <i>Returns the number of current notifications.</i>	
added in 4.21 Beta 1 (VR: 4.20)	size_t GetNumPrimaryWeapons () <i>Returns the number of primary weapons in the player ship.</i>	Before 4.22 Beta 2 (VR: 4.22) this returned the number of weapons in the currently controlled player object.
added in 4.21 Beta 1 (VR: 4.20)	size_t GetNumSecondaryWeapons () <i>Returns the number of secondary weapons in the player ship.</i>	Before 4.22 Beta 2 (VR: 4.22) this returned the number of weapons in the currently controlled player object.

added in 4.00 Beta 4	uint32_t GetNumStoredUnits (UniverseID defensibleid, const char* cat, bool virtualammo) <i>Returns the number of stored units on the specified defensible and category.</i>	
added in 4.00 Beta 2	uint32_t GetNumTradeOfferStatistics (UniverseID containerorspaceid, double starttime, double endtime, size_t numdatapoints) <i>Returns the number of entries in the trade offer statistics.</i>	
added in 4.00 Beta 3	uint32_t GetNumTradeWares () <i>Returns the number of possible trade wares in the game.</i>	
added in 4.00 Beta 5	uint32_t GetNumUnreadLibraryEntries (const char* libraryid) <i>Returns the number of unread library entries in the specified library.</i>	
added in 4.00 Beta 3	uint32_t GetNumWeaponMods ()	
	FactionDetails GetOwnerDetails (UniverseID componentid)	
	UniverseID GetParentComponent (UniverseID componentid) <i>Returns the parent of the specified component.</i>	
	UniverseID GetPickedMapComponent (UniverseID holomapid) <i>Returns the currently picked component on the specified holomap.</i>	
added in 4.21 Beta 1 (VR: 4.20)	UniverseID GetPlayerComputerID () <i>Returns the player computer entity (Betty).</i>	
added in 4.21 Beta 1 (VR: 4.20)	UniverseID GetPlayerControlledShipID () <i>Returns the ship or drone the player is currently controlling.</i>	
added in 4.21 Beta 1 (VR: 4.20)	UniverseID GetPlayerCoPilotID () <i>Returns the player's co-pilot entity (Yisha).</i>	
added in 4.21 Beta 1 (VR: 4.20)	UniverseID GetPlayerID () <i>Returns the player entity.</i>	
added in 4.21 Beta 1 (VR: 4.20)	UniverseID GetPlayerObjectID () <i>Returns the object context of the player (ship or station).</i>	
added in 4.21 Beta 1 (VR: 4.20)	UniverseID GetPlayerShipID () <i>Returns the ship context of the player if the player is on a ship.</i>	
added in 3.50 RC 1	int GetPlayerShipNumFreeActorSlots () <i>Returns the number of free actor slots on the player ship.</i>	
added in 4.21 Beta 1 (VR: 4.20)	UniverseID GetPlayerZoneID () <i>Returns the zone context of the player.</i>	
	const char* GetPlayerZoneName ()	
	POIDetails GetPOIDetails (const uint64_t poiid)	

added in 4.31 Beta 1	int GetPriorityMissiontargetPOSID()	 UNSUPPORTED. Not designed to be used by mods.
added in 4.21 Beta 1 (VR: 4.20)	const char* GetQuickMenuHotkey (uint32_t num) <i>Returns the hotkey assigned to the button of the specified number in the quick menu. num ranging from 1-10.</i>	
added in 4.21 Beta 1 (VR: 4.20) deprecated in 4.31 Beta 1	DEPRECATED float GetRectangleTargetElementExtends (const int posid) <i>Returns the rectangular extents for the specified target element.</i>	 DEPRECATED - use GetRectangleTargetElementExtends() instead
added in 4.31 Beta 1	float GetRectangleTargetElementExtends (const int posid) <i>Returns the rectangular extents for the specified target element.</i>	
changed in 3.50 Beta 3	int RelationDetails GetRelationStatus (const UniverseID componentid)	
added in 4.21 Beta 1 (VR: 4.20)	PosRot GetRelativeAimOffset (const UniverseID componentid) <i>Returns the position and rotation of the aim-at-offset (in worldspace) to shoot at in order to hit the specified object.</i>	
deprecated in 4.21 Beta 1 (VR: 4.20)	DEPRECATED Position2D GetRelativeAimPosition (const UniverseID componentid, const uint32_t iconsizeonscreen, const uint32_t iconsizeoffscreen) <i>Returns the 2D screen position to aim-at in order to hit the specified object. The on-/off-screen iconsizes specify the icon size of targetelements which icons have on screen.</i>	 DEPRECATED - use GetRelativeAimScreenPosition() instead Note: As of 4.00 Beta 1 this function returns the aim position relative to the object the player currently controls (aka: player ship or remote controlled drone/torpedo). Before, it always determined the aim position based on the player ship.
added in 4.21 Beta 1 (VR: 4.20)	Position2D GetRelativeAimScreenPosition (const UniverseID componentid, const uint32_t iconsizeonscreen, const uint32_t iconsizeoffscreen) <i>Returns the 2D screen position to aim-at in order to hit the specified object. The on-/off-screen iconsizes specify the icon size of targetelements which icons have on screen.</i>	
added in 4.10 Beta 1	ResolutionInfo GetRenderResolutionOption() <i>Returns the current render resolution;</i>	
	uint32_t GetRenderTargetAlpha (const int rendertargetid)	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	int32_t GetRequestedLanguage() <i>Returns the ID of the currently requested language (the actual setting)</i>	only functional in X Rebirth VR Edition
added in 4.00 Beta 1	bool GetSavesCompressedOption() <i>Returns whether save game compression is active.</i>	
	float GetScanProgress (UniverseID componentid)	
	const char* GetShipSize (const UniverseID componentid)	
	SofttargetDetails GetSofttarget()	

added in 4.00 Beta 6	SupplyBudget GetSupplyBudget (UniverseID containerid) <i>Returns the supply budget information of the specified container.</i>	
added in 4.21 Beta 1 (VR: 4.20)	bool GetTableAlertEffectColor (const int tableid, Color* color) <i>Retrieves alert effect color of a table if available. Returns false if there is no alert effect.</i>	
added in 4.21 Beta 1 (VR: 4.20)	PosRot GetTargetElementOffset (const int posid) <i>Retrieves the position and rotation values of the target element in worldspace.</i>	
added in 4.31 Beta 1	uint32_t GetTextHeight (const char*const text, const char*const fontname, const uint32_t fontsize, const uint32_t wordwrapwidth) <i>Returns the height in pixels of the specified text. If wordwrapwidth is set to 0, the text height will be calculated without performing any word wrapping.</i>	
added in 4.31 Beta 1	uint32_t GetTextWidth (const char*const text, const char*const fontname, const uint32_t fontsize) <i>Returns the width in pixels of the specified text.</i>	
added in 4.00 Beta 4	bool GetTradeMemoryBestBuyOffers (TradeID* result, uint32_t resultlen, UniverseID spaceid, UniverseID traderid, TradeID* uiselloffers, uint32_t uisellofferslen, UIWareAmount* uicargospace, uint32_t uicargospacecen) <i>Creates a list of trade ids of best buy offers matching the list of sell offers, if return value is true. Space is the range in which trades are searched; Trader is the tradeship used, defaults to the playership; Cargospace defines the possible amount of all wares in the sell offer list.</i>	
added in 4.00 Beta 2	uint32_t GetTradeOfferStatistics (UITradeOfferStat* result, uint32_t resultlen, size_t numdatapoints) <i>Stores the trade offer statistics in result. Returns the number of results.</i>	Only works if GetNumTradeOfferStatistics() was called before.
added in 4.00 Beta 3	uint32_t GetTradeWares (const char** result, uint32_t resultlen) <i>Stores all possible trade wares in the game in result. Returns number of results.</i>	
deprecated in 4.30 Beta 1 added in 4.10 Beta 1	DEPRECATED float GetUIScale () <i>Returns the UI scale factor.</i>	 DEPRECATED - use GetUIScale2() instead
added in 4.30 Beta 1	float GetUIScale2 (const bool scaleWithResolution) <i>Returns the UI scale factor depending on the scaling mode set via the scaleWithResolutionparameter. If scaleWithResolution is set to true, it's assumed that the to be scaled UI element should scale alongside larger game resolutions (i.e. the element will become/appear larger on higher resolutions). If set to false, it's assumed that the to be scaled UI element should remain in its current set size (i.e. appear smaller on higher resolutions).</i>	
added in 4.30 Beta 1	float GetUIScaleFactor () <i>Returns the plain UI scale factor.</i>	
added in 4.30 Beta 1	uint32_t GetUIScaledFontSize (uint32_t fontsize) <i>Returns the ui scaled fontsize of the specified one.</i>	


added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	int GetVRControllerAutoHide() <i>Gets the currently defined setting for hiding VR controllers in the VR display if using a different controller.</i>	
added in 4.21 Beta 1 (VR: 4.20)	int GetVRVivePointerHand() <i>Retrieves the controller currently defined as the active pointing device. (1 == Right, 0 == Left. For use with the Vive controller.)</i>	only functional in X Rebirth VR Edition
added in 4.22 Beta 2 (VR: 4.20 Hotfix 1)	float GetVRViveTouchpadLockTime() <i>Retrieves the currently defined latency for the touchpads on vive controllers.</i>	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	int GetVRWindowMode() <i>Gets the currently defined render mode on the monitor while VR headset is on.</i>	
added in 4.21 Beta 1 (VR: 4.20)	ResolutionInfo GetWindowSize() <i>Returns the current window size.</i>	
added in 4.00 Beta 3	uint32_t GetWeaponMods (UIWeaponMod* result, uint32_t resultlen)	
added in 4.00 Beta 2	UniverseID GetZoneAt (UniverseID sectorid, UIPosRot* uioffset) <i>Returns the zone at the given sector position. Can create a temporary zone if needed.</i>	
	ZoneDetails GetZoneDetails (const UniverseID zoneid)	
added in 3.52 RC 2	bool HasCustomConversation (UniverseID entityid)	
added in 4.00 Beta 6	bool HasEntityMoneyLogEntries (UniverseID entityid) <i>Returns whether the specified entity has a money log.</i>	
	bool HasDockedPassiveDrones()	
added 4.00 Beta 2	bool HasJumpDrive() <i>Indicates whether the player ship has a jumpdrive atm.</i>	
	bool HasPassiveDroneAbilities()	
added 4.00 Beta 3	bool HasRemoteControl() <i>Indicates whether the player has remote control atm (aka: controlling a drone or a torpedo).</i>	
added in 4.10 Beta 1	bool HasSavegame() <i>Indicates whether there's at least one savegame in the savegame directory.</i> <i>Note: doesn't necessarily have to be a valid/loadable savegame. Also note that unless <code>IsSaveListLoadingComplete()</code> indicates that the processing is done, <code>HasSavegame()</code> can return false, even though there would be a savegame in the savegame directory.</i>	
added 4.00 Beta 2	bool HasSeta() <i>Indicates whether the player has SETA.</i>	
	bool HasSmalltalkOption (const UniverseID componentid)	



added in 4.00 Beta 3	bool InstallWeaponMod (uint32_t weaponmodid)	
added in 4.00 Beta 6 (Hotfix 1)	bool IsAppStoreVersion ()	Note: This function is of little interest to mods. It's only required for some internal checks.
	bool IsAsteroid (const UniverseID componentid) <i>Indicates whether the specified component is an asteroid.</i>	
added in 4.30 RC 3 (VR: 4.30 Hotfix 2)	bool IsAutotargetingActive () <i>Indicates whether the autotarget function of the target system is active atm.</i>	
	bool IsBomb (const UniverseID componentid) <i>Indicates whether the specified component is a bomb.</i>	
	bool IsBuyOffer (UniverseID tradeofferdockid) <i>Checks whether the specified tradeofferdock is a buy offer.</i>	
added in 4.00 Beta 2	bool IsCheckBoxActive (const int checkboxid)	
added in 4.00 Beta 2	bool IsCheckBoxChecked (const int checkboxid)	
	bool IsComponentOperational (UniverseID componentid) <i>Checks whether the specified component is operational.</i>	
	bool IsCrate (const UniverseID componentid) <i>Indicates whether the specified component is a crate.</i>	
added in 4.00 Beta 5	bool IsDemoVersion () <i>Indicates whether we are running the demo version atm.</i>	Note: This function is of little interest to mods, since mods are disabled in the demo version.
	bool IsDestructible (const UniverseID componentid) <i>Indicates whether the specified component is a destructible.</i>	
	bool IsDetailMonitorFullscreenMode ()	
added in 4.00 Beta 1	bool IsDroneMode () <i>Indicates whether the UI is currently in drone-mode.</i>	
	bool IsEntity (const UniverseID componentid) <i>Returns true, if the specified component is an entity.</i>	
	bool IsErrorLogActive () <i>Indicates whether error logging is currently active.</i>	
	bool IsExternalViewActive () <i>Determined whether the game is currently in external view mode.</i>	
added in 4.00 Beta 2	bool IsExternalTargetMode () <i>Determines whether the game is currently in external target view mode.</i>	
added in 4.21 Beta 1 (VR: 4.20)	bool IsFPSEnabled () <i>Indicates whether the FPS counter is enabled atm.</i>	
added in 4.10 Beta 4	bool IsFrameUsingMiniWidgetSystem (const int frameid)	

added in 4.00 Beta 1	bool IsGamePaused() <i>Indicates whether the game is paused atm.</i>	
	bool IsGateTransition (const UniverseID zone1id, const UniverseID zone2id)	
added in 4.22 Beta 2 (VR: 4.22)	bool IsGestureSteeringActive()	only functional in XR VR
added in 4.31 Beta 1	bool IsHeadTrackingActive() <i>Indicates whether head tracking (f.e. TrackIR) is active atm.</i>	
added in 4.00 Beta 2	bool IsHUDActive() <i>Indicates whether the HUD is currently active.</i>	
	bool IsInfoUnlockedForPlayer (UniverseID componentid, const char* infostring) <i>Checks whether the specified info of the component is unlocked for the player.</i>	
	bool IsInHackRange (const UniverseID componentid)	
	bool IsInScanRange (const UniverseID componentid)	
added in 4.00 Beta 5	bool IsKnownItemRead (const char* libraryid, const char* itemid) <i>Returns whether the specified library entry is read.</i>	
added in 4.00 Beta 5	bool IsKnownRead (UniverseID componentid) <i>Returns whether the specified component is read.</i>	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	bool IsLanguageValid() <i>Returns whether the currently used language is a valid setting or defaulted to English.</i>	only functional in X Rebirth VR Edition
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	bool IsLanguageSettingEnabled() <i>Returns whether the language setting should be displayed in the menu.</i>	only functional in X Rebirth VR Edition
added in 4.31 Beta 1	bool IsLockbox (UniverseID targetid) Returns true, if the specified target is a lockbox.	
added in 4.22 Beta 2 (VR: 4.22)	bool IsMouseVRPointerAllowed() <i>Returns whether the mouse VR pointer is allowed to activate.</i>	only functional in X Rebirth VR Edition
added in 4.31 Beta 1	bool IsNavBeacon (UniverseID targetid) <i>Returns true, if the specified target is a nav beacon.</i>	
added in 4.00 Beta 6	bool IsPlayerCameraTargetViewPossible (UniverseID targetid, bool force) <i>Can the player camera be set to the specified target's external view?</i>	
added in 4.00 Beta 5	bool IsPlayerValid() <i>Check whether the player has a valid context (e.g. they are not in game over mode)</i>	

added in 4.00 Beta 1	bool IsPointingWithinAimingRange() <i>Indicates whether the cursor is pointing within the aiming range of the weapons.</i>	
added in 4.10 Beta 1	bool IsSaveListLoadingComplete() <i>Indicates whether processing the savegame directory is complete and a following call to GetSaveList2() will return a complete list (rather than a temporary list).</i>	
	bool IsSellOffer(UniverseID componentid) <i>Checks whether the specified tradeofferdock is a sell offer.</i>	
added in 4.00 Beta 2	bool IsSetaActive() <i>Indicates whether Seta is active atm.</i>	
	bool IsShip(const UniverseID componentid)	
	bool IsSofttargetManagerActive();	
added in 4.31 Beta 1	bool IsStartmenu() <i>Indicates whether the UI script is running inside the startmenu or not.</i>	
	bool IsStation(const UniverseID componentid)	
	bool IsSurfaceElement(const UniverseID componentid)	
	bool IsSwitchable(const UniverseID componentid)	
added in 4.30 RC 1 (VR: 4.30)	bool IsTargetable(const uint32_t messageid) <i>Indicates whether the target element (specified via its underlying messageid) is targetable atm.</i>	
	bool IsTargetInPlayerActivePrimaryWeaponRange(const UniverseID componentid) <i>Indicates whether the current target is within the weapon range of the current active primary weapon.</i>	Note: As of 4.00 Beta 1 this function takes the object the player controls (aka: player ship or remote controlled drone/torpedo) at the time of the call when determining whether the component is in weapon range. Before, it always determined whether the component is in weapon range to the playership.
added in 4.21 Beta 1	bool IsThrottleBidirectional() <i>Checks whether bidirectional throttle is enabled.</i>	
added in 3.50 Beta 2	bool IsValidComponent(const UniverseID componentid);	
added in 4.21 Beta 1 (VR: 4.20)	bool IsVROculusTouchActive() <i>Indicates if an Oculus Touch controller is active.</i>	only functional in X Rebirth VR Edition
added in 4.21 Beta 1 (VR: 4.20)	bool IsVRMode() <i>Indicates whether the VR mode is active. Note: If IsVRMode() returns true, it implies that IsVRVersion() returns true as well.</i>	only functional in X Rebirth VR Edition
added in 4.10 Beta 1	bool IsVRVersion() <i>Indicates whether we are running the VR version.</i>	
added in 4.10 Beta 4	bool IsVRPointerActive() <i>Indicates whether the in-game VR-pointer is displayed atm.</i>	
added in 4.21 Beta 1 (VR: 4.20)	bool IsVRViveControllerActive() <i>Indicates if a Vive controller is active.</i>	only functional in X Rebirth VR Edition
	bool IsWalkInAnimationRunning()	

added in 4.21 Beta 1 (VR: 4.20)	void LeaveInteractMenu() Ends the interact menu interaction mode.	
	bool MarkIndicatorShown (const UniverseID componentid);	
	void NotifyDisplayNotification (const int notificationid); <i>Notifies the game engine about that the given notification is currently displayed.</i>	
	void NotifyErrorLogDispatching() <i>Notifies the game engine that error log dispatching/processing starts.</i>	
	void NotifyErrorLogDispatchingDone() <i>Notifies the game engine that error log dispatching/processing was done.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void NotifyInteractMenuHidden() <i>Notifies the game engine that the interact menu is hidden.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void NotifyInteractMenuShown() <i>Notifies the game engine that the interact menu is shown.</i>	
added in 4.00 Beta 5	void OpenWebBrowser (const char* url)	 Designed for internal use only.
added in 4.21 Beta 1 (VR: 4.20)	bool QuickDock() <i>Quick access to player docking (if the player currently has targeted a dock). Return value indicates success.</i>	
added in 4.21 Beta 1 (VR: 4.20)	bool QuickMenuAccess (const char* menu);	 Designed for internal use only. Use Lua function: StartConversationFromMenu() instead.
added in 4.21 Beta 1 (VR: 4.20)	bool QuickUndock() <i>Quick access to player undocking (if the player is currently docked). Return value indicates success.</i>	
added in 4.00 Beta 5	void ReadKnownItem (const char* libraryid, const char* itemid, bool read); <i>Sets the specified library item to read.</i>	
	void RecallDrones() <i>Orders all drones to return to the player ship.</i>	
	void ReenableErrorLog() <i>Special function designed to be called to reenable the error log if it was automatically turned off to prevent potential deadlock conditions.</i>	
added in 4.31 Beta 1	void ReleaseInteractionDescriptor (int32_t id) <i>Given a interactionDescriptorID, the function frees the associated data.</i>	
added in 4.10 Beta 1	void ReloadSaveList() <i>Triggers processing of the savegame directory, starting the asynchronous loading of all savegame data in that directory. Used in combination with GetSaveList2().</i>	
	void RemoveFill (const UniverseID componentid) <i>Removes the fill effect of the specified component.</i>	

deprecated in 4.00 Beta 3	DEPRECATED void RemoveHoloMap (UniverseID holomapid) <i>Removes the holomap.</i>	 DEPRECATED - use RemoveHoloMap2() instead Note: As of 4.00 Beta 3 only a single holomap is supported. Therefore the specified ID is ignored.
added in 4.00 Beta 3	void RemoveHoloMap2 () <i>Removes the holomap.</i>	
added in 4.00 Beta 3	bool RemoveInstalledWeaponMod (UniverseID weaponid)	
	void RemoveOutline (const UniverseID componentid)	
added in 4.00 Beta 3	void RemoveTradeWare (UniverseID containerid, const char* wareid) <i>Removes the specified ware as a trade ware from the specified container.</i>	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	void RequestLanguageChange (int32_t id) <i>Sets the requested language to id.</i>	only functional in X Rebirth VR Edition NOTE: This only works if IsLanguageSettingEnabled() returned true
added in 4.00 Beta 5	void RevealEncyclopedia () <i>Completely reveals the encyclopedia.</i>	NOTE: This will show some entries that normally are not discoverable by the player. Needs cheats enabled.
added in 4.00 Beta 6	void ScheduleSupplyUpdate (UniverseID componentid) <i>Schedules an update of the supply calculations on the specified component.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void SetAdaptiveSamplingOption (float value)	
	void SetConfigSetting (const char*const setting, const bool value)	
added in 4.00 Beta 3	void SetContainerWareIsBuyable (UniverseID containerid, const char* wareid, bool allowed) <i>Sets whether the specified ware is buyable on the specified container according to the value of allowed.</i>	
added in 4.00 Beta 3	void SetContainerWareIsSellable (UniverseID containerid, const char* wareid, bool allowed) <i>Sets whether the specified ware is sellable on the specified container according to the value of allowed.</i>	
	void SetCrosshairMessage (const uint32_t messageid)	
	void SetFill (const UniverseID componentid, const uint8_t red, const uint8_t green, const uint8_t blue, const float alpha, const bool animated, const float minalpha, const float maxalpha, const float transitiontime)	
added in 4.10 Beta 1 / changed in 4.10 Beta 6	void SetForceShootingAtCursorOption (bool value) <i>Sets whether shooting is forced to always aim towards the cursor position (rather than also towards the crosshair).</i>	Renamed in 4.10 Beta 6; was ToggleForceShootingAtCursorOption(void) before
	void SetFXAAOption (uint32_t fxaa) <i>Sets the FXAA setting.</i>	
	void SetHighlightMapComponent (UniverseID holomapid, UniverseID componentid, bool resetplayerpan) <i>Sets the highlighted component on the specified holomap.</i>	

added in 4.00 Beta 5	void SetKnownRead (UniverseID componentid, bool read) <i>Sets the specified component to read.</i>	
added in 3.60 Beta 4	void SetKnownTo (UniverseID componentid, const char* factionid)	
added in 4.00 RC 2	void SetLargeHUDMenuOption (bool value)	
added in 4.00 Beta 3	void SetLootMagnetActive (bool active) <i>(De)Activates the playership's loot magnet.</i>	
deprecated in 4.31 Beta 1 changed in 4.00 Beta 2	bool SetMapRenderTarget (const char*const rendertargettexture, const UniverseID referencecomponent, const bool mapmode) <i>Displays a radar on the given texture in the specified mode. Referencecomponent will be the highlighted one in the radar. 0 indicates no highlighted object.</i>	 DEPRECATED - use SetRadarRenderTarget() instead In 4.00 Beta 2 the function creates a map which is focused around the object the player controls atm (so the drone in case it's called while the player controls a drone). Pre 4.00 Beta 2 behavior was always to focus the player ship.
deprecated in 4.31 Beta 1 added in 4.00 Beta 2	bool SetMapRenderTargetOnTarget (const char*const rendertargettexture, const UniverseID focuscomponent, const UniverseID referencecomponent, const bool mapmode) <i>Same as SetMapRenderTarget but allows specifying the component which the radar will be focused on.</i>	 DEPRECATED - use SetRadarRenderTargetOnTarget() instead
added in 4.00 Beta 1	void SetMouseOverTextOption (bool value)	
added in 4.22 Beta 2 (VR: 4.22)	void SetMouseVRPointerAllowed (bool newsetting) <i>Allows or disallows Mouse VR Pointer activation.</i>	only available in XR VR
added in 4.22 Beta 2 (VR: 4.22)	void SetMouseVRSensitivityPitch (float value)	only available in XR VR
added in 4.22 Beta 2 (VR: 4.22)	void SetMouseVRSensitivityYaw (float value)	only available in XR VR
	void SetOutline (const UniverseID componentid, const uint8_t red, const uint8_t green, const uint8_t blue, const bool animated)	
added in 4.00 Beta 3	void SetPlayerCameraCockpitView (bool force) <i>Set the player camera to the normal cockpit view.</i>	
added in 4.00 Beta 3	void SetPlayerCameraExternalView (bool force) <i>Set the player camera to the external view of the playership.</i>	
added in 4.00 Beta 3	void SetPlayerCameraTargetView (UniverseID targetid, bool force) <i>Set the player camera to the specified target's external view.</i>	
added in 4.31 Beta 1	bool SetRadarRenderTarget (const char*const rendertargettexture, const UniverseID referencecomponent, const bool mapmode) <i>Displays a radar on the given texture in the specified mode. Referencecomponent will be the highlighted one in the radar. 0 indicates no highlighted object.</i>	

added in 4.31 Beta 1	bool SetMapRenderTargetOnTarget (const char*const rendertargettexture, const UniverseID focuscomponent, const UniverseID referencecomponent, const bool mapmode) <i>Same as SetMapRenderTarget but allows specifying the component which the radar will be focused on.</i>	
added in 4.00 Beta 1	void SetSavesCompressedOption (bool value) <i>Sets the save game compression option.</i>	
added in 3.60 Beta 2	bool SetSofttarget (const UniverseID componentid) <i>Sets the current target to the specified one. The return value indicates whether setting succeeded.</i>	
added in 4.21 Beta 1 (VR: 4.20)	bool SetSofttargetByMissionOffer (const uint64_t missionofferid) <i>Sets the current target to the one representing the given mission offer. The return value indicates whether setting succeeded.</i>	
added in 4.21 Beta 1	void SetThrottleBidirectional (bool newsetting) <i>Enables/disables bidirectional throttle.</i>	
deprecated in 4.30 Beta 1 added in 4.10	DEPRECATED void SetUIScale (float scale) <i>Sets the uiscale to the given value.</i>	 DEPRECATED - use SetUIScaleFactor() instead
added in 4.30 Beta 1	void SetUIScaleFactor (float scale) <i>Sets the uiscale to the given value.</i>	
	void ShowUniverseMap (UniverseID holomapid, UniverseID componentid, bool resetplayerzoom, int overridezoom) <i>Set the specified holomap to display the map of given component.</i>	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	void SetVRControllerAutoHide (int newmode) <i>Sets newmode as the currently defined setting for hiding VR controllers in the VR display if using a different controller.</i>	
added in 4.22 Beta 2 (VR: 4.20 Hotfix 1)	void SetVRVivePointerHand (int hand) <i>Set the designated controller as active pointing device. (1 == Right, 0 == Left. For use with the Vive controller.)</i>	only functional in X Rebirth VR Edition
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	void SetVRViveTouchpadLockTime (float newtime) <i>Set newtime as the latency time for the touch pads on vive controllers.</i>	
added in 4.21 Beta 1 (VR: 4.20 Hotfix 1)	void SetVRWindowMode (int mode) <i>Set new mode for rendering on the monitor while VR headset is on.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void StartIntroAnimation () <i>Start an intro animation.</i>	
	void StartPanMap (UniverseID holomapid) <i>Signal the specified holomap that panning started.</i>	
	void StartRotateMap (UniverseID holomapid) <i>Signal the specified holomap that rotation started.</i>	

	void StopPanMap (UniverseID holomapid) <i>Signal the specified holomap that panning stopped.</i>	
	void StopRotateMap (UniverseID holomapid) <i>Signal the specified holomap that rotation stopped.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void StartStartMenuBGMusic () <i>Start the startmenu background music, cancelling all previous start menu music.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void StopStartMenuBGMusic () <i>Stops all previous start menu music.</i>	
added in 4.00 Beta 1	void ToggleAutoPilot () <i>Toggle the autopilot on the current mission target.</i>	
added in 4.00 Beta 3	void ToggleEventMonitorDetails () <i>Toggle the display of details on the eventmonitor.</i>	
added in 4.00 Beta 3	void ToggleFlightAssist () <i>Toggle the flight assist mode.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void ToggleLootMagnet () <i>Toggle the loot magnet.</i>	
added in 4.00 Beta 3	void ToggleMouseSteeringMode () <i>Toggle mouse steering mode.</i>	
added in 4.00 Beta 3	void ToggleRadarMode () <i>Toggle between the radar modes.</i>	
added in 4.21 Beta 1 (VR: 4.20)	void ToggleSeta () <i>Toggle Seta on/off.</i>	 Designed for internal use only. Use the MD action: ToggleTimeWarpAction instead.
added in 4.00 Beta 6	void UpdateBlackboardSupplyOverrides (UniverseID entityid, SupplyOverride* overrides, uint32_t numoverrides) <i>Updates the blackboard variables for supply overrides with the given data.</i>	
added in 4.31 Beta 1	void UnsetRadarRenderTarget () <i>Resets the render target for the radar/map display.</i>	
	void ZoomMap (UniverseID holomapid, float zoomstep) <i>Zoom the specified holomap.</i>	

DEPRECATED