```
Scriptname ActiveMagicEffect Hidden

; Add an inventory event filter to this effect. Item added/removed events matching the
; specified form (or in the specified form list) will now be let through.
Function AddInventoryEventFilter(Form akFilter) native

; Dispel this effect
Function Dispel() native

; Get the base MagicEffect this active effect is using
MagicEffect Function GetBaseObject() native

; Get the actor that cast this spell
Actor Function GetCasterActor() native

; Get the actor this spell is targeting (is attached to)
Actor Function GetTargetActor() native

; Register for the specified animation event from the specified object - returns true if it successfully registered
bool Function RegisterForAnimationEvent(ObjectReference akSender, string asEventName) native

; Register for LOS gain and lost events between the viewer and the target
; A loss or gain event will be sent immediately, depending on whether or not the viewer is already looking at the target or not
; If the viewer is not the player, the target must be another actor
Function RegisterForLOS(Actor akViewer, ObjectReference akTarget) native

; Register for only the first LOS gain event between the viewer and the target
; If the viewer is already looking at the target, an event will be received almost immediately
; If the viewer is not the player, the target must be another actor
Function RegisterForSingleLOSGain(Actor akViewer, ObjectReference akTarget) native

; Register for only the first LOS lost event between the viewer and the target
; If the viewer is already not looking at the target, an event will be received almost immediately
; If the viewer is not the player, the target must be another actor
Function RegisterForSingleLOSLost(Actor akViewer, ObjectReference akTarget) native

; Register for a single OnUpdate event, in afInterval seconds. All scripts attached to this magic effect will get the update events
; Of course, this means you don't need to call UnregisterForUpdate()
; If you find yourself doing this:
; Event OnUpdate()
;     UnregisterForUpdate()
;     {Do some stuff}
; endEvent
; Then you should use RegisterForSingleUpdate instead
Function RegisterForSingleUpdate(float afInterval) native

; Registers this magic effect to receive events when the player sleeps and wakes up
Function RegisterForSleep() native

; Registers this alias to receive events when tracked stats are updated
Function RegisterForTrackedStatsEvent() native

; Register for OnUpdate events, every X seconds, where X is the interval. All scripts attached to this magic effect will get the update events
Function RegisterForUpdate(float afInterval) native

; Register for OnUpdateGameTime events, every X hours of game time, where X is the interval. All scripts attached to this magic effect will get the update events
Function RegisterForUpdateGameTime(float afInterval) native

; Register for a single OnUpdateGameTime event, in afInterval hours of game time. All scripts attached to this magic effect will get the update events
Function RegisterForSingleUpdateGameTime(float afInterval) native
```

```

; Remove all inventory event filters from this effect - all item added/removed events will now be received
Function RemoveAllInventoryEventFilters() native

; Remove an inventory event filter from this effect. Item added/removed events matching the
; specified form (or in the specified form list) will no longer be let through.
Function RemoveInventoryEventFilter(Form akFilter) native

; Turns on profiling for this specific object and all scripts attached to it - setting doesn't persist across saves
; Will do nothing on release console builds, and if the Papyrus:bEnableProfiling ini setting is off
Function StartObjectProfiling() native

; Turns off profiling for this specific object and all scripts attached to it - setting doesn't persist across saves
; Will do nothing on release console builds, and if the Papyrus:bEnableProfiling ini setting is off
Function StopObjectProfiling() native

; Unregister for any LOS events between the viewer and target
Function UnregisterForLOS(Actor akViewer, ObjectReference akTarget) native

; Unregister for the specified animation event from the specified object
Function UnregisterForAnimationEvent(ObjectReference akSender, string asEventName) native

; Unregisters this magic effect to receive events when the player sleeps and wakes up
Function UnregisterForSleep() native

; Unregisters this magic effect from receiving events when tracked stats are updated
Function UnregisterForTrackedStatsEvent() native

; Unregister for OnUpdate events, all attached scripts will stop getting update events
Function UnregisterForUpdate() native

; Unregister for OnUpdateGameTime events, all attached scripts will stop getting update game time events
Function UnregisterForUpdateGameTime() native

; Animation event, sent when an object we are listening to hits one of the events we are listening for
Event OnAnimationEvent(ObjectReference akSource, string asEventName)
EndEvent

; Event sent when you have been unregistered from receiving an animation event because the target
; object's animation graph has been unloaded
Event OnAnimationEventUnregistered(ObjectReference akSource, string asEventName)
EndEvent

; Event received when this effect is first started (OnInit may not have been run yet!)
Event OnEffectStart(Actor akTarget, Actor akCaster)
EndEvent

; Event received when this effect is finished (effect may already be deleted, calling
; functions on this effect will fail)
Event OnEffectFinish(Actor akTarget, Actor akCaster)
EndEvent

; LOS event, sent whenever the viewer first sees the target (after registering)
Event OnGainLOS(Actor akViewer, ObjectReference akTarget)
EndEvent

; Lost LOS event, sent whenever the viewer first loses sight of the target (after registering)
Event OnLostLOS(Actor akViewer, ObjectReference akTarget)
EndEvent
```

```

; Received when the player sleeps. Start and desired end time are in game time days (after registering)
Event OnSleepStart(float afSleepStartTime, float afDesiredSleepEndTime)
EndEvent

; Received when the player stops sleeping - whether naturally or interrupted (after registering)
Event OnSleepStop(bool abInterrupted)
EndEvent

; Event received when a tracked stat is updated for the player
Event OnTrackedStatsEvent(string arStatName, int aiStatValue)
EndEvent

; Update event, sent every X seconds while this magic effect is registered for them
Event OnUpdate()
EndEvent

; Update event, sent every X hours of game time while this magic effect is registered for them
Event OnUpdateGameTime()
EndEvent

; The following events are received from the actor this effect is attached to:

; Event received when this reference is activated
Event OnActivate(ObjectReference akActionRef)
EndEvent

; Event received when this object has moved to an attached cell from a detached one
Event OnAttachedToCell()
EndEvent

; Event received when this object's parent cell is attached
Event OnCellAttach()
EndEvent

; Event received when this object's parent cell is detached
Event OnCellDetach()
EndEvent

; Event received when every object in this object's parent cell is loaded (TODO: Find restrictions)
Event OnCellLoad()
EndEvent

; Event received when this object is closed
Event OnClose(ObjectReference akActionRef)
EndEvent

; Event received when this object enters, exits, or changes containers
Event OnContainerChanged(ObjectReference akNewContainer, ObjectReference akOldContainer)
EndEvent

; Event received when this reference's destruction stage has changed
Event OnDestructionStageChanged(int aiOldStage, int aiCurrentStage)
EndEvent

; Event recieved when this object moves to a detached cell from an attached one
Event OnDetachedFromCell()
EndEvent

; Event received when this object is equipped by an actor
Event OnEquipped(Actor akActor)
EndEvent

; Event received when this object is grabbed by the player
Event OnGrab()
```

```
EndEvent

; Event received when this object is hit by a source (weapon, spell, explosion) or
projectile attack
Event OnHit(ObjectReference akAggressor, Form akSource, Projectile akProjectile, bool
abPowerAttack, bool abSneakAttack, bool abBashAttack, bool abHitBlocked)
EndEvent

; Event received when an item is added to this object's inventory. If the item is a
persistant reference, akItemReference will
; point at it - otherwise the parameter will be None
Event OnItemAdded(Form akBaseItem, int aiItemCount, ObjectReference akItemReference,
ObjectReference akSourceContainer)
EndEvent

; Event received when an item is removed from this object's inventory. If the item is a
persistant reference, akItemReference
; will point at it - otherwise the parameter will be None
Event OnItemRemoved(Form akBaseItem, int aiItemCount, ObjectReference akItemReference,
ObjectReference akDestContainer)
EndEvent

; Event recieved when this object is completely loaded - will be fired every time this
object is loaded
Event OnLoad()
EndEvent

; Event received when the lock on this object changes
Event OnLockStateChanged()
EndEvent

; Event received when a magic affect is being applied to this object
Event OnMagicEffectApply(ObjectReference akCaster, MagicEffect akEffect)
EndEvent

; Event received when this object is opened
Event OnOpen(ObjectReference akActionRef)
EndEvent

; Event received when this actor finishes changing its race
Event OnRaceSwitchComplete()
EndEvent

; Event received when this object, if a book, is read
Event OnRead()
EndEvent

; Event received when this object is released by the player
Event OnRelease()
EndEvent

; Event received when this reference is reset
Event OnReset()
EndEvent

; Event received when this reference is sold by an actor
Event OnSell(Actor akSeller)
EndEvent

; Event received when a spell is cast by this object
Event OnSpellCast(Form akSpell)
EndEvent

; Event received when translation is complete (from a call to TranslateTo)
Event OnTranslationComplete()
EndEvent

; Event received when translation is aborted (from a call to StopTranslateTo)
Event OnTranslationFailed()
```

```
EndEvent

; Event recieved when this reference hits a target
Event OnTrapHit(ObjectReference akTarget, float afXVel, float afYVel, float afZVel,
float afXPos, float afYPos, float afZPos, \
    int aeMaterial, bool abInitialHit, int aeMotionType)
EndEvent

; Event recieved when this starts hitting a target
Event OnTrapHitStart(ObjectReference akTarget, float afXVel, float afYVel, float afZVel,
float afXPos, float afYPos, float afZPos, \
    int aeMaterial, bool abInitialHit, int aeMotionType)
EndEvent

; Event recieved when this stops hitting a target
Event OnTrapHitStop(ObjectReference akTarget)
EndEvent

; Event received when a this trigger is tripped
Event OnTrigger(ObjectReference akActionRef)
EndEvent

; Event received when this trigger volume is entered
Event OnTriggerEnter(ObjectReference akActionRef)
EndEvent

; Event received when this trigger volume is left
Event OnTriggerLeave(ObjectReference akActionRef)
EndEvent

; Event received when this object is unequipped by an actor
Event OnUnequipped(Actor akActor)
EndEvent

; Event recieved when this object is being unloaded - will be fired every time this
object is unloaded
Event OnUnload()
EndEvent

; Event that is triggered when this actor's combat state against the target changes
; State is as follows:
; 0 - not in combat
; 1 - in combat
; 2 - searching
Event OnCombatStateChanged(Actor akTarget, int aeCombatState)
EndEvent

; Event that is triggered when this actor sits in the furniture
Event OnSit(ObjectReference akFurniture)
EndEvent

; Event that is triggered when this actor leaves the furniture
Event OnGetUp(ObjectReference akFurniture)
EndEvent

; Event that is triggered when this actor finishes dying
Event OnDeath(Actor akKiller)
EndEvent

; Event that is triggered when this actor begins dying
Event OnDying(Actor akKiller)
EndEvent

; Event that is triggered when this actor changes from one location to another
Event OnLocationChange(Location akOldLoc, Location akNewLoc)
EndEvent

; Received when the lycanthropy state of this actor changes (when
SendLycanthropyStateChanged is called)
```

```
Event OnLycanthropyStateChanged(bool abIsWerewolf)
EndEvent

; Event received when this actor equips something - akReference may be None if object is not persistent
Event OnObjectEquipped(Form akBaseObject, ObjectReference akReference)
EndEvent

; Event received when this actor unequips something - akReference may be None if object is not persistent
Event OnObjectUnequipped(Form akBaseObject, ObjectReference akReference)
EndEvent

; Event received when this actor starts a new package
Event OnPackageStart(Package akNewPackage)
EndEvent

; Event received when this actor's package changes
Event OnPackageChange(Package akOldPackage)
EndEvent

; Event received when this actor's package ends
Event OnPackageEnd(Package akOldPackage)
EndEvent

; Event received when this object's Ward is hit by a spell
Event OnWardHit(ObjectReference akCaster, Spell akSpell, int aiStatus)
EndEvent

; Received when the player fires a bow. akWeapon will be a bow, akAmmo is the ammo or None,
; afPower will be 1.0 for a full-power shot, less for a dud, and abSunGazing will be true if the player is looking at the sun.
Event OnPlayerBowShot(Weapon akWeapon, Ammo akAmmo, float afPower, bool abSunGazing)
EndEvent

; Received when the player finishes fast travel, gives the duration of game time the travel took
Event OnPlayerFastTravelEnd(float afTravelGameTimeHours)
EndEvent

; Received immediately after the player has loaded a save game. A good time to check for additional content.
Event OnPlayerLoadGame()
EndEvent

; Received when StartVampireFeed is called on an actor
Event OnVampireFeed(Actor akTarget)
EndEvent

; Received when the vampirism state of this actor changes (when SendVampirismStateChanged is called)
Event OnVampirismStateChanged(bool abIsVampire)
EndEvent


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; Additional useful effect information
float Function GetDuration() native
float Function GetTimeElapsed() native

; Registers for OnKeyDown and OnKeyUp events for the given keycode.
Function RegisterForKey(int keyCode) native
Function UnregisterForKey(int keyCode) native
Function UnregisterForAllKeys() native

Event OnKeyDown(int keyCode)
EndEvent

```

```
Event OnKeyUp(int keyCode, float holdTime)
EndEvent

; Registers for OnControlDown and OnControlUp events for the given control.
; For a list of valid controls, see Input.psc.
Function RegisterForControl(string control) native
Function UnregisterForControl(string control) native
Function UnregisterForAllControls() native

Event OnControlDown(string control)
EndEvent

Event OnControlUp(string control, float holdTime)
EndEvent

; Registers for OnMenuOpen and OnMenuClose events for the given menu.
; Registrations have to be refreshed after each game load.
; For a list of valid menu names, see UI.psc.
Function RegisterForMenu(string menuName) native
Function UnregisterForMenu(string menuName) native
Function UnregisterForAllMenus() native

Event OnMenuOpen(string menuName)
endEvent

Event OnMenuClose(string menuName)
endEvent

; Registers a custom event callback for given event name.
; Registrations have to be refreshed after each game load.
;
;   Examples:
;       RegisterForModEvent("myCustomEvent", "MyModEventCallback")
;
;   Event signature of custom event callbacks:
;       Event MyModEventCallback(string eventName, string strArg, float numArg, Form
sender)
;       endEvent
;
Function RegisterForModEvent(string eventName, string callbackName) native
Function UnregisterForModEvent(string eventName) native
Function UnregisterForAllModEvents() native

; Sends custom event with given generic parameters.
Function SendModEvent(string eventName, string strArg = "", float numArg = 0.0) native

; See Form.psc
Function RegisterForCameraState() native
Function UnregisterForCameraState() native

Event OnPlayerCameraState(int oldState, int newState)
EndEvent

; See Form.psc
Function RegisterForCrosshairRef() native
Function UnregisterForCrosshairRef() native

Event OnCrosshairRefChange(ObjectReference ref)
EndEvent

; See Form.psc
Function RegisterForActorAction(int actionType) native
Function UnregisterForActorAction(int actionType) native

Event OnActorAction(int actionType, Actor akActor, Form source, int slot)
EndEvent

; Registers the script for when a QueueNiNodeUpdate is called
Function RegisterForNiNodeUpdate() native
Function UnregisterForNiNodeUpdate() native

Event OnNiNodeUpdate(ObjectReference akActor)
EndEvent

; returns the magnitude of the active effect
float Function GetMagnitude() native   :: Add a newline between files
Scriptname Actor extends ObjectReference Hidden

; Relationship functions use the following values:
; 4 - Lover
; 3 - Ally
; 2 - Confidant
; 1 - Friend
; 0 - Acquaintance
; -1 - Rival
; -2 - Foe
; -3 - Enemy
; -4 - Archnemesis

; DEPRECATED - use MakePlayerFriend() instead
; replacement for ModFavorPoints
; if iFavorPoints > 0, will setRelationshipRank to 1 if 0
; otherwise, won't do anything
Function ModFavorPoints(int iFavorPoints = 1)
    if iFavorPoints > 0
        MakePlayerFriend()
    else
;       debug.trace(self + " ModFavorPoints called with negative param. NO EFFECT.")
    endif
endFunction

; also DEPRECATED
Function ModFavorPointsWithGlobal(GlobalVariable FavorPointsGlobal)
    ModFavorPoints(FavorPointsGlobal.GetValueInt())
endFunction

;this function will make an actor a friend of the player if allowed
Function MakePlayerFriend()
    ActorBase myBase = GetActorBase()
    if myBase.IsUnique()
        if GetRelationshipRank(Game.GetPlayer())== 0
;           debug.trace(self + " MakePlayerFriend called on neutral actor - changed to 
FRIEND.")
            SetRelationshipRank(Game.GetPlayer(), 1)
        else
;           debug.trace(self + " MakePlayerFriend called on non-neutral actor - NO 
EFFECT.")
        endif
    else
;       debug.trace(self + " MakePlayerFriend called on non-Unique actor. NO EFFECT.")
    endif
endFunction

; Adds the specified perk to this actor
Function AddPerk(Perk akPerk) native

; Adds the specified shout to this actor - returns true on success
bool Function AddShout(Shout akShout) native

; Adds the specified spell to this actor - returns true on success
bool Function AddSpell(Spell akSpell, bool abVerbose=true) native

; Sets this a essential actors ability to talk when in a bleedout state
Function AllowBleedoutDialogue(bool abCanTalk ) native

; overrides the race flag on an actor and determines if he can talk to the player in 
dialogue menu
Function AllowPCDialogue(bool abTalk) native
```

```
; Attaches an "ash pile" to this actor, placing it at this actor's location and using the specified
; base object (or leveled item list) to represent the pile. If None is passed, it will use the
; default ash pile object
Function AttachAshPile(Form akAshPileBase = None) native

; Can this actor fly here?
bool Function CanFlyHere() native

; Clears this actor's arrested state
Function ClearArrested() native

; Clears any expression override on the actor
Function ClearExpressionOverride() native

; Clears this actor's extra arrows 3D
Function ClearExtraArrows() native

; Remove the obligation to use a particular marker when this actor has to land.
Function ClearForcedLandingMarker()
  SetForcedLandingMarker( None )
endFunction

; Clear any keep offset from actor settings
Function ClearKeepOffsetFromActor() native

; Clears this actor's look at target
Function ClearLookAt() native

; Damages the specified actor value
Function DamageActorValue(string asValueName, float afDamage) native

; Alias for DamageActorValue - damages the specified actor value
Function DamageAV(string asValueName, float afDamage)
  DamageActorValue(asValueName, afDamage)
EndFunction

; Initiates a dismount.
bool Function Dismount() native

; Dispel all spells from this actor
Function DispelAllSpells() native

; Dispel a spell from this actor
bool Function DispelSpell( Spell akSpell ) native

; Apply a spell to a target in combat
Function DoCombatSpellApply( Spell akSpell, ObjectReference akTarget ) native

; Enables or disable's this actor's AI
Function EnableAI(bool abEnable = true) native

; End the Deferred Kill state. This must only be called if StartDeferredKill was called first.
Function EndDeferredKill() native

; Forces this actor to equip the specified item, preventing removal if requested
Function EquipItem(Form akItem, bool abPreventRemoval = false, bool abSilent = false) native

; Forces this actor to equip the specified shout
Function EquipShout(Shout akShout) native

; Forces this actor to equip the specified spell. The casting source can be:
; 0 - Left hand
; 1 - Right hand
Function EquipSpell(Spell akSpell, int aiSource) native
```

```

; Forces the AI to re-evaluate its package stack
Function EvaluatePackage() native

; Force the specified actor value to a specified value
Function ForceActorValue(string asValueName, float afNewValue) native

; Alias for ForceActorValue - force the specified actor value to a specified value
Function ForceAV(string asValueName, float afNewValue)
  ForceActorValue(asValueName, afNewValue)
EndFunction

;returns the ActorBase
ActorBase function GetActorBase()
    return GetBaseObject() as ActorBase
endFunction

; Gets the specified actor value - returns 0 and logs an error if the value is unknown
float Function GetActorValue(string asValueName) native

; Gets the specified actor value's max, taking into account buffs/debuffs
float Function GetActorValueMax(string asValueName) native

; Gets the specified actor value as a percentage of its max value - from 0 to 1
float Function GetActorValuePercentage(string asValueName) native

; Alias for GetActorValue - retrives the specified actor value
float Function GetAV(string asValueName)
  return GetActorValue(asValueName)
EndFunction

; Alias of GetActorValueMax - retrives actor value's max, taking into account
buffs/debuffs
float Function GetAVMax(string asValueName)
    return GetActorValueMax(asValueName)
EndFunction

; Alias for GetActorValuePercentage - gets the actor value as a percent of max
float Function GetAVPercentage(string asValueName)
  return GetActorValuePercentage(asValueName)
EndFunction

; Gets the base value of the specified actor value - returns 0 and logs an error if the
value is unknown
float Function GetBaseActorValue(string asValueName) native

; Alias for GetBaseActorValue - retrieves the specified actor value's base value
float Function GetBaseAV(string asValueName)
  return GetBaseActorValue(asValueName)
EndFunction

; Obtains how much it would cost to bribe this actor
int Function GetBribeAmount() native

; Get the faction this actor reports crimes to
Faction Function GetCrimeFaction() native

; Gets this actor's current combat state
int Function GetCombatState() native

; Gets this actor's current combat target
Actor Function GetCombatTarget() native

; Gets this actor's current AI package
Package Function GetCurrentPackage() native

; Gets this actor's current dialogue target
Actor Function GetDialogueTarget() native

```

```
; Obtain the armor currently equipped in the specified slot
Armor Function GetEquippedArmorInSlot(int aiSlot) native

; Obtains the item quipped in the specified hand (0 - Left hand, 1 - Right hand)
; Return values are:
; -1 - Error
; 0 - Nothing
; 1 - One-handed sword
; 2 - One-handed dagger
; 3 - One-handed axe
; 4 - One-handed mace
; 5 - Two-handed sword
; 6 - Two-handed axe
; 7 - Bow
; 8 - Staff
; 9 - Magic spell
; 10 - Shield
; 11 - Torch
int Function GetEquippedItemType(int aiHand) native

; Gets this actor's currently equipped shout
Shout Function GetEquippedShout() native

; Gets this actor's currently equipped weapon
; false - Default - Right Hand
; true - Left Hand
Weapon Function GetEquippedWeapon(bool abLeftHand = false) native

; Gets this actor's currently equipped shield
Armor Function GetEquippedShield() native

; Gets the spell currently equipped in the specified source
; 0 - Left Hand
; 1 - Right Hand
; 2 - Other
; 3 - Instant
Spell Function GetEquippedSpell(int aiSource) native

; Obtains this actor's rank with the specified faction - returns -1 if the actor is not
a member
int Function GetFactionRank(Faction akFaction) native

; Obtains this actor's faction-based reaction to the other actor
; 0 - Neutral
; 1 - Enemy
; 2 - Ally
; 3 - Friend
int Function GetFactionReaction(Actor akOther) native

; Obtains this actor's current flight state
; 0 - Not flying
; 1 - Taking off
; 2 - Cruising
; 3 - Hovering
; 4 - Landing
int Function GetFlyingState() native

; Get the ref at which this actor is obliged to land, if one is set (or none, if not).
ObjectReference Function GetForcedLandingMarker() native

; Retrieves the amount of gold this actor has
int Function GetGoldAmount() native

; Gets this actor's highest relationship rank - returns 0 if they have no relationships
int Function GetHighestRelationshipRank() native

; Returns this actor's killer - or None if this actor is still alive
Actor Function GetKiller() native

```

```
; Returns this actor's current level.
int Function GetLevel() native

; Returns this actor's current light level.
float Function GetLightLevel() native

; Gets this actor's highest relationship rank - returns 0 if they have no relationships
int Function GetLowestRelationshipRank() native

; Obtains a leveled actor's "fake" base (the one generated by the game when the
; actor is leveled. This differs from GetActorBase which will return the editor base
; object)
ActorBase Function GetLeveledActorBase() native

; Queries whether this actor has no bleedout recovery flag set.
bool Function GetNoBleedoutRecovery() native

; Queries whether this actor receives player input
bool Function GetPlayerControls() native

; Returns this actor's race
Race Function GetRace() native

; Obtains the relationship rank between this actor and another
int Function GetRelationshipRank(Actor akOther) native

; Obtains this actor's sit state, which is one of the following:
; 0 - Not sitting
; 2 - Not sitting, wants to sit
; 3 - Sitting
; 4 - Sitting, wants to stand
int Function GetSitState() native

; Obtains this actor's sleep state, which is one of the following:
; 0 - Not sleeping
; 2 - Not sleeping, wants to sleep
; 3 - Sleeping
; 4 - Sleeping, wants to wake
int Function GetSleepState() native

; Gets the voice recovery timer from the actor
float Function GetVoiceRecoveryTime() native

; Gets the total "warmth rating" for this actor
float Function GetWarmthRating() native

; Checks to see if this actor has the specified association with the other actor - or
anyone (if no actor is passed)
bool Function HasAssociation(AssociationType akAssociation, Actor akOther = None) native

; Checks to see if this actor has a family relationship with the other actor - or anyone
(if no actor is passed)
bool Function HasFamilyRelationship(Actor akOther = None) native

; Sees if this actor has line-of-sight to another object. Only the player can check LOS
to a non-actor
bool Function HasLOS(ObjectReference akOther) native

; Checks to see if this actor is currently being affected by the given Magic Effect
bool Function HasMagicEffect(MagicEffect akEffect) native

; Checks to see if this actor is currently being affected by a Magic Effect with the
given Keyword
bool Function HasMagicEffectWithKeyword(Keyword akKeyword) native

; Checks to see if this actor has a parent relationship with the other actor
bool Function HasParentRelationship(Actor akOther) native

; Checks to see if this actor has the given Perk
```

```
bool Function HasPerk(Perk akPerk) native

; Checks to see if this actor has the given Spell or Shout
bool Function HasSpell(Form akForm) native

; Returns if this actor is alarmed or not
bool Function IsAlarmed() native

; Returns if this actor is alerted or not
bool Function IsAlerted() native

; Is this actor allowed to fly?
bool Function IsAllowedToFly() native

; Is this actor currently arrested?
bool Function IsArrested() native

; Is this actor currently arresting his target? (Must be a guard and alarmed)
bool Function IsArrestingTarget() native

; Is the actor being ridden?
bool Function IsBeingRidden() native

; Is this actor currently bleeding out?
bool Function IsBleedingOut() native

; Queries whether this actor has player bribe flag set.
bool Function IsBribed() native

; Is this actor a child?
bool Function IsChild() native

; Is this actor a commanded by another?
bool Function IsCommandedActor() native

; Returns if this actor is dead or not
bool Function IsDead() native

; Returns if this actor is detected by the other one
bool Function IsDetectedBy(Actor akOther) native

; Is this actor doing a favor for the player?
bool Function IsDoingFavor() native

; Returns if the specified object is equipped on this actor
bool Function IsEquipped(Form akItem) native

; Is this actor essential?
bool Function IsEssential() native

; Returns if this actor is flying or not
bool Function IsFlying() native

; Returns if this actor is a guard or not
bool Function IsGuard() native

; Is this actor flagged as a ghost?
bool Function IsGhost() native

; Is this actor hostile to another actor?
bool Function IsHostileToActor(Actor akActor) native

; Returns if this actor is currently in combat
bool Function IsInCombat() native

; Checks to see if this actor is a member of the specified faction
bool Function IsInFaction(Faction akFaction) native

; Returns if this actor is in a kill move or not
```

```
bool Function IsInKillMove() native

; Queries whether this actor has player intimidated flag set.
bool Function IsIntimidated() native

; Is the actor on a mount?
bool Function IsOnMount() native

; Is the actor over-encumbered?
bool Function IsOverEncumbered() native

; Checks to see if this actor the last ridden horse of the player
bool Function IsPlayersLastRiddenHorse() native

; Is this actor currently a teammate of the player?
bool Function IsPlayerTeammate() native

; Is this actor currently running?
bool Function IsRunning() native

; Is this actor currently sneaking?
bool Function IsSneaking() native

; Is this actor currently sprinting?
bool Function IsSprinting() native

; Is this actor trespassing?
bool Function IsTrespassing() native

; Is this actor unconscious?
bool Function IsUnconscious() native

; Does this actor have his weapon and/or magic drawn?
bool Function IsWeaponDrawn() native

; Sets the actor to a mode where it will keep a given offset from another actor
Function KeepOffsetFromActor(Actor arTarget, float afOffsetX, float afOffsetY, float
afOffsetZ, float afOffsetAngleX = 0.0,  float afOffsetAngleY = 0.0,  float
afOffsetAngleZ = 0.0, float afCatchUpRadius = 20.0, float afFollowRadius = 5.0) native

; Kills this actor with the killer being the guilty party
Function Kill(Actor akKiller = None) native

; Kills this actor even if essential
Function KillEssential(Actor akKiller = None)
    ActorBase akActorBase = GetBaseObject() as ActorBase
    if akActorBase.IsUnique()
        akActorBase.SetEssential(0)
    endif
    Kill(akKiller)
endFunction

; Kills this actor without a kill event with the killer being the guilty party
Function KillSilent(Actor akKiller = None) native

; Modifies the specified actor value
Function ModActorValue(string asValueName, float afAmount) native

; Alias for ModActorValue - modifies the specified actor value
Function ModAV(string asValueName, float afAmount)
  ModActorValue(asValueName, afAmount)
EndFunction

; Modifies this actor's rank in the faction
Function ModFactionRank(Faction akFaction, int aiMod) native

; Pop this actor to the initial location for a package. Mainly for use on
; disabled actors, since they would normally start at their editor locations.
Function MoveToPackageLocation( ) native
```

```

; Opens this actor's inventory, as if you were pick-pocketing them. Only works on
teammates, or anyone if forced.
Function OpenInventory(bool abForceOpen = false) native

; Make the actor path to a reference, latent version
; Note: this method doesn't return until the goal is reached or pathing
; failed or was interrupted (by another request for instance)
bool Function PathToReference(ObjectReference aTarget, float afWalkRunPercent) native

; Send an idle to the actor to load in and play.
bool Function PlayIdle(Idle akIdle) native

; Send an idle to the actor to play, overriding its target with the specified reference
bool Function PlayIdleWithTarget(Idle akIdle, ObjectReference akTarget) native

; Send an event to the subgraphs of an actor.
Function PlaySubGraphAnimation(string asEventName) native

; Removes this actor from the specified faction
Function RemoveFromFaction(Faction akFaction) native

; Removes this actor from all factions
Function RemoveFromAllFactions() native

; Removes the specified perk from this actor
Function RemovePerk(Perk akPerk) native

; Removes the specified shout from this actor - returns true on success
bool Function RemoveShout(Shout akShout) native

; Removes the specified spell from this actor - returns true on success
bool Function RemoveSpell(Spell akSpell) native

; Resets this actor's health and limb state
Function ResetHealthAndLimbs() native

; Restores damage done to the actor value (up to 0 damage)
Function RestoreActorValue(string asValueName, float afAmount) native

; Resurrects this actor
Function Resurrect() native

; Alias for RestoreActorValue - restores damage done to the actor value
Function RestoreAV(string asValueName, float afAmount)
  RestoreActorValue(asValueName, afAmount)
EndFunction

; Has this actor behave as if assaulted
Function SendAssaultAlarm() native

; Tell anyone who cares that the lycanthropy state of this actor has changed
Function SendLycanthropyStateChanged(bool abIsWerewolf) native

; Has this actor behave as if they caught the target trespassing
Function SendTrespassAlarm(Actor akCriminal) native

; Tell anyone who cares that the vampirism state of this actor has changed
Function SendVampirismStateChanged(bool abIsVampire) native

; Sets the specified actor value
Function SetActorValue(string asValueName, float afValue) native

; Sets the actor in an alerted state
Function SetAlert(bool abAlerted = true) native

; Sets whether this actor is allowed to fly or not - if not, will land the actor
Function SetAllowFlying(bool abAllowed = true) native

```

```
; Sets whether this actor is allowed to fly or not - if not, will land the actor
Function SetAllowFlyingEx(bool abAllowed = true, bool abAllowCrash = true, bool abAllowSearch = false) native

; Sets this actor's alpha - with an optional fade to that alpha
; The alpha will be clamped between 0 and 1
Function SetAlpha(float afTargetAlpha, bool abFade = false) native

; Sets this actor to be attacked by all other actors on sight
Function SetAttackActorOnSight(bool abAttackOnSight = true) native

; Alias for SetActorValue - sets the specified actor value
Function SetAV(string asValueName, float afValue)
  SetActorValue(asValueName, afValue)
EndFunction

; Flags/unflags this actor as bribed by the player
Function SetBribed(bool abBribe = true) native

; Sets the faction this actor reports crimes to
Function SetCrimeFaction(Faction akFaction) native

; Sets this actor's critical stage, which is one of the following (properties below also match this)
; 0 - None
; 1 - Goo start
; 2 - Goo end
; 3 - Disintegrate start
; 4 - Disintegrate end
Function SetCriticalStage(int aiStage) native

; Flag this actor as currently doing a favor for the player
Function SetDoingFavor(bool abDoingFavor = true) native

; Sets this actor as "don't move" or not
Function SetDontMove(bool abDontMove = true) native

; Sets an expression to override any other expression other systems may give this actor.
;                            7 - Mood Neutral
; 0 - Dialogue Anger         8 - Mood Anger      15 - Combat Anger
; 1 - Dialogue Fear          9 - Mood Fear       16 - Combat Shout
; 2 - Dialogue Happy         10 - Mood Happy
; 3 - Dialogue Sad           11 - Mood Sad
; 4 - Dialogue Surprise      12 - Mood Surprise
; 5 - Dialogue Puzzled       13 - Mood Puzzled
; 6 - Dialogue Disgusted     14 - Mood Disgusted
; aiStrength is from 0 to 100 (percent)
Function SetExpressionOverride(int aiMood, int aiStrength = 100) native

;forces the eye texture for this actor to the give texture set
Function SetEyeTexture(TextureSet akNewTexture) native

; Sets this actor's rank with the specified faction
Function SetFactionRank(Faction akFaction, int aiRank) native

; Set a specific marker as the place at which this actor must land from flight.
; params:
; - aMarker:  The ObjectReference to set as this actor's landing marker
Function SetForcedLandingMarker( ObjectReference aMarker ) native

; Flags/unflags this actor as a ghost
Function SetGhost(bool abIsGhost = true) native

; Adds this actor to a faction at rank 0 if they aren't already in it
Function AddToFaction(Faction akFaction)
    if (!IsInFaction(akFaction))
        SetFactionRank(akFaction, 0)
    endif
EndFunction
```

```
; Turns on/off headtracking on this actor
Function SetHeadTracking(bool abEnable = true) native

; Flags/unflags this actor as intimidated by the player
Function SetIntimidated(bool abIntimidate = true) native

; Sets this actor's head tracking target, optionally forcing it as their pathing look-at target
Function SetLookAt(ObjectReference akTarget, bool abPathingLookAt = false) native

; Set the no bleedout recovery flag on this actor
Function SetNoBleedoutRecovery(bool abAllowed) native

; Sets this actor to not effect the detection level on the stealth meter if he is not hostile to the player
Function SetNotShowOnStealthMeter(bool abNotShow) native

; Sets the actors outfit and makes him wear it
Function SetOutfit( Outfit akOutfit, bool abSleepOutfit = false ) native

; Set/reset whether player input being sent to the actor
Function SetPlayerControls(bool abControls) native

; Sets the player as resisting arrest from this actor's faction
Function SetPlayerResistingArrest() native

; Sets or clears this actor as a teammate of the player
; abCanDoFavor - OPTIONAL default is true the teammate can do favors
Function SetPlayerTeammate(bool abTeammate = true, bool abCanDoFavor=true) native

; Sets the actors race
; akRace - OPTIONAL (Def=None) New race for this actor. Default, no race, to switch back to the original race.
Function SetRace( Race akRace = None ) native

; Sets the relationship rank between this actor and another (See GetRelationshipRank for the ranks)
Function SetRelationshipRank(Actor akOther, int aiRank) native

; Sets this actor as restrained or not
Function SetRestrained(bool abRestrained = true) native

; Set a variable on all of an actor's subgraphs
Function SetSubGraphFloatVariable(string asVariableName, float afValue) native

; Sets this actor as unconscious or not
Function SetUnconscious(bool abUnconscious = true) native

; Attach the actor to (or detach it from) a horse, cart, or other vehicle.
; akVehicle is the vehicle ref.  To detach the actor from its current vehicle, set akVehicle to None (or to the Actor itself).
Function SetVehicle( ObjectReference akVehicle ) native

; Sets the voice recovery timer on the actor
; afTime is recovery time in seconds
Function SetVoiceRecoveryTime( float afTime ) native

; Opens the Barter menu
Function ShowBarterMenu() native

; Opens the Gift menu
; Params:
; - abGivingGift: True if we're giving a gift to this Actor, false if the player is taking a gift from this Actor
; - apFilterList: OPTIONAL (Def=None) -- If present, this form list is used to filter the item list.  Only items
; that match keywords / items in the list will get shown
; - abShowStolenItems: OPTIONAL (Def=false) -- If true, stolen items are shown
; - abUseFavorPoints: OPTIONAL (Def=true) -- If true, favor points are added / subtracted with each transaction.  If false, FPs aren't used at all.
; Returns: The number of favor points spent / gained while in the menu.
int Function ShowGiftMenu( bool abGivingGift, FormList apFilterList = None, bool abShowStolenItems = false, bool abUseFavorPoints = true ) native

; Starts Cannibal with the target
Function StartCannibal(Actor akTarget) native

; Starts combat with the target
Function StartCombat(Actor akTarget) native

; Start the Deferred Kill state. Be sure to call EndDeferredKill or the actor will be invulnerable.
Function StartDeferredKill() native

; Starts vampire feed with the target
Function StartVampireFeed(Actor akTarget) native

; Removes this actor from combat
Function StopCombat() native

; Stops all combat and alarms against this actor
Function StopCombatAlarm() native

; Returns whether the actor can trap the soul of the given actor.
bool Function TrapSoul(Actor akTarget) native

; Unequips the all items from this actor
Function UnequipAll() native

; Unequips the specified item from this actor
Function UnequipItem(Form akItem, bool abPreventEquip = false, bool abSilent = false) native

; Unequips the all items in this slot for the actor
Function UnequipItemSlot(int aiSlot) native

; Forces this actor to unequip the specified shout
Function UnequipShout(Shout akShout) native

; Forces this actor to unequip the specified spell. The casting source can be:
; 0 - Left hand
; 1 - Right hand
Function UnequipSpell(Spell akSpell, int aiSource) native

; This actor will unlock all the doors that he qualifies for ownership in his current parentcell
Function UnLockOwnedDoorsInCell() native

; Returns whether intimidate will succeed against this actor or not
bool Function WillIntimidateSucceed() native

; Returns whether anything the actor is wearing has the specified keyword
bool Function WornHasKeyword(Keyword akKeyword) native

; Makes this actor start sneaking
Function StartSneaking() native

; Makes this actor draw his weapon
Function DrawWeapon() native

; Event that is triggered when this actor's combat state against the target changes
; State is as follows:
; 0 - not in combat
; 1 - in combat
; 2 - searching
Event OnCombatStateChanged(Actor akTarget, int aeCombatState)
EndEvent
```

```

; Event that is triggered when this actor sits in the furniture
Event OnSit(ObjectReference akFurniture)
EndEvent

; Event that is triggered when this actor leaves the furniture
Event OnGetUp(ObjectReference akFurniture)
EndEvent

; Event that is triggered when this actor finishes dying
Event OnDeath(Actor akKiller)
EndEvent

; Event that is triggered when this actor begins to die
Event OnDying(Actor akKiller)
EndEvent

; Event received when an actor enters bleedout.
Event OnEnterBleedout()
EndEvent

; Event that is triggered when this actor changes from one location to another
Event OnLocationChange(Location akOldLoc, Location akNewLoc)
EndEvent

; Received when the lycanthropy state of this actor changes (when
SendLycanthropyStateChanged is called)
Event OnLycanthropyStateChanged(bool abIsWerewolf)
EndEvent

; Event received when this actor equips something - akReference may be None if object is
not persistent
Event OnObjectEquipped(Form akBaseObject, ObjectReference akReference)
EndEvent

; Event received when this actor unequips something - akReference may be None if object
is not persistent
Event OnObjectUnequipped(Form akBaseObject, ObjectReference akReference)
EndEvent

; Event received when this actor starts a new package
Event OnPackageStart(Package akNewPackage)
EndEvent

; Event received when this actor's package changes
Event OnPackageChange(Package akOldPackage)
EndEvent

; Event received when this actor's package ends
Event OnPackageEnd(Package akOldPackage)
EndEvent

; Event received when this actor finishes changing its race
Event OnRaceSwitchComplete()
EndEvent

; Received when the player fires a bow. akWeapon will be a bow, akAmmo is the ammo or
None,
; afPower will be 1.0 for a full-power shot, less for a dud, and abSunGazing will be
true if the player is looking at the sun.
Event OnPlayerBowShot(Weapon akWeapon, Ammo akAmmo, float afPower, bool abSunGazing)
EndEvent


; Received immediately after the player has loaded a save game. A good time to check for
additional content.
Event OnPlayerLoadGame()
EndEvent

```

```
; Received when the player finishes fast travel, gives the duration of game time the travel took
Event OnPlayerFastTravelEnd(float afTravelGameTimeHours)
EndEvent

; Received when StartVampireFeed is called on an actor
Event OnVampireFeed(Actor akTarget)
EndEvent

; Received when the vampirism state of this actor changes (when SendVampirismStateChanged is called)
Event OnVampirismStateChanged(bool abIsVampire)
EndEvent

; Set of read-only properties to essentually make a fake enum for critical stages
int Property CritStage_None = 0 AutoReadOnly
int Property CritStage_GooStart = 1 AutoReadOnly
int Property CritStage_GooEnd = 2 AutoReadOnly
int Property CritStage_DisintegrateStart = 3 AutoReadOnly
int Property CritStage_DisintegrateEnd = 4 AutoReadOnly

; **** For Debugging Movement Animations (not in release builds) ****
; Forces the movement direction on the actor
; afXAngle, afYAngle and afZAngle are in degrees
Function ForceMovementDirection(float afXAngle = 0.0, float afYAngle = 0.0, float afZAngle = 0.0) native

; Forces the movement speed on the actor
; afSpeedMult is a speed multiplier based on the current max speeds
; - 0 -> 1 Scales between 0 and the Walk speed
; - 1 -> 2 Scales between Walk speed and Run Speed
; - 2 and above is a multiplier of the run speed (less 1.0 since Run is 2.0)
Function ForceMovementSpeed(float afSpeedMult) native

; Forces the movement rotation speed on the actor
; Each component of the rotation speed is a multiplier following these rules:
; - 0 -> 1 Scales between 0 and the Walk speed
; - 1 -> 2 Scales between Walk speed and Run Speed
; - 2 and above is a multiplier of the run speed (less 1.0 since Run is 2.0)
Function ForceMovementRotationSpeed(float afXMult = 0.0, float afYMult = 0.0, float afZMult = 0.0) native

; Ramps the movement direction on the actor to the passed in value over the passed in time
; afXAngle, afYAngle and afZAngle are in degrees
; afRampTime is in seconds
Function ForceMovementDirectionRamp(float afXAngle = 0.0, float afYAngle = 0.0, float afZAngle = 0.0, float afRampTime = 0.1) native

; Ramps the movement speed on the actor to the passed in value over the passed in time
; afSpeedMult is a speed multiplier based on the current max speeds
; - 0 -> 1 Scales between 0 and the Walk speed
; - 1 -> 2 Scales between Walk speed and Run Speed
; - 2 and above is a multiplier of the run speed (less 1.0 since Run is 2.0)
; afRampTime is in seconds
Function ForceMovementSpeedRamp(float afSpeedMult, float afRampTime = 0.1) native

; Ramps the movement rotation speed on the actor to the passed in value over the passed in time
; Each component of the rotation speed is a multiplier following these rules:
; - 0 -> 1 Scales between 0 and the Walk speed
; - 1 -> 2 Scales between Walk speed and Run Speed
; - 2 and above is a multiplier of the run speed (less 1.0 since Run is 2.0)
; afRampTime is in seconds
Function ForceMovementRotationSpeedRamp(float afXMult = 0.0, float afYMult = 0.0, float afZMult = 0.0, float afRampTime = 0.1) native

; Sets the target movement direction on the actor
; afXAngle, afYAngle and afZAngle are in degrees
```

```
Function ForceTargetDirection(float afXAngle = 0.0, float afYAngle = 0.0, float afZAngle = 0.0) native

; Sets the target movement speed on the actor
; afSpeedMult is a speed multiplier based on the current max speeds
; - 0 -> 1 Scales between 0 and the Walk speed
; - 1 -> 2 Scales between Walk speed and Run Speed
; - 2 and above is a multiplier of the run speed (less 1.0 since Run is 2.0)
Function ForceTargetSpeed(float afSpeed) native

; Sets the target facing angle on the actor
; afXAngle, afYAngle and afZAngle are in degrees
Function ForceTargetAngle(float afXAngle = 0.0, float afYAngle = 0.0, float afZAngle = 0.0) native

; Clears any forced movement on the actor and return it to its standard state
Function ClearForcedMovement() native


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; returns the form for the item worn at the specified slotMask
; use Armor.GetMaskForSlot() to generate appropriate slotMask
Form Function GetWornForm(int slotMask) native

; returns the itemId for the item worn at the specified slotMask
int Function GetWornItemId(int slotMask) native

; returns the object currently equipped in the specified location
; 0 - left hand
; 1 - right hand
; 2 - shout
Form Function GetEquippedObject(int location) native

; returns the itemId of the object currently equipped in the specified hand
; 0 - left hand
; 1 - right hand
int Function GetEquippedItemId(int location) native

; returns the number of added spells for the actor
Int Function GetSpellCount() native

; returns the specified added spell for the actor
Spell Function GetNthSpell(int n) native

; Updates an Actors meshes (Used for Armor mesh/texture changes and face changes)
; DO NOT USE WHILE MOUNTED
Function QueueNiNodeUpdate() native

; Updates an Actors head mesh
Function RegenerateHead() native

int Property EquipSlot_Default = 0 AutoReadOnly
int Property EquipSlot_RightHand = 1 AutoReadOnly
int Property EquipSlot_LeftHand = 2 AutoReadOnly

; equips item at the given slot
Function EquipItemEx(Form item, int equipSlot = 0, bool preventUnequip = false, bool equipSound = true) native

; equips item with matching itemId at the given slot
Function EquipItemById(Form item, int itemId, int equipSlot = 0, bool preventUnequip = false, bool equipSound = true) native

; unequips item at the given slot
Function UnequipItemEx(Form item, int equipSlot = 0, bool preventEquip = false) native

; Adds a headpart, if the type exists it will replace, must not be misc type
; Beware: This function also affects the ActorBase
Function ChangeHeadPart(HeadPart hPart) native
```

```

; Replaces a headpart on the loaded mesh does not affect ActorBase
; Both old and new must exist, and be of the same type
Function ReplaceHeadPart(HeadPart oPart, HeadPart newPart) native

; Visually updates the actors weight
; neckDelta = (oldWeight / 100) - (newWeight / 100)
; Neck changes are player persistent, but actor per-session
; Weight itself is persistent either way so keep track of your
; original weight if you use this for Actors other than the player
; DO NOT USE WHILE MOUNTED
Function UpdateWeight(float neckDelta) native

; Returns whether the actors AI is enabled
bool Function IsAIEnabled() native

; Resets Actor AI
Function ResetAI() native

; Returns whether the actor is currently swimming
bool Function IsSwimming() native

; Sheathes the actors weapon
Function SheatheWeapon() native

; Returns the reference of the furniture the actor is currently using
ObjectReference Function GetFurnitureReference() native

; 0 - "Aah"
; 1 - "BigAah"
; 2 - "BMP"
; 3 - "ChJSh"
; 4 - "DST"
; 5 - "Eee"
; 6 - "Eh"
; 7 - "FV"
; 8 - "I"
; 9 - "K"
; 10 - "N"
; 11 - "Oh"
; 12 - "OohQ"
; 13 - "R"
; 14 - "Th"
; 15 - "W"
Function SetExpressionPhoneme(int index, float value) native

; 0 - "BlinkLeft"
; 1 - "BlinkRight"
; 2 - "BrowDownLeft"
; 3 - "BrowDownRight"
; 4 - "BrowInLeft"
; 5 - "BrowInRight"
; 6 - "BrowUpLeft"
; 7 - "BrowUpRight"
; 8 - "LookDown"
; 9 - "LookLeft"
; 10 - "LookRight"
; 11 - "LookUp"
; 12 - "SquintLeft"
; 13 - "SquintRight"
; 14 - "HeadPitch"
; 15 - "HeadRoll"
; 16 - "HeadYaw"
Function SetExpressionModifier(int index, float value) native

; Resets all expression, phoneme, and modifiers
Function ResetExpressionOverrides() native

; Returns all factions with the specified min and max ranks (-128 to 127)
```

```
Faction[] Function GetFactions(int minRank, int maxRank) native   :: Add a newline
between files
Scriptname ActorBase extends Form Hidden

; Returns this actor's class
Class Function GetClass() native

; Gets the number of actors of this type that have been killed
int Function GetDeadCount() native

; Returns this actor's gift filter formlist
FormList Function GetGiftFilter() native

; Returns this actor's race
Race Function GetRace() native

; Returns this actor's sex. Values for sex are:
; -1 - None
; 0 - Male
; 1 - Female
int Function GetSex() native

; Is this actor essential?
bool Function IsEssential() native

; Is this actor invulnerable?
bool Function IsInvulnerable() native

; Is this actor protected (can only be killed by player)?
bool Function IsProtected() native

; Is this actor base unique?
bool Function IsUnique() native

; Sets this actor as essential or not - if set as essential, will UNSET protected
Function SetEssential(bool abEssential = true) native

; Sets this actor as invulnerable or not
Function SetInvulnerable(bool abInvulnerable = true) native

; Sets this actor as protected or not - if set as protected, will UNSET essential
Function SetProtected(bool abProtected = true) native

; Sets the actors outfit
Function SetOutfit( Outfit akOutfit, bool abSleepOutfit = false ) native


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; get/set the CombatStyle of the actor
CombatStyle Function GetCombatStyle() native
Function SetCombatStyle(CombatStyle cs) native

; Get the Outfit of the actor
Outfit Function GetOutfit(bool bSleepOutfit = false) native

; set the Class of the actor
Function SetClass(Class c) native

; Get/Set the actors body height
float Function GetHeight() native
Function SetHeight(float height) native

; Get/Set the actors body weight
float Function GetWeight() native
Function SetWeight(float weight) native

; Get/Set actors HeadPart by index
int Function GetNumHeadParts() native
HeadPart Function GetNthHeadPart(int slotPart) native
```

```
Function SetNthHeadPart(HeadPart headPart, int slotPart) native
int Function GetIndexOfHeadPartByType(int type) native

; These functions are READ-ONLY they are for accessing the
; HeadPart list when the ActorBase's Race has been overlayed
; with another race (e.g. Vampires)
int Function GetNumOverlayHeadParts() native
HeadPart Function GetNthOverlayHeadPart(int slotPart) native
int Function GetIndexOfOverlayHeadPartByType(int type) native

; Get/Set actors face morph value by index
float Function GetFaceMorph(int index) native
Function SetFaceMorph(float value, int index) native

; Get/Set actors facemorph preset by index
; 0 - Nose
; 1 - ??
; 2 - Mouth
; 3 - Eyes
int Function GetFacePreset(int index) native
Function SetFacePreset(int value, int index) native

ColorForm Function GetHairColor() native
Function SetHairColor(ColorForm color) native

; returns the number of spells defined in the base actor form
int Function GetSpellCount() native

; returns the specified spell defined in the base actor  form
Spell Function GetNthSpell(int n) native

; returns the face textureset of the actor (Player Only?)
TextureSet Function GetFaceTextureSet() native
Function SetFaceTextureSet(TextureSet textures) native

; Gets/sets the Actor's voicetype
VoiceType Function GetVoiceType() native
Function SetVoiceType(VoiceType nVoice) native

; Gets/sets the skin of the actorbase
Armor Function GetSkin() native
Function SetSkin(Armor skin) native

; Gets/sets the far away skin of the actorbase
Armor Function GetSkinFar() native
Function SetSkinFar(Armor skin) native

; Gets the root template of the ActorBase
ActorBase Function GetTemplate() native   :: Add a newline between files
Scriptname ActorValueInfo extends Form Hidden

; Returns the AVI by name
ActorValueInfo Function GetActorValueInfoByName(string avName) global native
ActorValueInfo Function GetAVIByName(string avName) global
    return GetActorValueInfoByName(avName)
EndFunction

; Returns the AVI by id (0-164)
ActorValueInfo Function GetActorValueInfoByID(int id) global native
ActorValueInfo Function GetAVIByID(int id) global
    return GetActorValueInfoByID(id)
EndFunction

; Returns whether this AVI is a skill
bool Function IsSkill() native

; Skill Multiplier manipulation
float Function GetSkillUseMult() native
Function SetSkillUseMult(float value) native
```

```
float Function GetSkillOffsetMult() native
Function SetSkillOffsetMult(float value) native

float Function GetSkillImproveMult() native
Function SetSkillImproveMult(float value) native

float Function GetSkillImproveOffset() native
Function SetSkillImproveOffset(float value) native

; Returns the amount of experienced gained in this skill
float Function GetSkillExperience() native

; Does not trigger skill-up
Function SetSkillExperience(float exp) native

; Adds experience to this skill (Same as console AdvanceSkill, triggers skill-up)
Function AddSkillExperience(float exp) native

; Returns the experience required for skill-up
; (ImproveMult * currentLevel ^ fSkillUseCurve + ImproveOffset)
float Function GetExperienceForLevel(int currentLevel) native

; Returns the legendary level of this skill
int Function GetSkillLegendaryLevel() native

; Sets the legendary level of this skill
Function SetSkillLegendaryLevel(int level) native

; Returns perks from the skill into the FormList
; Actor filter applies to unowned and allRanks
; unowned will add perks that the actor does not own, or only perks the actor owns
; allRanks will add all ranks of each perk to the list, unowned/owned filter also applies
Function GetPerkTree(FormList list, Actor akActor = None, bool unowned = true, bool
allRanks = false) native

; Same as GetPerkTree except returns into a new array
Perk[] Function GetPerks(Actor akActor = None, bool unowned = true, bool allRanks =
false) native

; Same as Actor.GetActorValue (convenience function)
float Function GetCurrentValue(Actor akActor) native

; Same as Actor.GetBaseActorValue (convenience function)
float Function GetBaseValue(Actor akActor) native

; Acquires the Maximum value for the current ActorValue
float Function GetMaximumValue(Actor akActor) native   :: Add a newline between files
Scriptname Alias Hidden

; Returns the quest that owns this alias
Quest Function GetOwningQuest() native

; Register for the specified animation event from the specified object - returns true if
it successfully registered
bool Function RegisterForAnimationEvent(ObjectReference akSender, string asEventName)
native

; Register for LOS gain and lost events between the viewer and the target
; A loss or gain event will be sent immediately, depending on whether or not the viewer
is already looking at the target or not
; If the viewer is not the player, the target must be another actor
Function RegisterForLOS(Actor akViewer, ObjectReference akTarget) native

; Register for only the first LOS gain event between the viewer and the target
; If the viewer is already looking at the target, an event will be received almost
immediately
; If the viewer is not the player, the target must be another actor
Function RegisterForSingleLOSGain(Actor akViewer, ObjectReference akTarget) native
```

```
; Register for only the first LOS lost event between the viewer and the target
; If the viewer is already not looking at the target, an event will be received almost immediately
; If the viewer is not the player, the target must be another actor
Function RegisterForSingleLOSLost(Actor akViewer, ObjectReference akTarget) native

; Register for a single OnUpdate event, in afInterval seconds. All scripts attached to this alias will get the update events
; Of course, this means you don't need to call UnregisterForUpdate()
; If you find yourself doing this:
; Event OnUpdate()
;     UnregisterForUpdate()
;     {Do some stuff}
; endEvent
; Then you should use RegisterForSingleUpdate instead
Function RegisterForSingleUpdate(float afInterval) native

; Register for OnUpdate events, every X seconds, where X is the interval. All scripts attached to this alias will get the update events
Function RegisterForUpdate(float afInterval) native

; Register for OnUpdateGameTime events, every X hours of game time, where X is the interval. All scripts attached to this alias will get the update events
Function RegisterForUpdateGameTime(float afInterval) native

; Register for a single OnUpdateGameTime event, in afInterval hours of game time. All scripts attached to this alias will get the update events
Function RegisterForSingleUpdateGameTime(float afInterval) native

; Registers this alias to receive events when the player sleeps and wakes up
Function RegisterForSleep() native

; Registers this alias to receive events when tracked stats are updated
Function RegisterForTrackedStatsEvent() native

; Turns on profiling for this specific object and all scripts attached to it - setting doesn't persist across saves
; Will do nothing on release console builds, and if the Papyrus:bEnableProfiling ini setting is off
Function StartObjectProfiling() native

; Turns off profiling for this specific object and all scripts attached to it - setting doesn't persist across saves
; Will do nothing on release console builds, and if the Papyrus:bEnableProfiling ini setting is off
Function StopObjectProfiling() native

; Unregister for any LOS events between the viewer and target
Function UnregisterForLOS(Actor akViewer, ObjectReference akTarget) native

; Unregister for the specified animation event from the specified object
Function UnregisterForAnimationEvent(ObjectReference akSender, string asEventName) native

; Unregisters this alias to receive events when the player sleeps and wakes up
Function UnregisterForSleep() native

; Unregisters this alias from receiving events when tracked stats are updated
Function UnregisterForTrackedStatsEvent() native

; Unregister for OnUpdate events, all attached scripts will stop getting update events
Function UnregisterForUpdate() native

; Unregister for OnUpdateGameTime events, all attached scripts will stop getting update game time events
Function UnregisterForUpdateGameTime() native

; Animation event, sent when an object we are listening to hits one of the events we are listening for
```

```
Event OnAnimationEvent(ObjectReference akSource, string asEventName)
EndEvent

; Event sent when you have been unregistered from receiving an animation event because
the target
; object's animation graph has been unloaded
Event OnAnimationEventUnregistered(ObjectReference akSource, string asEventName)
EndEvent

; LOS event, sent whenever the viewer first sees the target (after registering)
Event OnGainLOS(Actor akViewer, ObjectReference akTarget)
EndEvent

; Lost LOS event, sent whenever the viewer first loses sight of the target (after
registering)
Event OnLostLOS(Actor akViewer, ObjectReference akTarget)
EndEvent

; Received when the player sleeps. Start and desired end time are in game time days
(after registering)
Event OnSleepStart(float afSleepStartTime, float afDesiredSleepEndTime)
EndEvent

; Received when the player stops sleeping - whether naturally or interrupted (after
registering)
Event OnSleepStop(bool abInterrupted)
EndEvent

; Event received when a tracked stat is updated for the player
Event OnTrackedStatsEvent(string arStatName, int aiStatValue)
EndEvent

; Update event, sent every X seconds while this alias is registered for them
Event OnUpdate()
EndEvent

; Update event, sent every X hours of game time while this alias is registered for them
Event OnUpdateGameTime()
EndEvent

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; return the name of the alias
string Function GetName() native

; return the id of the alias
int Function GetID() native

; Registers for OnKeyDown and OnKeyUp events for the given keycode.
Function RegisterForKey(int keyCode) native
Function UnregisterForKey(int keyCode) native
Function UnregisterForAllKeys() native

Event OnKeyDown(int keyCode)
EndEvent

Event OnKeyUp(int keyCode, float holdTime)
EndEvent

; Registers for OnControlDown and OnControlUp events for the given control.
; For a list of valid controls, see Input.psc.
Function RegisterForControl(string control) native
Function UnregisterForControl(string control) native
Function UnregisterForAllControls() native

Event OnControlDown(string control)
EndEvent

Event OnControlUp(string control, float holdTime)
EndEvent
```

```

; Registers for OnMenuOpen and OnMenuClose events for the given menu.
; Registrations have to be refreshed after each game load.
; For a list of valid menu names, see UI.psc.
Function RegisterForMenu(string menuName) native
Function UnregisterForMenu(string menuName) native
Function UnregisterForAllMenus() native

Event OnMenuOpen(string menuName)
endEvent

Event OnMenuClose(string menuName)
endEvent

; Registers a custom event callback for given event name.
; Registrations have to be refreshed after each game load.
;
;	Examples:
;		RegisterForModEvent("myCustomEvent", "MyModEventCallback")
;
;	Event signature of custom event callbacks:
;		Event MyModEventCallback(string eventName, string strArg, float numArg, Form sender)
;		endEvent
;
Function RegisterForModEvent(string eventName, string callbackName) native
Function UnregisterForModEvent(string eventName) native
Function UnregisterForAllModEvents() native

; Sends custom event with given generic parameters.
Function SendModEvent(string eventName, string strArg = "", float numArg = 0.0) native

; See Form.psc
Function RegisterForCameraState() native
Function UnregisterForCameraState() native

Event OnPlayerCameraState(int oldState, int newState)
EndEvent

; See Form.psc
Function RegisterForCrosshairRef() native
Function UnregisterForCrosshairRef() native

Event OnCrosshairRefChange(ObjectReference ref)
EndEvent

; See Form.psc
Function RegisterForActorAction(int actionType) native
Function UnregisterForActorAction(int actionType) native

Event OnActorAction(int actionType, Actor akActor, Form source, int slot)
EndEvent

; Registers the script for when a QueueNiNodeUpdate is called
Function RegisterForNiNodeUpdate() native
Function UnregisterForNiNodeUpdate() native

Event OnNiNodeUpdate(ObjectReference akActor)
EndEvent   :: Add a newline between files
Scriptname Ammo extends Form Hidden

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

; Returns whether this ammo is a bolt
bool Function IsBolt() native

; Returns the projectile associated with this ammo
Projectile Function GetProjectile() native

```

```
; Returns the base damage of this ammo
float Function GetDamage() native
   :: Add a newline between files
Scriptname Apparatus extends MiscObject Hidden


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

int Function GetQuality() native
Function SetQuality(int quality) native   :: Add a newline between files
Scriptname Armor extends Form Hidden

; Returns the "warmth rating" for this armor
float Function GetWarmthRating() native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
int Function GetArmorRating() native
int Function GetAR()
    return GetArmorRating()
endFunction

Function SetArmorRating(int armorRating) native
Function SetAR(int armorRating)
    return SetArmorRating(armorRating)
endFunction

Function ModArmorRating(int modBy) native
Function ModAR(int modBy)
    return ModArmorRating(modBy)
endFunction

; works on the path to the nif file representing the in-game model of the weapon
string Function GetModelPath(bool bFemalePath) native
Function SetModelPath(string path, bool bFemalePath) native

; works on the path to the nif file representing the icon for the weapon in the inventory
string Function GetIconPath(bool bFemalePath) native
Function SetIconPath(string path, bool bFemalePath) native

; works on the path to the file representing the message icon for the weapon
string Function GetMessageIconPath(bool bFemalePath) native
Function SetMessageIconPath(string path, bool bFemalePath) native

; Weight Class
; 0 = Light Armor
; 1 = Heavy Armor
; 2 = None
int Function GetWeightClass() native
Function SetWeightClass(int weightClass) native

; works on the enchantment associated with the armor
Enchantment Function GetEnchantment() native
Function SetEnchantment(Enchantment e) native

; Armor info by keyword
bool Function IsLightArmor()
    return HasKeywordString("ArmorLight")
endFunction

bool Function IsHeavyArmor()
    return HasKeywordString("ArmorHeavy")
endFunction

bool Function IsClothing()
    return HasKeywordString("ArmorClothing")
endFunction

bool Function IsBoots()
    return HasKeywordString("ArmorBoots")
```

```
endFunction

bool Function IsCuirass()
    return HasKeywordString("ArmorCuirass")
endFunction

bool Function IsGauntlets()
    return HasKeywordString("ArmorGauntlets")
endFunction

bool Function IsHelmet()
    return HasKeywordString("ArmorHelmet")
endFunction

bool Function IsShield()
    return HasKeywordString("ArmorShield")
endFunction

bool Function IsJewelry()
    return HasKeywordString("ArmorJewelry")
endFunction

bool Function IsClothingHead()
    return HasKeywordString("ClothingHead")
endFunction

bool Function IsClothingBody()
    return HasKeywordString("ClothingBody")
endFunction

bool Function IsClothingFeet()
    return HasKeywordString("ClothingFeet")
endFunction

bool Function IsClothingHands()
    return HasKeywordString("ClothingHands")
endFunction

bool Function IsClothingRing()
    return HasKeywordString("ClothingRing")
endFunction

bool Function IsClothingRich()
    return HasKeywordString("ClothingRich")
endFunction

bool Function IsClothingPoor()
    return HasKeywordString("ClothingPoor")
endFunction


; Functions and Flags dealing the BipedObject slot values from the CK
; These are the equivalent of 1 << (SlotMask-30).  Basically
; these are a flags where 30 is the first bit, and 61 is the 31st bit.

; returns the slot mask for the armor.
int Function GetSlotMask() native
; sets the slot mask for the armor
Function SetSlotMask(int slotMask) native
; adds the specified slotMask to the armor
int Function AddSlotToMask(int slotMask) native
; removes the specified slot masks from the armor
int Function RemoveSlotFromMask(int slotMask) native

; calculates the equivalent value for the properties below
int Function GetMaskForSlot(int slot) global native

; returns the number of armor addons for this armor
int Function GetNumArmorAddons() native
```

```

; returns the nth armor addon for this armor
ArmorAddon Function GetNthArmorAddon(int n) native

; returns the SlotMask for a single slot from the CK
; can be used with the non-global SlotMask functions above
; and with the Math bit shifting functions
int Property kSlotMask30 =  0x00000001 AutoReadOnly
int Property kSlotMask31 =  0x00000002 AutoReadOnly
int Property kSlotMask32 =  0x00000004 AutoReadOnly
int Property kSlotMask33 =  0x00000008 AutoReadOnly
int Property kSlotMask34 =  0x00000010 AutoReadOnly
int Property kSlotMask35 =  0x00000020 AutoReadOnly
int Property kSlotMask36 =  0x00000040 AutoReadOnly
int Property kSlotMask37 =  0x00000080 AutoReadOnly
int Property kSlotMask38 =  0x00000100 AutoReadOnly
int Property kSlotMask39 =  0x00000200 AutoReadOnly
int Property kSlotMask40 =  0x00000400 AutoReadOnly
int Property kSlotMask41 =  0x00000800 AutoReadOnly
int Property kSlotMask42 =  0x00001000 AutoReadOnly
int Property kSlotMask43 =  0x00002000 AutoReadOnly
int Property kSlotMask44 =  0x00004000 AutoReadOnly
int Property kSlotMask45 =  0x00008000 AutoReadOnly
int Property kSlotMask46 =  0x00010000 AutoReadOnly
int Property kSlotMask47 =  0x00020000 AutoReadOnly
int Property kSlotMask48 =  0x00040000 AutoReadOnly
int Property kSlotMask49 =  0x00080000 AutoReadOnly
int Property kSlotMask50 =  0x00100000 AutoReadOnly
int Property kSlotMask51 =  0x00200000 AutoReadOnly
int Property kSlotMask52 =  0x00400000 AutoReadOnly
int Property kSlotMask53 =  0x00800000 AutoReadOnly
int Property kSlotMask54 =  0x01000000 AutoReadOnly
int Property kSlotMask55 =  0x02000000 AutoReadOnly
int Property kSlotMask56 =  0x04000000 AutoReadOnly
int Property kSlotMask57 =  0x08000000 AutoReadOnly
int Property kSlotMask58 =  0x10000000 AutoReadOnly
int Property kSlotMask59 =  0x20000000 AutoReadOnly
int Property kSlotMask60 =  0x40000000 AutoReadOnly
int Property kSlotMask61 =  0x80000000 AutoReadOnly   :: Add a newline between files
Scriptname ArmorAddon extends Form Hidden

; returns the model path of the particular model
string Function GetModelPath(bool firstPerson, bool female) native

; sets the model path of the particular model
Function SetModelPath(string path, bool firstPerson, bool female) native

; returns the number of texturesets for the particular model
int Function GetModelNumTextureSets(bool first, bool female) native

; returns the nth textureset for the particular model
TextureSet Function GetModelNthTextureSet(int n, bool first, bool female) native

; sets the nth textureset for the particular model
Function SetModelNthTextureSet(TextureSet texture, int n, bool first, bool female) native

; returns the number of races this armor addon applies to
int Function GetNumAdditionalRaces() native

; returns the nth race this armor addon applies to
Race Function GetNthAdditionalRace(int n) native

; Functions and Flags dealing the BipedObject slot values from the CK
; These are the equivalent of 1 << (SlotMask-30).  Basically
; these are a flags where 30 is the first bit, and 61 is the 31st bit.

; returns the slot mask for the armor addon.
int Function GetSlotMask() native
; sets the slot mask for the armor addon
```

```
Function SetSlotMask(int slotMask) native
; adds the specified slotMask to the armor addon
int Function AddSlotToMask(int slotMask) native
; removes the specified slot masks from the armor addon
int Function RemoveSlotFromMask(int slotMask) native

; calculates the equivalent mask value for the slot
; This is a global function, use it directly from Armor as it is faster
int Function GetMaskForSlot(int slot) global
    return Armor.GetMaskForSlot(slot)
EndFunction   :: Add a newline between files
Scriptname Art extends Form Hidden

string Function GetModelPath() native
Function SetModelPath(string path) native   :: Add a newline between files
Scriptname Book Extends Form Hidden

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; Returns the spell that this book teaches
Spell Function GetSpell() native
Int Function GetSkill() native
bool Function IsRead() native
bool Function IsTakeable() native   :: Add a newline between files
Scriptname Camera Hidden


; Returns the character's current camera state
; 0 - first person
; 1 - auto vanity
; 2 - VATS
; 3 - free
; 4 - iron sights
; 5 - furniture
; 6 - transition
; 7 - tweenmenu
; 8 - third person 1
; 9 - third person 2
; 10 - horse
; 11 - bleedout
; 12 - dragon
int Function GetCameraState() global native

; Updates the camera when changing Shoulder positions
Function UpdateThirdPerson() global native

; Returns the player's camera FOV
float Function GetWorldFieldOfView() global native
float Function GetWorldFOV() global
    return GetWorldFieldOfView()
EndFunction

; Sets the player's camera FOV
Function SetWorldFieldOfView(float fov) global native
Function SetWorldFOV(float fov) global
    SetWorldFieldOfView(fov)
EndFunction

; Returns the player's camera FOV
float Function GetFirstPersonFieldOfView() global native
float Function GetFirstPersonFOV() global
    return GetFirstPersonFieldOfView()
EndFunction

; Sets the player's camera FOV
Function SetFirstPersonFieldOfView(float fov) global native
Function SetFirstPersonFOV(float fov) global
    SetFirstPersonFieldOfView(fov)
EndFunction   :: Add a newline between files
Scriptname Cell extends Form Hidden
```

```

; Gets the actor that owns this cell (or none if not owned by an actor)
ActorBase Function GetActorOwner() native

; Gets the faction that owns this cell (or none if not owned by a faction)
Faction Function GetFactionOwner() native

; Is this cell "attached"? (In the loaded area)
bool Function IsAttached() native

; Is this cell an interior cell?
bool Function IsInterior() native

; Flags the cell for reset on next load
Function Reset() native

; Sets this cell's owner as the specified actor
Function SetActorOwner(ActorBase akActor) native

; Sets this cell's owner as the specified faction
Function SetFactionOwner(Faction akFaction) native

; Sets the fog color for this cell (interior, non-sky-lit cells only)
Function SetFogColor(int aiNearRed, int aiNearGreen, int aiNearBlue, \
    int aiFarRed, int aiFarGreen, int aiFarBlue) native

; Adjusts this cell's fog near and far planes (interior, non-sky-lit cells only)
Function SetFogPlanes(float afNear, float afFar) native

; Sets the fog power for this cell (interior, non-sky-lit cells only)
Function SetFogPower(float afPower) native

; Sets this cell as public or private
Function SetPublic(bool abPublic = true) native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; Returns the number of refs in the cell
int Function GetNumRefs(int formTypeFilter = 0) native

; returns the ref at the specified index
ObjectReference Function GetNthRef(int n, int formTypeFilter = 0) native

; Returns the water level of the cell (-2147483648 if no water)
float Function GetWaterLevel() native

; Returns water level of the cell, if default returns water level from worldspace
float Function GetActualWaterLevel() native   :: Add a newline between files
Scriptname ColorComponent Hidden

int Function GetAlpha(int argb) global native
int Function GetRed(int argb) global native
int Function GetGreen(int argb) global native
int Function GetBlue(int argb) global native

float Function GetHue(int argb) global native
float Function GetSaturation(int argb) global native
float Function GetValue(int argb) global native

int Function SetAlpha(int argb, int a) global native
int Function SetRed(int argb, int r) global native
int Function SetGreen(int argb, int g) global native
int Function SetBlue(int argb, int b) global native

int Function SetHue(int argb, float h) global native
int Function SetSaturation(int argb, float s) global native
int Function SetValue(int argb, float v) global native   :: Add a newline between files
Scriptname ColorForm extends Form Hidden

int Function GetColor() native
```

```
Function SetColor(int color) native

int Function GetRed()
    return ColorComponent.GetRed(Self.GetColor())
EndFunction

int Function GetGreen()
    return ColorComponent.GetGreen(Self.GetColor())
EndFunction

int Function GetBlue()
    return ColorComponent.GetBlue(Self.GetColor())
EndFunction

float Function GetHue()
    return ColorComponent.GetHue(Self.GetColor())
EndFunction

float Function GetSaturation()
    return ColorComponent.GetSaturation(Self.GetColor())
EndFunction

float Function GetValue()
    return ColorComponent.GetValue(Self.GetColor())
EndFunction   :: Add a newline between files
Scriptname CombatStyle extends Form Hidden

; functions related to the General Tab values
float Function GetOffensiveMult() native
float Function GetDefensiveMult() native
float Function GetGroupOffensiveMult() native
float Function GetAvoidThreatChance() native
float Function GetMeleeMult() native
float Function GetRangedMult() native
float Function GetMagicMult() native
float Function GetShoutMult() native
float Function GetStaffMult() native
float Function GetUnarmedMult() native

Function SetOffensiveMult(float mult) native
Function SetDefensiveMult(float mult) native
Function SetGroupOffensiveMult(float mult) native
Function SetAvoidThreatChance(float chance) native
Function SetMeleeMult(float mult) native
Function SetRangedMult(float mult) native
Function SetMagicMult(float mult) native
Function SetShoutMult(float mult) native
Function SetStaffMult(float mult) native
Function SetUnarmedMult(float mult) native

; functions related to the Melee tab values
float Function GetMeleeAttackStaggeredMult() native
float Function GetMeleePowerAttackStaggeredMult() native
float Function GetMeleePowerAttackBlockingMult() native
float Function GetMeleeBashMult() native
float Function GetMeleeBashRecoiledMult() native
float Function GetMeleeBashAttackMult() native
float Function GetMeleeBashPowerAttackMult() native
float Function GetMeleeSpecialAttackMult() native
bool Function GetAllowDualWielding() native

Function SetMeleeAttackStaggeredMult(float mult) native
Function SetMeleePowerAttackStaggeredMult(float mult) native
Function SetMeleePowerAttackBlockingMult(float mult) native
Function SetMeleeBashMult(float mult) native
Function SetMeleeBashRecoiledMult(float mult) native
Function SetMeleeBashAttackMult(float mult) native
Function SetMeleeBashPowerAttackMult(float mult) native
Function SetMeleeSpecialAttackMult(float mult) native
Function SetAllowDualWielding(bool allow) native

; functions related to the Close Range tab values
float Function GetCloseRangeDuelingCircleMult() native
float Function GetCloseRangeDuelingFallbackMult() native
float Function GetCloseRangeFlankingFlankDistance() native
float Function GetCloseRangeFlankingStalkTime() native

Function SetCloseRangeDuelingCircleMult(float mult) native
Function SetCloseRangeDuelingFallbackMult(float mult) native
Function SetCloseRangeFlankingFlankDistance(float mult) native
Function SetCloseRangeFlankingStalkTime(float mult) native

; functions related to the LongRange tab values
float Function GetLongRangeStrafeMult() native
Function SetLongRangeStrafeMult(float mult) native

; functions related to the Flight tab values
float Function GetFlightHoverChance() native
float Function GetFlightDiveBombChance() native
float Function GetFlightFlyingAttackChance() native

Function SetFlightHoverChance(float chance) native
Function SetFlightDiveBombChance(float chance) native
Function SetFlightFlyingAttackChance(float mult) native


   :: Add a newline between files
Scriptname ConstructibleObject extends MiscObject Hidden

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

; Gets/Sets the result of this recipe
Form Function GetResult() native
Function SetResult(Form result) native

; Gets/Sets the amount of results of this recipe
int Function GetResultQuantity() native
Function SetResultQuantity(int quantity) native

; Gets the number of ingredients
int Function GetNumIngredients() native

; Gets/Sets the Nth ingredient required
Form Function GetNthIngredient(int n) native
Function SetNthIngredient(Form required, int n) native

; Gets/Sets the quantity of Nth ingredient required
int Function GetNthIngredientQuantity(int n) native
Function SetNthIngredientQuantity(int value, int n) native

; Gets/Sets the Workbench keyword (Which apparatus creates this)
Keyword Function GetWorkbenchKeyword() native
Function SetWorkbenchKeyword(Keyword aKeyword) native   :: Add a newline between files
Scriptname DefaultObjectManager extends Form Hidden

; Returns the default form for this key e.g. 'GOLD'
Form Function GetForm(string key) native

; Sets the default form for the particular key
Function SetForm(string key, Form newForm) native

; Valid Keys
; WWSP - Werewolf Spell
; SALT - Sitting Angle Limit
; APSH - Allow Player Shout
; GOLD - Gold
; LKPK - Lockpick
; SKLK - SkeletonKey
; PFAC - Player Faction
; GFAC - Guard Faction
; DFMS - Default Music
; BTMS - Battle Music
; DTMS - Death Music
; SCMS - Success Music
; LUMS - Level Up Music
; DCMS - Dungeon Cleared Music
; PVMA - Player Voice (Male)
; PVMC - Player Voice (Male Child)
; PVFA - Player Voice (Female)
; PVFC - Player Voice (Female Child)
; EPDF - Eat Package Default Food
; LHEQ - LeftHand Equip
; RHEQ - RightHand Equip
; EHEQ - EitherHand Equip
; VOEQ - Voice Equip
; POEQ - Potion Equip
; EACA - Every Actor Ability
; CACA - Commanded Actor Ability
; DEIS - Drug Wears Off Image Space
; DFTS - Footstep Set
; DLMT - Landscape Material
; DLZM - Dragon Land Zone Marker
; DCZM - Dragon Crash Zone Marker
; CSTY - Combat Style
; PLST - Default Pack List
; PWFD - Wait-For-Dialogue Package
; LRTB - LocRefType Boss
; VLOC - Virtual Location
; PLOC - PersistAll Location
; INVP - Inventory Player
; PTNP - Pathing Test NPC
; FPCS - Favor Cost Small
; FPCM - Favor Cost Medium
; FPCL - Favor Cost Large
; FGPD - Favor Gifts Per Day
; AASW - Action Swim State Change
; AALK - Action Look
; AALA - Action LeftAttack
; AALD - Action LeftReady
; AALR - Action LeftRelease
; AALI - Action LeftInterrupt
; AARA - Action RightAttack
; AARD - Action RightReady
; AARR - Action RightRelease
; AARI - Action RightInterrupt
; AADA - Action DualAttack
; AADL - Action DualRelease
; AAAC - Action Activate
; AAJP - Action Jump
; AAFA - Action Fall
; AALN - Action Land
; AASN - Action Sneak
; AAVC - Action Voice
; AAVD - Action VoiceReady
; AAVR - Action VoiceRelease
; AAVI - Action VoiceInterrupt
; AAID - Action Idle
; AAST - Action Sprint Start
; AASP - Action Sprint Stop
; AADR - Action Draw
; AASH - Action Sheath
; ALPA - Action Left Power Attack
; AAPA - Action Right Power Attack
; ADPA - Action Dual Power Attack
; AAS1 - Action Stagger Start
; AABH - Action Block Hit
; AABA - Action Block Anticipate
; AARC - Action Recoil
; AAR2 - Action Large Recoil
; AAB1 - Action Bleedout Start
; AAB2 - Action Bleedout Stop
; AAIS - Action Idle Stop
; AAWH - Action Ward Hit
; AAFQ - Action Force Equip
; AASC - Action Shield Change
; AAPS - Action Path Start
; AAPE - Action Path End
; AALM - Action Large Movement Delta
; AAF1 - Action Fly Start
; AAF2 - Action Fly Stop
; AAH1 - Action Hover Start
; AAH2 - Action Hover Stop
; AABI - Action Bumped Into
; AASS - Action Summoned Start
; ATKI - Action Talking Idle
; ALTI - Action Listen Idle
; AADE - Action Death
; AADW - Action Death Wait
; AIDW - Action Idle Warn
; AMST - Action Move Start
; AMSP - Action Move Stop
; ATRI - Action Turn Right
; ATLE - Action Turn Left
; ATSP - Action Turn Stop
; AMFD - Action Move Forward
; AMBK - Action Move Backward
; AMLT - Action Move Left
; AMRT - Action Move Right
; ARAG - Action Reset Animation Graph
; AKDN - Action Knockdown
; AAGU - Action Get Up
; ASID - Action Idle Stop Instant
; ARGI - Action Ragdoll Instant
; AWWS - Action Waterwalk Start
; AREL - Action Reload
; PUSG - Pickup Sound Generic
; PDSG - Putdown Sound Generic
; PUSW - Pickup Sound Weapon
; PDSW - Putdown Sound Weapon
; PUSA - Pickup Sound Armor
; PDSA - Putdown Sound Armor
; PUSB - Pickup Sound Book
; PDSB - Putdown Sound Book
; PUSI - Pickup Sound Ingredient
; PDSI - Putdown Sound Ingredient
; HVSS - Harvest Sound
; HVFS - Harvest Failed Sound
; WBSN - Ward Break Sound
; WASN - Ward Absorb Sound
; WDSN - Ward Deflect Sound
; MFSN - Magic Fail Sound
; SFSN - Shout Fail Sound
; HFSD - Heartbeat Sound Fast
; HSSD - Heartbeat Sound Slow
; IMLH - Imagespace: Low Health
; SCSD - Soul Captured Sound
; NASD - No-Activation Sound
; MMSD - Map Menu Looping Sound
; DDSC - Dialogue Voice Category
; NDSC - Non-Dialogue Voice Category
; SFDC - SFX To Fade In Dialogue Category
; PDMC - Pause During Menu Category (Fade)
; PIMC - Pause During Menu Category (Immediate)
; PDLC - Pause During Loading Menu Category
; MDSC - Music Sound Category
; SMSC - Stats Mute Category
; SSSC - Stats Music
; MTSC - Master Sound Category
; TSSC - Time Sensitive Sound Category
; DOP2 - Dialogue Output Model (3D)
; DOP3 - Dialogue Output Model (2D)
; POPM - Player's Output Model (1st Person)
; P3OM - Player's Output Model (3rd Person)
; IOPM - Interface Output Model
; RVBT - Reverb Type
; UWLS - Underwater Loop Sound
; URVT - Underwater Reverb Type
; HRSK - Keyword - Horse
; UNDK - Keyword - Undead
; NPCK - Keyword - NPC
; KWBR - Keyword - BeastRace
; KWDM - Keyword - DummyObject
; KWGE - Keyword - UseGeometryEmitter
; KWMS - Keyword - MustStop
; KWUA - Keyword - UpdateDuringArchery
; KWOT - Keyword - Skip Outfit Items
; FTHD - Male Face Texture Set: Head
; FTMO - Male Face Texture Set: Mouth
; FTEL - Male Face Texture Set: Eyes
; FTHF - Female Face Texture Set: Head
; FTMF - Female Face Texture Set: Mouth
; FTRF - Female Face Texture Set: Eyes
; IMID - ImageSpaceModifier for inventory menu.
; PTEM - Package template
; MMCL - Main Menu Cell
; DMWL - Default MovementType: Walk
; DMRN - Default MovementType: Run
; DMSW - Default MovementType: Swim
; DMFL - Default MovementType: Fly
; DMSN - Default MovementType: Sneak
; DMSP - Default MovementType: Sprint
; SPFK - Keyword - Special Furniture
; FFFP - Keyword - Furniture Forces 1st Person
; FFTP - Keyword - Furniture Forces 3rd Person
; AFNP - Keyword - Activator Furniture No Player
; TKGS - Telekinesis Grab Sound
; TKTS - Telekinesis Throw Sound
; WMWE - World Map Weather
; HMPC - Help Manual PC
; HMXB - Help Manual XBox
; TKAM - Keyword - Type Ammo
; TKAR - Keyword - Type Armor
; TKBK - Keyword - Type Book
; TKIG - Keyword - Type Ingredient
; TKKY - Keyword - Type Key
; TKMS - Keyword - Type Misc
; TKSG - Keyword - Type SoulGem
; TKWP - Keyword - Type Weapon
; TKPT - Keyword - Type Potion
; BENW - Base Weapon Enchantment
; BENA - Base Armor Enchantment
; BAPO - Base Potion
; BAPS - Base Poison
; DRAK - Keyword - Dragon
; MVBL - Keyword - Movable
; ABSE - Art Object - Absorb Effect
; WEML - Weapon Material List
; ARTL - Armor Material List
; DIEN - Keyword - Disallow Enchanting
; FTML - Favor travel marker location
; LKHO - Keyword - Hold Location
; CWOK - Keyword - Civil War Owner
; CWNE - Keyword - Civil War Neutral
; LRSO - LocRefType - Civil War Soldier
; KWDO - Keyword - ClearableLocation
; LRRD - LocRefType - Resource Destructible
; HCLL - FormList - Hair Color List
; CMPX - Complex Scene Object
; RUSG - Keyword - Reusable SoulGem
; ANML - Keyword - Animal
; DAED - Keyword - Daedra
; BEEP - Keyword - Robot
; NRNT - Keyword - Nirnroot
; FTGF - Fighters' Guild Faction
; MGGF - Mages' Guild Faction
; TVGF - Thieves' Guild Faction
; DBHF - Dark Brotherhood Faction
; JRLF - Jarl Faction
; AWWW - Bunny Faction
; PIVV - Player Is Vampire Variable
; PIWV - Player Is Werewolf Variable
; NMRD - Road Marker
; SAT1 - Keyword: Scale Actor To 1.0
; VAMP - Keyword: Vampire
; FORG - Keyword: Forge
; COOK - Keyword: Cooking Pot
; SMLT - Keyword: Smelter
; TANN - Keyword: Tanning Rack
; HBLK - Help - Basic Lockpicking (PC)
; HBLX - Help - Basic Lockpicking (Console)
; HBFG - Help - Basic Forging
; HBCO - Help - Basic Cooking
; HBML - Help - Basic Smelting
; HBTA - Help - Basic Tanning
; HBOC - Help - Basic Object Creation
; HBEC - Help - Basic Enchanting
; HBSM - Help - Basic Smithing Weapon
; HBSA - Help - Basic Smithing Armor
; HBAL - Help - Basic Alchemy
; HBBR - Help - Barter
; HBLU - Help - Leveling up
; HBSK - Help - Skills Menu
; HBMM - Help - Map Menu
; HBJL - Help - Journal
; HBLH - Help - Low Health
; HBLM - Help - Low Magicka
; HBLS - Help - Low Stamina
; HBHJ - Help - Jail
; HBFT - Help - Teamate Favor
; HBWC - Help - Weapon Charge
; HBFS - Help - Favorites
; KHFL - Kinect Help FormList
; HBFM - Help - Flying Mount
; HBTL - Help - Target Lock
; HBAT - Help - Attack Target
; LSIS - Imagespace: Load screen
; WMDA - Keyword - Weapon Material Daedric
; WMDR - Keyword - Weapon Material Draugr
; WMDH - Keyword - Weapon Material DraugrHoned
; WMDW - Keyword - Weapon Material Dwarven
; WMEB - Keyword - Weapon Material Ebony
; WMEL - Keyword - Weapon Material Elven
; WMFA - Keyword - Weapon Material Falmer
; WMFH - Keyword - Weapon Material FalmerHoned
; WMGL - Keyword - Weapon Material Glass
; WMIM - Keyword - Weapon Material Imperial
; WMIR - Keyword - Weapon Material Iron
; WMOR - Keyword - Weapon Material Orcish
; WMST - Keyword - Weapon Material Steel
; WMWO - Keyword - Weapon Material Wood
; WTBA - Keyword - WeaponTypeBoundArrow
; AODA - Keyword - Armor Material Daedric
; AODP - Keyword - Armor Material Dragonplate
; AODS - Keyword - Armor Material Dragonscale
; AODB - Keyword - Armor Material Dragonbone
; AODW - Keyword - Armor Material Dwarven
; AOEB - Keyword - Armor Material Ebony
; AOEL - Keyword - Armor Material Elven
; AOES - Keyword - Armor Material ElvenSplinted
; AOFL - Keyword - Armor Material FullLeather
; AOGL - Keyword - Armor Material Glass
; AOHI - Keyword - Armor Material Hide
; AOIM - Keyword - Armor Material Imperial
; AOIH - Keyword - Armor Material ImperialHeavy
; AOIR - Keyword - Armor Material ImperialReinforced
; AOFE - Keyword - Armor Material Iron
; AOIB - Keyword - Armor Material IronBanded
; AOOR - Keyword - Armor Material Orcish
; AOSC - Keyword - Armor Material Scaled
; AOST - Keyword - Armor Material Steel
; AOSP - Keyword - Armor Material SteelPlate
; AOSK - Keyword - Armor Material Stormcloak
; AOSD - Keyword - Armor Material Studded
; GCK1 - Keyword - Generic Craftable Keyword 01
; GCK2 - Keyword - Generic Craftable Keyword 02
; GCK3 - Keyword - Generic Craftable Keyword 03
; GCK4 - Keyword - Generic Craftable Keyword 04
; GCK5 - Keyword - Generic Craftable Keyword 05
; GCK6 - Keyword - Generic Craftable Keyword 06
; GCK7 - Keyword - Generic Craftable Keyword 07
; GCK8 - Keyword - Generic Craftable Keyword 08
; GCK9 - Keyword - Generic Craftable Keyword 09
; GCKX - Keyword - Generic Craftable Keyword 10
; JWLR - Keyword - Jewelry
; KWCU - Keyword - Cuirass
; LMHP - Local Map Hide Plane
; SLDM - Snow LOD Material
; SLHD - Snow LOD Material (HD)
; ALDM - Ash LOD Material
; ALHD - Ash LOD Material (HD)
; DGFL - DialogueFollower Quest
; PTFR - PotentialFollower Faction
; AVWP - Werewolf Available Perks
; AVVP - Vampire Available Perks
; RIWR - Werewolf Race
; RIVR - Vampire Race
; RIVS - Vampire Spells
; DMXL - Dragon Mount No Land List
; PCMD - Player Can Mount Dragon Here List
; FMYS - Flying Mount - Allowed Spells
; FMNS - Flying Mount - Disallowed Spells
; MNT2 - Keyword - Mount
; AIVC - Verlet Cape
; FTNP - Furniture Test NPC
; COEX - Keyword - Conditional Explosion
; VFNC - Vampire Feed No Crime Faction
; KWSP - Skyrim - Worldspace
; ALBM - Keyword - Armor Material Light Bonemold
; ALCH - Keyword - Armor Material Light Chitin
; ALNC - Keyword - Armor Material Light Nordic
; ALSM - Keyword - Armor Material Light Stalhrim
; FMFF - Flying Mount - Fly Fast Worldspaces
; AHBM - Keyword - Armor Material Heavy Bonemold
; AHCH - Keyword - Armor Material Heavy Chitin
; AHNC - Keyword - Armor Material Heavy Nordic
; AHSM - Keyword - Armor Material Heavy Stalhrim
; WPNC - Keyword - Weapon Material Nordic
; WPSM - Keyword - Weapon Material Stalhrim   :: Add a newline between files
Scriptname Enchantment extends Form Hidden

; Is this enchantment classified as hostile?
bool Function IsHostile() native

```

```
; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; return the number of the effects
int Function GetNumEffects() native

; return the magnitude of the specified effect
float Function GetNthEffectMagnitude(int index) native

; return the area of the specified effect
int Function GetNthEffectArea(int index) native

; return the duration of the specified effect
int Function GetNthEffectDuration(int index) native

; return the magic effect of the specified effect
MagicEffect Function GetNthEffectMagicEffect(int index) native

; return the index of the costliest effect
int Function GetCostliestEffectIndex() native

; sets the magnitude of the specified effect
Function SetNthEffectMagnitude(int index, float value) native

; sets the area of the specified effect
Function SetNthEffectArea(int index, int value) native

; sets the duration of the specified effect
Function SetNthEffectDuration(int index, int value) native

; returns the base enchantment of this enchantment
Enchantment Function GetBaseEnchantment() native

; Returns a Formlist of Keywords
FormList Function GetKeywordRestrictions() native

; Sets the FormList of keywords
Function SetKeywordRestrictions(FormList newKeywordList) native   :: Add a newline
between files
Scriptname EquipSlot extends Form Hidden

; Returns the number of parent slots
int Function GetNumParents() native

; Returns the Nth parent slot
EquipSlot Function GetNthParent(int n) native   :: Add a newline between files
Scriptname Faction extends Form Hidden

; Checks to see if the player can pay the crime gold for this faction
bool Function CanPayCrimeGold() native

; Gets the amount of gold the player is to pay to this faction for crimes
int Function GetCrimeGold() native

; Gets the amount of gold the player is to pay to this faction for non-violent crimes
int Function GetCrimeGoldNonViolent() native

; Gets the amount of gold the player is to pay to this faction for violent crimes
int Function GetCrimeGoldViolent() native

; Get the player's "infamy" with this faction (accumulated crime gold)
int Function GetInfamy() native

; Get the player's "non-violent infamy" with this faction (accumulated non-violent crime
gold)
int Function GetInfamyNonViolent() native

; Get the player's "violent infamy" with this faction (accumulated violent crime gold)
int Function GetInfamyViolent() native

; Gets this faction's reaction towards the other
```

```
int Function GetReaction(Faction akOther) native

; Obtains the value of all items stolen by the player from this faction that was witnessed
int Function GetStolenItemValueCrime() native

; Obtains the value of all items stolen by the player from this faction that was NOT witnessed
int Function GetStolenItemValueNoCrime() native

; Is the passed in faction in this faction's crime group
bool Function IsFactionInCrimeGroup(Faction akOther) native

; Is the player expelled from this faction?
bool Function IsPlayerExpelled() native

; Modifies the amount of crime gold for this faction - violent or non-violent
Function ModCrimeGold(int aiAmount, bool abViolent = false) native

; Modifies this faction's reaction towards the other faction
Function ModReaction(Faction akOther, int aiAmount) native

; Has the player pay the crime gold for this faction
Function PlayerPayCrimeGold(bool abRemoveStolenItems = true, bool abGoToJail = true) native

; Finds a nearby NPC in this faction and has them behave as if assaulted
Function SendAssaultAlarm() native

; Sends the player to this faction's jail - removing inventory if requested, and to a "real" jail or not
Function SendPlayerToJail(bool abRemoveInventory = true, bool abRealJail = true) native

; Sets this faction and the other as allies or friends - if the friend booleans are true - the specified one-way relationship
; is a friend instead of an ally
Function SetAlly(Faction akOther, bool abSelfIsFriendToOther = false, bool abOtherIsFriendToSelf = false) native

; Sets the non-violent crime gold on this faction
Function SetCrimeGold(int aiGold) native

; Sets the violent crime gold on this faction
Function SetCrimeGoldViolent(int aiGold) native

; Sets this faction and the other as enemies or neutral - if the friend booleans are true - the specified one-way relationship
; is a neutral instead of an enemy
Function SetEnemy(Faction akOther, bool abSelfIsNeutralToOther = false, bool abOtherIsNeutralToSelf = false) native

; Sets or clears the player as an enemy of this faction
Function SetPlayerEnemy(bool abIsEnemy = true) native

; Sets or clears the expelled flag for this faction on the player
Function SetPlayerExpelled(bool abIsExpelled = true) native

; Sets this faction's reaction towards the other
Function SetReaction(Faction akOther, int aiNewValue) native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

int property kFaction_HiddenFromNPC            = 0x00000001 AutoReadOnly
int property kFaction_SpecialCombat            = 0x00000002 AutoReadOnly
int property kFaction_TrackCrime               = 0x00000010 AutoReadOnly
int property kFaction_IgnoreMurder             = 0x00000020 AutoReadOnly
int property kFaction_IgnoreAssault            = 0x00000040 AutoReadOnly
int property kFaction_IgnoreStealing           = 0x00000080 AutoReadOnly
int property kFaction_IgnoreTrespass           = 0x00000100 AutoReadOnly
int property kFaction_NoReportCrime              = 0x00000200 AutoReadOnly
int property kFaction_CrimeGoldDefaults          = 0x00000400 AutoReadOnly
int property kFaction_IgnorePickpocket           = 0x00000800 AutoReadOnly
int property kFaction_Vendor                     = 0x00001000 AutoReadOnly
int property kFaction_CanBeOwner                 = 0x00002000 AutoReadOnly
int property kFaction_IgnoreWerewolf             = 0x00004000 AutoReadOnly

; Not recommended unless the faction was previously a vendor
; due to the faction not having a package location the vendor
; may not be able to set up shop anywhere at all
Function MakeVendor()
    SetFactionFlag(self.kFaction_Vendor)
EndFunction

bool Function IsVendor()
    return IsFactionFlagSet(self.kFaction_Vendor)
EndFunction

Function ClearVendor()
    ClearFactionFlag(self.kFaction_Vendor)
EndFunction

bool Function IsFactionFlagSet(int flag) native
Function SetFactionFlag(int flag) native
Function ClearFactionFlag(int flag) native

bool Function OnlyBuysStolenItems() native
Function SetOnlyBuysStolenItems(bool onlyStolen) native

int Function GetVendorStartHour() native
Function SetVendorStartHour(int hour) native

int Function GetVendorEndHour() native
Function SetVendorEndHour(int hour) native

int Function GetVendorRadius() native
Function SetVendorRadius(int radius) native

ObjectReference Function GetMerchantContainer() native
Function SetMerchantContainer(ObjectReference akContainer) native

bool Function IsNotSellBuy() native
Function SetNotSellBuy(bool notSellBuy) native

FormList Function GetBuySellList() native
Function SetBuySellList(FormList akList) native   :: Add a newline between files
Scriptname Flora extends Activator Hidden


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
SoundDescriptor Function GetHarvestSound() native
Function SetHarvestSound(SoundDescriptor akSoundDescriptor) native

Form Function GetIngredient() native
Function SetIngredient(Form akIngredient) native   :: Add a newline between files
Scriptname Form Hidden

; Returns the formID for this object
Int Function GetFormID() native

; Obtains this form's value in gold. Will return -1 if the form doesn't have any value
(like a quest)
int Function GetGoldValue() native

; Returns if this form has the specified keyword attached
bool Function HasKeyword(Keyword akKeyword) native

; Is the "Known" flag set for this form?
bool Function PlayerKnows() native
```

```

; Register for the specified animation event from the specified object - returns true if it successfully registered
bool Function RegisterForAnimationEvent(ObjectReference akSender, string asEventName) native

; Register for LOS gain and lost events between the viewer and the target
; A loss or gain event will be sent immediately, depending on whether or not the viewer is already looking at the target or not
; If the viewer is not the player, the target must be another actor
Function RegisterForLOS(Actor akViewer, ObjectReference akTarget) native

; Register for only the first LOS gain event between the viewer and the target
; If the viewer is already looking at the target, an event will be received almost immediately
; If the viewer is not the player, the target must be another actor
Function RegisterForSingleLOSGain(Actor akViewer, ObjectReference akTarget) native

; Register for only the first LOS lost event between the viewer and the target
; If the viewer is already not looking at the target, an event will be received almost immediately
; If the viewer is not the player, the target must be another actor
Function RegisterForSingleLOSLost(Actor akViewer, ObjectReference akTarget) native

; Register for a single OnUpdate event, in afInterval seconds. All scripts attached to this form will get the update events
; Of course, this means you don't need to call UnregisterForUpdate()
; If you find yourself doing this:
; Event OnUpdate()
;     UnregisterForUpdate()
;     {Do some stuff}
; endEvent
; Then you should use RegisterForSingleUpdate instead
Function RegisterForSingleUpdate(float afInterval) native

; Registers this form to receive events when the player sleeps and wakes up
Function RegisterForSleep() native

; Registers this form to receive events when tracked stats are updated
Function RegisterForTrackedStatsEvent() native

; Register for OnUpdate events, every X seconds, where X is the interval. All scripts attached to this form will get the update events
Function RegisterForUpdate(float afInterval) native

; Register for OnUpdateGameTime events, every X hours of game time, where X is the interval. All scripts attached to this form will get the update events
Function RegisterForUpdateGameTime(float afInterval) native

; Register for a single OnUpdateGameTime event, in afInterval hours of game time. All scripts attached to this form will get the update events
Function RegisterForSingleUpdateGameTime(float afInterval) native

; Turns on profiling for this specific object and all scripts attached to it - setting doesn't persist across saves
; Will do nothing on release console builds, and if the Papyrus:bEnableProfiling ini setting is off
Function StartObjectProfiling() native

; Turns off profiling for this specific object and all scripts attached to it - setting doesn't persist across saves
; Will do nothing on release console builds, and if the Papyrus:bEnableProfiling ini setting is off
Function StopObjectProfiling() native

; Unregister for the specified animation event from the specified object
Function UnregisterForAnimationEvent(ObjectReference akSender, string asEventName) native

; Unregister for any LOS events between the viewer and target
```

```
Function UnregisterForLOS(Actor akViewer, ObjectReference akTarget) native

; Unregisters this form to receive events when the player sleeps and wakes up
Function UnregisterForSleep() native

; Unregisters this form from receiving events when tracked stats are updated
Function UnregisterForTrackedStatsEvent() native

; Unregister for OnUpdate events, all attached scripts will stop getting update events
Function UnregisterForUpdate() native

; Unregister for OnUpdateGameTime events, all attached scripts will stop getting update
game time events
Function UnregisterForUpdateGameTime() native

; Animation event, sent when an object we are listening to hits one of the events we are
listening for
Event OnAnimationEvent(ObjectReference akSource, string asEventName)
EndEvent

; Event sent when you have been unregistered from receiving an animation event because
the target
; object's animation graph has been unloaded
Event OnAnimationEventUnregistered(ObjectReference akSource, string asEventName)
EndEvent

; LOS event, sent whenever the viewer first sees the target (after registering)
Event OnGainLOS(Actor akViewer, ObjectReference akTarget)
EndEvent

; Lost LOS event, sent whenever the viewer first loses sight of the target (after
registering)
Event OnLostLOS(Actor akViewer, ObjectReference akTarget)
EndEvent

; Received when the player sleeps. Start and desired end time are in game time days
(after registering)
Event OnSleepStart(float afSleepStartTime, float afDesiredSleepEndTime)
EndEvent

; Received when the player stops sleeping - whether naturally or interrupted (after
registering)
Event OnSleepStop(bool abInterrupted)
EndEvent

; Event received when a tracked stat is updated for the player
Event OnTrackedStatsEvent(string arStatName, int aiStatValue)
EndEvent

; Update event, sent every X seconds while this form is registered for them
Event OnUpdate()
EndEvent

; Update event, sent every X hours of game time while this form is registered for them
Event OnUpdateGameTime()
EndEvent

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

; Returns the typecode for this form object
Int Function GetType() native

; returns the form's name, full name if possible
string Function GetName() native

; sets the name of the form
Function SetName(string name) native

; returns the weight of the form
```

```
float Function GetWeight() native

; sets the weight of the form
Function SetWeight(float weight) native

; sets the gold value of the form
Function SetGoldValue(int value) native

; returns the number of keywords on the form
int Function GetNumKeywords() native

; returns the keyword at the specified index
Keyword Function GetNthKeyword(int index) native

; returns all keywords of the form
Keyword[] Function GetKeywords() native

bool Function HasKeywordString(string s)
    Keyword k = Keyword.GetKeyword(s)
    if k == None
        return false
    endif
    return HasKeyword(k)
endFunction

; Sets whether the player knows this form
; Should only be used for Magic Effects,
; Words of Power, and Enchantments
Function SetPlayerKnows(bool knows) native

; Registers for OnKeyDown and OnKeyUp events for the given keycode.
Function RegisterForKey(int keyCode) native
Function UnregisterForKey(int keyCode) native
Function UnregisterForAllKeys() native

Event OnKeyDown(int keyCode)
EndEvent

Event OnKeyUp(int keyCode, float holdTime)
EndEvent

; Registers for OnControlDown and OnControlUp events for the given control.
; For a list of valid controls, see Input.psc.
Function RegisterForControl(string control) native
Function UnregisterForControl(string control) native
Function UnregisterForAllControls() native

Event OnControlDown(string control)
EndEvent

Event OnControlUp(string control, float holdTime)
EndEvent

; Registers for OnMenuOpen and OnMenuClose events for the given menu.
; Registrations have to be refreshed after each game load.
; For a list of valid menu names, see UI.psc.
Function RegisterForMenu(string menuName) native
Function UnregisterForMenu(string menuName) native
Function UnregisterForAllMenus() native

Event OnMenuOpen(string menuName)
endEvent

Event OnMenuClose(string menuName)
endEvent

; Registers a custom event callback for given event name.
; Registrations have to be refreshed after each game load.
;
```

```
;   Examples:
;       RegisterForModEvent("myCustomEvent", "MyModEventCallback")
;
;   Event signature of custom event callbacks:
;       Event MyModEventCallback(string eventName, string strArg, float numArg, Form
sender)
;       endEvent
;
Function RegisterForModEvent(string eventName, string callbackName) native
Function UnregisterForModEvent(string eventName) native
Function UnregisterForAllModEvents() native

; Sends custom event with given generic parameters.
Function SendModEvent(string eventName, string strArg = "", float numArg = 0.0) native

; Registers for OnPlayerCameraState events
Function RegisterForCameraState() native
Function UnregisterForCameraState() native

Event OnPlayerCameraState(int oldState, int newState)
EndEvent

; Registers for OnCrosshairRefChange events
Function RegisterForCrosshairRef() native
Function UnregisterForCrosshairRef() native

; Note: ref is none for no target
Event OnCrosshairRefChange(ObjectReference ref)
EndEvent

Function RegisterForActorAction(int actionType) native
Function UnregisterForActorAction(int actionType) native

; ActionTypes
; 0 - Weapon Swing (Melee weapons that are swung, also barehand)
; 1 - Spell Cast (Spells and staves)
; 2 - Spell Fire (Spells and staves)
; 3 - Voice Cast
; 4 - Voice Fire
; 5 - Bow Draw
; 6 - Bow Release
; 7 - Unsheathe Begin
; 8 - Unsheathe End
; 9 - Sheathe Begin
; 10 - Sheathe End
; Slots
; 0 - Left Hand
; 1 - Right Hand
; 2 - Voice
Event OnActorAction(int actionType, Actor akActor, Form source, int slot)
EndEvent

; Registers the script for when a QueueNiNodeUpdate is called
Function RegisterForNiNodeUpdate() native
Function UnregisterForNiNodeUpdate() native

Event OnNiNodeUpdate(ObjectReference akActor)
EndEvent

; Returns a temporary clone of this form
Form Function TempClone() native

; Returns whether this Form has a World Model (fast)
bool Function HasWorldModel() native

; Returns the world model path of this Form, if it has a world model
string Function GetWorldModelPath() native
Function SetWorldModelPath(string path) native

```

```
; Returns the number of texture sets the world model has, if its textures can be swapped
int Function GetWorldModelNumTextureSets() native

; Returns the Nth texture set of the world model, if the textures can be swapped
TextureSet Function GetWorldModelNthTextureSet(int n) native

; Sets the world models Nth texture set, if the textures can be set
Function SetWorldModelNthTextureSet(TextureSet nSet, int n) native

; Returns whether this Form is playable, only applied to Forms with the playable flag
bool Function IsPlayable() native   :: Add a newline between files
Scriptname FormList extends Form

; Adds the given form to this form list
Function AddForm(Form apForm) native

; Finds the specified form in the form list and returns its index.
; If not found, returns a negative number
int Function Find(Form apForm) native

; Returns the number of forms in the list
int Function GetSize() native

; Returns the form at index 'aiIndex' in the list
Form Function GetAt(int aiIndex) native

; Queries the form list to see if it contains the passed in form
bool Function HasForm(Form akForm) native

; Removes the given added form from this form list
Function RemoveAddedForm(Form apForm) native

; Removes all script added forms from this form list
Function Revert() native


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; Returns a Form array of this list (Invalid entries will be None)
Form[] Function ToArray() native

; Adds an Array of Forms to this list
Function AddForms(Form[] forms) native   :: Add a newline between files
Scriptname FormType Hidden

int Property kNone =    0 AutoReadOnly
int Property kTES4 = 1 AutoReadOnly
int Property kGroup = 2 AutoReadOnly
int Property kGMST = 3 AutoReadOnly
int Property kKeyword = 4 AutoReadOnly
int Property kLocationRef = 5 AutoReadOnly
int Property kAction = 6 AutoReadOnly
int Property kTextureSet = 7 AutoReadOnly
int Property kMenuIcon = 8 AutoReadOnly
int Property kGlobal = 9 AutoReadOnly
int Property kClass = 10 AutoReadOnly
int Property kFaction = 11 AutoReadOnly
int Property kHeadPart = 12 AutoReadOnly
int Property kEyes = 13 AutoReadOnly
int Property kRace = 14 AutoReadOnly
int Property kSound = 15 AutoReadOnly
int Property kAcousticSpace = 16 AutoReadOnly
int Property kSkill = 17 AutoReadOnly
int Property kEffectSetting = 18 AutoReadOnly
int Property kScript = 19 AutoReadOnly
int Property kLandTexture = 20 AutoReadOnly
int Property kEnchantment = 21 AutoReadOnly
int Property kSpell = 22 AutoReadOnly
int Property kScrollItem = 23 AutoReadOnly
int Property kActivator = 24 AutoReadOnly
int Property kTalkingActivator = 25 AutoReadOnly
int Property kArmor = 26 AutoReadOnly
int Property kBook = 27 AutoReadOnly
int Property kContainer = 28 AutoReadOnly
int Property kDoor = 29 AutoReadOnly
int Property kIngredient = 30 AutoReadOnly
int Property kLight = 31 AutoReadOnly
int Property kMisc = 32 AutoReadOnly
int Property kApparatus = 33 AutoReadOnly
int Property kStatic = 34 AutoReadOnly
int Property kStaticCollection = 35 AutoReadOnly
int Property kMovableStatic = 36 AutoReadOnly
int Property kGrass = 37 AutoReadOnly
int Property kTree = 38 AutoReadOnly
int Property kFlora = 39 AutoReadOnly
int Property kFurniture = 40 AutoReadOnly
int Property kWeapon = 41 AutoReadOnly
int Property kAmmo = 42 AutoReadOnly
int Property kNPC = 43 AutoReadOnly
int Property kLeveledCharacter = 44 AutoReadOnly
int Property kKey = 45 AutoReadOnly
int Property kPotion = 46 AutoReadOnly
int Property kIdleMarker = 47 AutoReadOnly
int Property kNote = 48 AutoReadOnly
int Property kConstructibleObject = 49 AutoReadOnly
int Property kProjectile = 50 AutoReadOnly
int Property kHazard = 51 AutoReadOnly
int Property kSoulGem = 52 AutoReadOnly
int Property kLeveledItem = 53 AutoReadOnly
int Property kWeather = 54 AutoReadOnly
int Property kClimate = 55 AutoReadOnly
int Property kShaderParticleGeometryData = 56 AutoReadOnly
int Property kReferenceEffect = 57 AutoReadOnly
int Property kRegion = 58 AutoReadOnly
int Property kNAVI = 59 AutoReadOnly
int Property kCell = 60 AutoReadOnly
int Property kReference = 61 AutoReadOnly
int Property kCharacter = 62 AutoReadOnly
int Property kMissile = 63 AutoReadOnly
int Property kArrow = 64 AutoReadOnly
int Property kGrenade = 65 AutoReadOnly
int Property kBeamProjectile = 66 AutoReadOnly
int Property kFlameProjectile = 67 AutoReadOnly
int Property kConeProjectile = 68 AutoReadOnly
int Property kBarrierProjectile = 69 AutoReadOnly
int Property kPHZD = 70 AutoReadOnly
int Property kWorldSpace = 71 AutoReadOnly
int Property kLand = 72 AutoReadOnly
int Property kNavMesh = 73 AutoReadOnly
int Property kTLOD = 74 AutoReadOnly
int Property kTopic = 75 AutoReadOnly
int Property kTopicInfo = 76 AutoReadOnly
int Property kQuest = 77 AutoReadOnly
int Property kIdle = 78 AutoReadOnly
int Property kPackage = 79 AutoReadOnly
int Property kCombatStyle = 80 AutoReadOnly
int Property kLoadScreen = 81 AutoReadOnly
int Property kLeveledSpell = 82 AutoReadOnly
int Property kANIO = 83 AutoReadOnly
int Property kWater = 84 AutoReadOnly
int Property kEffectShader = 85 AutoReadOnly
int Property kTOFT = 86 AutoReadOnly
int Property kExplosion = 87 AutoReadOnly
int Property kDebris = 88 AutoReadOnly
int Property kImageSpace = 89 AutoReadOnly
int Property kImageSpaceModifier = 90 AutoReadOnly
int Property kList = 91 AutoReadOnly
int Property kPerk = 92 AutoReadOnly
int Property kBodyPartData = 93 AutoReadOnly
int Property kAddonNode = 94 AutoReadOnly
int Property kActorValueInfo = 95 AutoReadOnly
int Property kCameraShot = 96 AutoReadOnly
int Property kCameraPath = 97 AutoReadOnly
int Property kVoiceType = 98 AutoReadOnly
int Property kMaterialType = 99 AutoReadOnly
int Property kImpactData = 100 AutoReadOnly
int Property kImpactDataSet = 101 AutoReadOnly
int Property kARMA = 102 AutoReadOnly
int Property kEncounterZone = 103 AutoReadOnly
int Property kLocation = 104 AutoReadOnly
int Property kMessage = 105 AutoReadOnly
int Property kRagdoll = 106 AutoReadOnly
int Property kDefaultObject = 107 AutoReadOnly
int Property kLightingTemplate = 108 AutoReadOnly
int Property kMusicType = 109 AutoReadOnly
int Property kFootstep = 110 AutoReadOnly
int Property kFootstepSet = 111 AutoReadOnly
int Property kStoryBranchNode = 112 AutoReadOnly
int Property kStoryQuestNode = 113 AutoReadOnly
int Property kStoryEventNode = 114 AutoReadOnly
int Property kDialogueBranch = 115 AutoReadOnly
int Property kMusicTrack = 116  AutoReadOnly
int Property kDLVW = 117 AutoReadOnly
int Property kWordOfPower = 118 AutoReadOnly
int Property kShout = 119 AutoReadOnly
int Property kEquipSlot = 120 AutoReadOnly
int Property kRelationship = 121 AutoReadOnly
int Property kScene = 122 AutoReadOnly
int Property kAssociationType = 123 AutoReadOnly
int Property kOutfit = 124 AutoReadOnly
int Property kArt = 125 AutoReadOnly
int Property kMaterial = 126 AutoReadOnly
int Property kMovementType = 127 AutoReadOnly
int Property kSoundDescriptor = 128 AutoReadOnly
int Property kDualCastData = 129 AutoReadOnly
int Property kSoundCategory = 130 AutoReadOnly
int Property kSoundOutput = 131 AutoReadOnly
int Property kCollisionLayer = 132 AutoReadOnly
int Property kColorForm = 133 AutoReadOnly
int Property kReverbParam = 134 AutoReadOnly
   :: Add a newline between files
Scriptname Game Hidden

; Adds the specified achievement to the player's profile
Function AddAchievement(int aiAchievementID) native global

; Add the specified number of perk points to the player
Function AddPerkPoints(int aiPerkPoints) native global

; Advance the given skill on the player by the provided amount of skill usage
Function AdvanceSkill(string asSkillName, float afMagnitude) native global

; Adds a ball-and-socket constraint between two rigid bodies, identified by their ref and node names
bool Function AddHavokBallAndSocketConstraint( ObjectReference arRefA, string arRefANode, ObjectReference arRefB, string arRefBNode, float afRefALocalOffsetX = 0.0, float afRefALocalOffsetY = 0.0, float afRefALocalOffsetZ = 0.0, float afRefBLocalOffsetX = 0.0, float afRefBLocalOffsetY = 0.0, float afRefBLocalOffsetZ = 0.0) native global

; Removes any constraint between two rigid bodies
bool Function RemoveHavokConstraints(ObjectReference arFirstRef, string arFirstRefNodeName, ObjectReference arSecondRef, string arSecondRefNodeName) native global

; Calculates how much a x point favor would cost the player
int Function CalculateFavorCost(int aiFavorPrice) native global

; Clears the prison variables on the player
```

```
Function ClearPrison() native global

; Clears temp effects from game
Function ClearTempEffects() native global

; Disables the user's controls
Function DisablePlayerControls(bool abMovement = true, bool abFighting = true, bool abCamSwitch = false, bool abLooking = false, \
  bool abSneaking = false, bool abMenu = true, bool abActivate = true, bool abJournalTabs = false, int aiDisablePOVType = 0) native global

; Enables the user's controls
Function EnablePlayerControls(bool abMovement = true, bool abFighting = true, bool abCamSwitch = true, bool abLooking = true, \
  bool abSneaking = true, bool abMenu = true, bool abActivate = true, bool abJournalTabs = true, int aiDisablePOVType = 0) native global

; Enables or disables the ability to fast travel
Function EnableFastTravel(bool abEnable = true) native global

; Fades out the game to black, or vice versa
Function FadeOutGame(bool abFadingOut, bool abBlackFade, float afSecsBeforeFade, float afFadeDuration) native global

; Fast-travels the player to the specified object's location
Function FastTravel(ObjectReference akDestination) native global

; Finds the closest reference of a given base object within a given radius of a location
ObjectReference Function FindClosestReferenceOfType(Form arBaseObject, float afX, float afY, float afZ, float afRadius) native global

; Finds a random reference of a given base object within a given radius of a location
ObjectReference Function FindRandomReferenceOfType(Form arBaseObject, float afX, float afY, float afZ, float afRadius) native global

; Finds the closest reference of any base object in the list within a given radius of a location
ObjectReference Function FindClosestReferenceOfAnyTypeInList(FormList arBaseObjects, float afX, float afY, float afZ, float afRadius) native global

; Finds a random reference of a any base object in the list within a given radius of a location
ObjectReference Function FindRandomReferenceOfAnyTypeInList(FormList arBaseObjects, float afX, float afY, float afZ, float afRadius) native global

; Finds the closest reference of a given base object within a given radius of a reference
ObjectReference Function FindClosestReferenceOfTypeFromRef(Form arBaseObject, ObjectReference arCenter, float afRadius) global
    return FindClosestReferenceOfType(arBaseObject, arCenter.X, arCenter.Y, arCenter.Z, afRadius)
endFunction

; Finds a random reference of a given base object within a given radius of a reference
ObjectReference Function FindRandomReferenceOfTypeFromRef(Form arBaseObject, ObjectReference arCenter, float afRadius) global
    return FindRandomReferenceOfType(arBaseObject, arCenter.X, arCenter.Y, arCenter.Z, afRadius)
endFunction

; Finds the closest reference of a given base object within a given radius of a reference
ObjectReference Function FindClosestReferenceOfAnyTypeInListFromRef(FormList arBaseObjects, ObjectReference arCenter, float afRadius) global
    return FindClosestReferenceOfAnyTypeInList(arBaseObjects, arCenter.X, arCenter.Y, arCenter.Z, afRadius)
endFunction

; Finds a random reference of a given base object within a given radius of a reference
ObjectReference Function FindRandomReferenceOfAnyTypeInListFromRef(FormList arBaseObjects, ObjectReference arCenter, float afRadius) global
```

```
    return FindRandomReferenceOfAnyTypeInList(arBaseObjects, arCenter.X, arCenter.Y,
    arCenter.Z, afRadius)
endFunction

; Finds the closest actor within a given radius of a location
Actor Function FindClosestActor(float afX, float afY, float afZ, float afRadius) native
global

; Finds a random actor within a given radius of a location
Actor Function FindRandomActor(float afX, float afY, float afZ, float afRadius) native
global

; Finds the closest actor within a given radius of a reference
Actor Function FindClosestActorFromRef(ObjectReference arCenter, float afRadius) global
    return FindClosestActor(arCenter.X, arCenter.Y, arCenter.Z, afRadius)
endFunction

; Finds a random actor within a given radius of a reference
Actor Function FindRandomActorFromRef(ObjectReference arCenter, float afRadius) global
    return FindRandomActor(arCenter.X, arCenter.Y, arCenter.Z, afRadius)
endFunction

; Make the player got to 3rd person camera mode
Function ForceThirdPerson() native global

; Make the player got to 1st person camera mode
Function ForceFirstPerson() native global

; Show the players first person geometry.
Function ShowFirstPersonGeometry( bool abShow = true ) native global

; Returns the form specified by the ID
Form Function GetForm(int aiFormID) native global

; Returns the form specified by the ID originating in the given file
Form Function GetFormFromFile(int aiFormID, string asFilename) native global

; Obtains the value of a game setting - one for each type of game setting
float Function GetGameSettingFloat(string asGameSetting) native global
int Function GetGameSettingInt(string asGameSetting) native global
string Function GetGameSettingString(string asGameSetting) native global

; Returns the player actor
Actor Function GetPlayer() native global

; Returns the reference the player is currently grabbing
ObjectReference Function GetPlayerGrabbedRef() native global

; Returns the horse last ridden by the player
Actor Function GetPlayersLastRiddenHorse() native global

; Returns the X position of the Sun.
float Function GetSunPositionX() native global

; Returns the Y position of the Sun.
float Function GetSunPositionY() native global

; Returns the Z position of the Sun.
float Function GetSunPositionZ() native global

; Returns the number of days spent in play
float Function GetRealHoursPassed() native global

; Increment the given skill on the player by the one point
Function IncrementSkill(string asSkillName) native global

; Increment the given skill on the player by the given number of points
Function IncrementSkillBy(string asSkillName, int aiCount) native global

```

```
; Modifies the specified MiscStat by the given amount.
Function IncrementStat(string asStatName, int aiModAmount = 1) native global

; Are the activation controls enabled?
bool Function IsActivateControlsEnabled() native global

; Are the camera switch controls enabled?
bool Function IsCamSwitchControlsEnabled() native global

; Is fast travel controls enabled? Returns false if EnableFastTravel(false) has been
called
bool Function IsFastTravelControlsEnabled() native global

; Is fast travel enabled?
bool Function IsFastTravelEnabled() native global

; Are the fighting controls enabled?
bool Function IsFightingControlsEnabled() native global

; Are the journal menu controls enabled?
bool Function IsJournalControlsEnabled() native global

; Are the looking controls enabled?
bool Function IsLookingControlsEnabled() native global

; Are the menu controls enabled?
bool Function IsMenuControlsEnabled() native global

; Are the movement controls enabled?
bool Function IsMovementControlsEnabled() native global

; Is the player looking at the sun?
bool Function IsPlayerSungazing() native global

; Are the sneaking controls enabled?
bool Function IsSneakingControlsEnabled() native global

; Is the specified Word of Power Unlocked?
bool Function IsWordUnlocked(WordOfPower akWord) native global

; Plays a bink video - does not return until bink has finished, use with care!
Function PlayBink(string asFileName, bool abInterruptible = false, bool abMuteAudio =
true, bool abMuteMusic = true, \
  bool abLetterbox = true ) native global

; Precaches character gen data.
Function PrecacheCharGen() native global

; Clears Precached character gen data.
Function PrecacheCharGenClear() native global

; Queries the given stat and returns its value
int Function QueryStat(string asStat) native global

; Forces the game back to the main menu
Function QuitToMainMenu() native global

; Request that an auto-save be made
Function RequestAutoSave() native global

; Requests the specified model
Function RequestModel(string asModelName) native global

; Request that a normal save be made
Function RequestSave() native global

; Has the player serve their prison time
Function ServeTime() native global

```

```
; Finds an actor in high who can detect the player to call werewolf crime on the player
Function SendWereWolfTransformation() native global

; Called as we enter/exit beast form
Function SetBeastForm(bool abEntering) native global

; Sets the camera target actor
Function SetCameraTarget(Actor arTarget) native global

; Sets or clears "cart mode" for the HUD
Function SetHudCartMode(bool abSetCartMode = true) native global

; Informs the game whether we are in CharGen or not
Function SetInChargen(bool abDisableSaving, bool abDisableWaiting, bool abShowControlsDisabledMessage) native global

; Enables or disables the AI driven flag on Player
Function SetPlayerAIDriven(bool abAIDriven = true) native global

; Enables or disables  crime reporting on Player
Function SetPlayerReportCrime(bool abReportCrime = true) native global

; Set the players sitting camera rotation - in degrees, offset from the standard angle.
Function SetSittingRotation(float afValue) native global

; Shakes the object from the location of the passed-in object. If none, it will shake the camera from the player's location.
; Strength is clamped from 0 to 1
; Duration in seconds. By default (0.0) use the game setting.
Function ShakeCamera(ObjectReference akSource = None, float afStrength = 0.5, float afDuration = 0.0) native global

; Shakes the controller for the specified length of time (in seconds). The strength values are clamped from 0 to 1
Function ShakeController(float afSmallMotorStrength, float afBigMotorStreangth, float afDuration) native global

; Displays the race/sex menu
Function ShowRaceMenu() native global
Function ShowLimitedRaceMenu() native global

; Title Sequence menu functions
Function ShowTitleSequenceMenu() native global
Function HideTitleSequenceMenu() native global
Function StartTitleSequence(string asSequenceName) native global

; Allow or disallow player requests to have a flying mount land.
Function SetAllowFlyingMountLandingRequests(bool abAllow) native global

; Sets the Image Space Modifier that is triggered when the player gazes at the sun.
Function SetSunGazeImageSpaceModifier(ImageSpaceModifier apImod = NONE ) native global

; Displays the training menu based on passed in trainer actor
Function ShowTrainingMenu(Actor aTrainer) native global

; Teaches the specified word of power to the player
Function TeachWord(WordOfPower akWord) native global

; Trigger screen blood with the given count
Function TriggerScreenBlood(int aiValue) native global

; Unlocks the specified word of power so the player can use it
Function UnlockWord(WordOfPower akWord) native global

; Returns true if we're using a gamepad
bool Function UsingGamepad() native global


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; Get/Set Perk Points
int Function GetPerkPoints() global native
Function SetPerkPoints(int perkPoints) global native
Function ModPerkPoints(int perkPoints) global native

; returns the number of active mods
int Function GetModCount() native global

; returns the index of the specified mod
int Function GetModByName(string name) native global

; returns the name of the mod at the specified modIndex
string Function GetModName(int modIndex) native global

; returns the author of the mod at the specified modIndex
string Function GetModAuthor(int modIndex) native global

; returns the description of the mod at the specified modIndex
string Function GetModDescription(int modIndex) native global

; gets the count of mods the specified mod depends upon
int Function GetModDependencyCount(int modIndex) native global

; gets the index of the nth mod dependency of the specfied mod
; int Function GetNthModDependency(int modIndex, int n) native global

bool Function IsPluginInstalled(string name) native global

; light mod functions
int Function GetLightModCount() native global
int Function GetLightModByName(string name) native global
string Function GetLightModName(int idx) native global
string Function GetLightModAuthor(int idx) native global
string Function GetLightModDescription(int idx) native global
int Function GetLightModDependencyCount(int idx) native global
int Function GetNthLightModDependency(int modIdx, int idx) native global

; GameSetting functions - SKSE 1.5.10
Function SetGameSettingFloat(string setting, float value) global native
Function SetGameSettingInt(string setting, int value) global native
Function SetGameSettingBool(string setting, bool value) global native
Function SetGameSettingString(string setting, string value) global native

; save/load game
Function SaveGame(string name) native global
Function LoadGame(string name) native global

; TintMasks (AARRGGBB)

; Returns the total number of tints for the player
int Function GetNumTintMasks() native global

; Returns the color of the Nth tint mask
int Function GetNthTintMaskColor(int n) native global

; Returns the type of the Nth tint mask
int Function GetNthTintMaskType(int n) native global

; Sets the color of the Nth tint mask
Function SetNthTintMaskColor(int n, int color) native global

; Returns the texture path of the Nth tint mask
string Function GetNthTintMaskTexturePath(int n) native global

; Sets the texturepath of the Nth tint mask
Function SetNthTintMaskTexturePath(string path, int n) native global

; Types
; 0 - Frekles
; 1 - Lips
; 2 - Cheeks
; 3 - Eyeliner
; 4 - Upper Eyesocket
; 5 - Lower Eyesocket
; 6 - SkinTone
; 7 - Warpaint
; 8 - Frownlines
; 9 - Lower Cheeks
; 10 - Nose
; 11 - Chin
; 12 - Neck
; 13 - Forehead
; 14 - Dirt

; Returns how many indexes there are for this type
int Function GetNumTintsByType(int type) native global

; Returns the color for the particular tintMask type and index
int Function GetTintMaskColor(int type, int index) global native

; Sets the tintMask color for the particular type and index
Function SetTintMaskColor(int color, int type, int index) global native

; Returns the texture path for the particular tintMask type and index
string Function GetTintMaskTexturePath(int type, int index) global native

; Sets the tintMask texture for the particular type and index
Function SetTintMaskTexturePath(string path, int type, int index) global native

; Updates tintMask colors without updating the entire model
Function UpdateTintMaskColors() global native

; Updates the players hair color immediately
Function UpdateHairColor() global native

; Returns the character's current camera state
; 0 - first person
; 1 - auto vanity
; 2 - VATS
; 3 - free
; 4 - iron sights
; 5 - furniture
; 6 - transition
; 7 - tweenmenu
; 8 - third person 1
; 9 - third person 2
; 10 - horse
; 11 - bleedout
; 12 - dragon
int Function GetCameraState() global
    return Camera.GetCameraState()
EndFunction

; set a misc stat value
; use QueryStat to read the value
Function SetMiscStat(string name, int value) global native

; Sets the players last ridden horse, None will clear the lastRiddenHorse
Function SetPlayersLastRiddenHorse(Actor horse) global native

; Returns the legendary level for the skill
; -1 indicates the particular skill cannot have a legendary level
; DEPRECATED
int Function GetSkillLegendaryLevel(string actorValue) global
    return ActorValueInfo.GetActorValueInfoByName(actorValue).GetSkillLegendaryLevel()
EndFunction

; Sets the legendary level for the skill
; DEPRECATED
Function SetSkillLegendaryLevel(string actorValue, int level) global
    ActorValueInfo.GetActorValueInfoByName(actorValue).SetSkillLegendaryLevel(level)
EndFunction

; Returns the players experience for this level (not total experience)
float Function GetPlayerExperience() global native

; Sets the players experience, does not trigger level-up notification
Function SetPlayerExperience(float exp) global native

; Calculates the experience required for to level-up
; (fXPLevelUpBase + currentLevel * fXPLevelUpMult)
float Function GetExperienceForLevel(int currentLevel) global native

; Returns true if in run mode, false if in walk mode
; Does not reflect actual movement state, only the control mode
bool Function GetPlayerMovementMode() global native

; Updates the camera when changing Shoulder positions
Function UpdateThirdPerson() global
    Camera.UpdateThirdPerson()
EndFunction

; Hotkeys 0-7 reflect keys 1-8
; Unbinds a favorited item bound to the specified hotkey
Function UnbindObjectHotkey(int hotkey) global native

; Returns the base form object that is bound to the specified hotkey
Form Function GetHotkeyBoundObject(int hotkey) global native

; Returns if base form is favorited by the player
bool Function IsObjectFavorited(Form form) global native

; Same as GetForm, but also works for formIds >= 0x80000000
Form Function GetFormEx(int formId) global native

; Returns the object reference the player is in dialogue with
ObjectReference Function GetDialogueTarget() global native

; Returns the current crosshair ref
ObjectReference Function GetCurrentCrosshairRef() global native

; Returns the currently selected ref in the console
ObjectReference Function GetCurrentConsoleRef() global native

; Sets the player level
Function SetPlayerLevel(int level) global native   :: Add a newline between files
Scriptname GameData Hidden

; Keywords are AND operations, must have all listed keywords
; IgnoreTemplates will exclude items that are inherited from other items with slightly
altered stats
; IgnoreEnchantments will exclude any item with an enchantment
; WeaponTypes are a bitfield, will filter weapons by type
; Add together to filter by multiple types
int Property WeaponTypeHandToHand = 1 AutoReadOnly
int Property WeaponTypeOneHandSword = 2 AutoReadOnly
int Property WeaponTypeOneHandDagger = 4 AutoReadOnly
int Property WeaponTypeOneHandAxe = 8 AutoReadOnly
int Property WeaponTypeOneHandMace = 16 AutoReadOnly
int Property WeaponTypeTwoHandSword = 32 AutoReadOnly
int Property WeaponTypeTwoHandAxe = 64 AutoReadOnly
int Property WeaponTypeBow = 128 AutoReadOnly
int Property WeaponTypeStaff = 256 AutoReadOnly
int Property WeaponTypeCrossbow = 512 AutoReadOnly

Form[] Function GetAllWeapons(string modName, Keyword[] keywords = None, bool playable =
true, bool ignoreTemplates = true, bool ignoreEnchantments = true, bool onlyEnchanted =
false, int weaponTypes = 0xFFFFFFFF) global native

Form[] Function GetAllArmor(string modName, Keyword[] keywords = None, bool playable = true, bool ignoreTemplates = true, bool ignoreEnchantments = true, bool onlyEnchanted = false, bool ignoreSkin = true) global native

Form[] Function GetAllAmmo(string modName, Keyword[] keywords = None, bool playable = true) global native

Form[] Function GetAllBooks(string modName, Keyword[] keywords = None, bool regular = true, bool spell = false, bool skill = false) global native

Form[] Function GetAllPotions(string modName, Keyword[] keywords = None, bool potions = true, bool food = false, bool poison = false) global native

Form[] Function GetAllIngredients(string modName, Keyword[] keywords = None) global native

Form[] Function GetAllScrolls(string modName, Keyword[] keywords = None) global native

Form[] Function GetAllKeys(string modName, Keyword[] keywords = None) global native

Form[] Function GetAllMiscItems(string modName, Keyword[] keywords = None) global native   :: Add a newline between files
Scriptname HeadPart extends Form Hidden

int Property Type_Misc = 0 AutoReadOnly
int Property Type_Face = 1 AutoReadOnly
int Property Type_Eyes = 2 AutoReadOnly
int Property Type_Hair = 3 AutoReadOnly
int Property Type_FacialHair = 4 AutoReadOnly
int Property Type_Scar = 5 AutoReadOnly
int Property Type_Brows = 6 AutoReadOnly

HeadPart Function GetHeadPart(string name) native global

; Returns the head part type
int Function GetType() native

int Function GetNumExtraParts() native
HeadPart Function GetNthExtraPart(int n) native

bool Function HasExtraPart(HeadPart p) native
int Function GetIndexOfExtraPart(HeadPart p) native

; Returns a formlist of the valid races for this head part
FormList Function GetValidRaces() native
Function SetValidRaces(FormList vRaces) native

; Returns whether the head part is an extra part
bool Function IsExtraPart() native

; Returns the EditorID of the HeadPart
string Function GetPartName() native   :: Add a newline between files
Scriptname Ingredient extends Form

; Is this ingredient classified as hostile?
bool Function IsHostile() native

; Flags the effect with the given 0 based index as known by the player
Function LearnEffect(int aiIndex) native

; Flags the next unknown effect as known by the player, returning index of effect learned
int Function LearnNextEffect() native

; Flags the all effects as known by the player
Function LearnAllEffects() native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; return the number of the effects
int Function GetNumEffects() native

; return the magnitude of the specified effect
float Function GetNthEffectMagnitude(int index) native

; return the area of the specified effect
int Function GetNthEffectArea(int index) native

; return the duration of the specified effect
int Function GetNthEffectDuration(int index) native

; return the magic effect of the specified effect
MagicEffect Function GetNthEffectMagicEffect(int index) native

; return the index of the costliest effect
int Function GetCostliestEffectIndex() native

; sets the magnitude of the specified effect
Function SetNthEffectMagnitude(int index, float value) native

; sets the area of the specified effect
Function SetNthEffectArea(int index, int value) native

; sets the duration of the specified effect
Function SetNthEffectDuration(int index, int value) native

; determines whether the player knows this effect
bool Function GetIsNthEffectKnown(int index) native

; Returns all the magnitudes of this object in order
float[] Function GetEffectMagnitudes() native

; Returns all the areas of this object in order
int[] Function GetEffectAreas() native

; Returns all the durations of this object in order
int[] Function GetEffectDurations() native

; Returns all the magic effects of this object in order
MagicEffect[] Function GetMagicEffects() native   :: Add a newline between files
Scriptname Input Hidden

; returns whether a key is pressed
bool Function IsKeyPressed(Int dxKeycode) global native

; taps the specified key
Function TapKey(Int dxKeycode) global native

; holds down the specified key until released
Function HoldKey(Int dxKeycode) global native

; releases the specified key
Function ReleaseKey(Int dxKeycode) global native

; how many keys are pressed
int Function GetNumKeysPressed() global native

; for walking over the pressed keys
int Function GetNthKeyPressed(int n) global native

; returns keycode bound to a control for given device
;
; Valid controls:
;   "Forward", "Back", "Strafe Left", "Strafe Right", "Move", "Look", "Left
Attack/Block", "Right Attack/Block"
;   "Activate", "Ready Weapon", "Tween Menu", "Toggle POV", "Zoom Out", "Zoom In",
"Jump", "Sprint", "Shout",
;   "Sneak", "Run", "Toggle Always Run", "Auto-Move", "Favorites", "Hotkey1", "Hotkey2",
"Hotkey3", "Hotkey4",
;   "Hotkey5", "Hotkey6", "Hotkey7", "Hotkey8", "Quicksave", "Quickload", "Wait", "Journal", "Pause", "Screenshot",
;   "Multi-Screenshot", "Console", "CameraPath", "Quick Inventory", "Quick Magic", "Quick Stats", "Quick Map"
;
; Valid device types:
;   (default)   auto detect
;   0           keyboard
;   1           mouse
;   2           gamepad
int Function GetMappedKey(string control, int deviceType = 0xFF) global native

; returns name of control bound to given keycode, or "" if unbound
string Function GetMappedControl(int keycode) global native   :: Add a newline between files
Scriptname Keyword Extends Form Hidden

; Sends this keyword as a story event to the story manager
Function SendStoryEvent(Location akLoc = None, ObjectReference akRef1 = None, ObjectReference akRef2 = None, int aiValue1 = 0, \
    int aiValue2 = 0) native

; Sends this keyword as a story event to the story manager and waits for it to be processed. Returns true if a quest was started.
bool Function SendStoryEventAndWait(Location akLoc = None, ObjectReference akRef1 = None, ObjectReference akRef2 = None, \
    int aiValue1 = 0, int aiValue2 = 0) native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; return the keyword with the specified key
Keyword Function GetKeyword(string key) global native

; return the string value of the keyword
string Function GetString() native   :: Add a newline between files
Scriptname LeveledActor extends Form Hidden

; Adds the given count of the given form to the under the given level in this leveled list
Function AddForm(Form apForm, int aiLevel) native

; Removes all script added forms from this leveled list
Function Revert() native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
int Function GetNumForms() native
Form Function GetNthForm(int n) native

int Function GetNthLevel(int n) native
Function SetNthLevel(int n, int level) native

int Function GetNthCount(int n) native
Function SetNthCount(int n, int count) native   :: Add a newline between files
Scriptname LeveledItem extends Form Hidden

; Adds the given count of the given form to the under the given level in this leveled list
Function AddForm(Form apForm, int aiLevel, int aiCount) native

; Removes all script added forms from this leveled list
Function Revert() native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
int function GetChanceNone() native
Function SetChanceNone(int chance) native

GlobalVariable Function GetChanceGlobal() native
Function SetChanceGlobal(GlobalVariable glob) native

```

```
int Function GetNumForms() native
Form Function GetNthForm(int n) native

int Function GetNthLevel(int n) native
Function SetNthLevel(int n, int level) native

int Function GetNthCount(int n) native
Function SetNthCount(int n, int count) native   :: Add a newline between files
Scriptname LeveledSpell extends Form Hidden

; Adds the given count of the given form to the under the given level in this leveled
list
Function AddForm(Form apForm, int aiLevel) native

; Removes all script added forms from this leveled list
Function Revert() native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
int function GetChanceNone() native
Function SetChanceNone(int chance) native

int Function GetNumForms() native
Form Function GetNthForm(int n) native

int Function GetNthLevel(int n) native
Function SetNthLevel(int n, int level) native
   :: Add a newline between files
Scriptname Location extends Form Hidden

; Returns the float value attached to the specified keyword attached to this location
float Function GetKeywordData(Keyword akKeyword) native

; Returns the number of alive references matching the specified reference type
int Function GetRefTypeAliveCount(LocationRefType akRefType) native

; Returns the number of dead references matching the specified reference type
int Function GetRefTypeDeadCount(LocationRefType akRefType) native

; Returns if these two locations have a common parent - filtered with the keyword, if
provided
bool Function HasCommonParent(Location akOther, Keyword akFilter = None) native

; Returns if this location has the specified reference type
bool Function HasRefType(LocationRefType akRefType) native

; Returns whether this location is flagged as "cleared" or not
bool Function IsCleared() native

; Returns whether the other location is a child of this one
bool Function IsChild(Location akOther) native

; Is this location loaded in game?
bool Function IsLoaded() native

bool Function IsSameLocation(Location akOtherLocation, Keyword akKeyword = None)
{Returns true if the calling location is the same as the supplied location - if an
optional keyword is supplied, it also returns true if the locations share a parent with
that keyword, or if either location is a child of the other and the other has that
keyword.}
;jduvall
    bool bmatching = self == akOtherLocation
    if !bmatching && akKeyword
        bmatching = HasCommonParent(akOtherLocation, akKeyword)

        if !bmatching && akOtherLocation.HasKeyword(akKeyword)
            bmatching = akOtherLocation.IsChild(self)
        elseif !bmatching && self.HasKeyword(akKeyword)
            bmatching = self.IsChild(akOtherLocation)
        endif

    endif
  return bmatching
endFunction


; Sets the specified keyword's data on the location
Function SetKeywordData(Keyword akKeyword, float afData) native

; Sets this location as cleared or not
Function SetCleared(bool abCleared = true) native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
Location Function GetParent() native   :: Add a newline between files
Scriptname MagicEffect extends Form Hidden
; Get the Associated Skill for this MagicEffect
string Function GetAssociatedSkill() native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
Function SetAssociatedSkill(string skill) native

string Function GetResistance() native
Function SetResistance(string skill) native

; Hostile           0x00000001
; Recover           0x00000002
; Detrimental       0x00000004
; NoHitEvent        0x00000010
; DispelKeywords    0x00000100
; NoDuration        0x00000200
; NoMagnitude       0x00000400
; NoArea            0x00000800
; FXPersist         0x00001000
; GloryVisuals      0x00004000
; HideInUI          0x00008000
; NoRecast          0x00020000
; Magnitude         0x00200000
; Duration          0x00400000
; Painless          0x04000000
; NoHitEffect       0x08000000
; NoDeathDispel     0x10000000

bool Function IsEffectFlagSet(int flag) native
Function SetEffectFlag(int flag) native
Function ClearEffectFlag(int flag) native

float Function GetCastTime() native
Function SetCastTime(float castTime) native

int Function GetSkillLevel() native
Function SetSkillLevel(int level) native

int Function GetArea() native
Function SetArea(int area) native

float Function GetSkillUsageMult() native
Function SetSkillUsageMult(float usageMult) native

float Function GetBaseCost() native
Function SetBaseCost(float cost) native

Light Function GetLight() native
Function SetLight(Light obj) native

EffectShader Function GetHitShader() native
Function SetHitShader(EffectShader obj) native

EffectShader Function GetEnchantShader() native
Function SetEnchantShader(EffectShader obj) native
```

```

Projectile Function GetProjectile() native
Function SetProjectile(Projectile obj) native

Explosion Function GetExplosion() native
Function SetExplosion(Explosion obj) native

Art Function GetCastingArt() native
Function SetCastingArt(Art obj) native

Art Function GetHitEffectArt() native
Function SetHitEffectArt(Art obj) native

Art Function GetEnchantArt() native
Function SetEnchantArt(Art obj) native

ImpactDataSet Function GetImpactDataSet() native
Function SetImpactDataSet(ImpactDataSet obj) native

Spell Function GetEquipAbility() native
Function SetEquipAbility(Spell obj) native

ImageSpaceModifier Function GetImageSpaceMod() native
Function SetImageSpaceMod(ImageSpaceModifier obj) native

Perk Function GetPerk() native
Function SetPerk(Perk obj) native

int Function GetCastingType() native
; Constant Effect     0
; Fire And Forget     1
; Concentration       2

int Function GetDeliveryType() native
; Self                0
; Contact             1
; Aimed               2
; Target Actor        3
; Target Location     4

; Entries will be None if there is no sound
; will always return an array of size 6
Sound[] Function GetSounds() native
; Draw Sheathe        0
; Charge              1
; Ready               2
; Release             3
; Loop                4
; Hit                 5   :: Add a newline between files
Scriptname Math Hidden

; Calculates the absolute value of the passed in value - N for N, and N for (-N)
float Function abs(float afValue) global native

; Calculates the arccosine of the passed in value, returning degrees
float Function acos(float afValue) global native

; Calculates the arcsine of the passed in value, returning degrees
float Function asin(float afValue) global native

; Calculates the arctangent of the passed in value, returning degrees
float Function atan(float afValue) global native

; Calculates the ceiling of the passed in value - the smallest integer greater than or equal to the value
int Function Ceiling(float afValue) global native

; Calculates the cosine of the passed in value (in degrees)
float Function cos(float afValue) global native
```

```

; Converts degrees to radians
float Function DegreesToRadians(float afDegrees) global native

; Calculates the floor of the passed in value - the largest integer less than or equal to the value
int Function Floor(float afValue) global native

; Calculates x raised to the y power
float Function pow(float x, float y) global native

; Converts radians to degrees
float Function RadiansToDegrees(float afRadians) global native

; Calculates the sine of the passed in value (in degrees)
float Function sin(float afValue) global native

; Calculate the square root of the passed in value
float Function sqrt(float afValue) global native

; Calculates the tangent of the passed in value (in degrees)
float Function tan(float afValue) global native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
int Function LeftShift(int value, int shiftBy) global native
int Function RightShift(int value, int shiftBy) global native
int Function LogicalAnd(int arg1, int arg2) global native
int Function LogicalOr(int arg1, int arg2) global native
int Function LogicalXor(int arg1, int arg2) global native
int Function LogicalNot(int arg1) global native
float Function Log(float arg1) global native   :: Add a newline between files
Scriptname ModEvent Hidden

; ModEvent allows sending mod events with any number/type of arguments, unlike the more limited Form.SendModEvent.
;
; Example:
;
;   (Sender)
;
;       int handle = ModEvent.Create("MYPREFIX_myCustomEvent")
;       if (handle)
;           ModEvent.PushForm(handle, self)
;           ModEvent.PushForm(handle, someOtherForm)
;           ModEvent.PushInt(handle, 1000)
;           ModEvent.PushString(handle, "It worked!")
;           UIDelegate.Send(handle)
;       endIf
;
;   (Receiver)
;
;       function OnInit()
;           RegisterForModEvent("MYPREFIX_myCustomEvent", "OnMyCustomEvent")
;       endFunction
;
;       event OnMyCustomEvent(Form sender, Form theForm, int theInt, string theString)
;           ; sender == (Sender)
;           ; theForm == someOtherForm
;           ; theInt == 1000
;           ; theString == "It worked!"
;       endEvent

; Creates a new ModEvent and returns the handle.
int Function Create(string eventName) global native

; Sends the ModEvent and releases it.
; Returns true, if it was sent successfully, false if an error happened.
bool Function Send(int handle) global native

```

```
; Releases the ModEvent without sending it.
Function Release(int handle) global native

; Push single parameter.
;
; For arguments 1 .. N, the signature of the receiving event callback has to look like this:
;
;   event MyCallback(TYPE_1 PARAM_1, ... , TYPE_N PARAM_N)
;
Function PushBool(int handle, bool value) global native
Function PushInt(int handle, int value) global native
Function PushFloat(int handle, float value) global native
Function PushString(int handle, string value) global native
Function PushForm(int handle, Form value) global native
   :: Add a newline between files
Scriptname NetImmerse Hidden

; Note that only local transforms can be set as the world transform
; is computed based on the entire hierarchy rather than a single node

; Return whether the object has the particular node
bool Function HasNode(ObjectReference ref, string node, bool firstPerson) native global

; NiNode Manipulation
float Function GetNodeWorldPositionX(ObjectReference ref, string node, bool firstPerson) native global
float Function GetNodeWorldPositionY(ObjectReference ref, string node, bool firstPerson) native global
float Function GetNodeWorldPositionZ(ObjectReference ref, string node, bool firstPerson) native global

; Returns nodeB - nodeA
float Function GetRelativeNodePositionX(ObjectReference ref, string nodeA, string nodeB, bool firstPerson) native global
float Function GetRelativeNodePositionY(ObjectReference ref, string nodeA, string nodeB, bool firstPerson) native global
float Function GetRelativeNodePositionZ(ObjectReference ref, string nodeA, string nodeB, bool firstPerson) native global

float Function GetNodeLocalPositionX(ObjectReference ref, string node, bool firstPerson) native global
float Function GetNodeLocalPositionY(ObjectReference ref, string node, bool firstPerson) native global
float Function GetNodeLocalPositionZ(ObjectReference ref, string node, bool firstPerson) native global

Function SetNodeLocalPositionX(ObjectReference ref, string node, float x, bool firstPerson) native global
Function SetNodeLocalPositionY(ObjectReference ref, string node, float y, bool firstPerson) native global
Function SetNodeLocalPositionZ(ObjectReference ref, string node, float z, bool firstPerson) native global

; Sets the scale of a particular Nif node
float Function GetNodeScale(ObjectReference ref, string node, bool firstPerson) native global
Function SetNodeScale(ObjectReference ref, string node, float scale, bool firstPerson) native global

; Sets a NiTriShape's textures by name of the Nif node
Function SetNodeTextureSet(ObjectReference ref, string node, TextureSet tSet, bool firstPerson) native global


; Array based functions, return true when successful, false when unsuccessful (node did not exist, or array wrong size)

; returns the node's world position into the specify array, must be size of 3
```

```
bool Function GetNodeWorldPosition(ObjectReference ref, string node, float[] in, bool firstPerson) native global

; returns the node's relative world position of nodeB minus nodeA into the specify array, must be size of 3
bool Function GetRelativeNodePosition(ObjectReference ref, string nodeA, string nodeB, float[] in, bool firstPerson) native global

; returns the node's local position into the specify array, must be size of 3
bool Function GetNodeLocalPosition(ObjectReference ref, string node, float[] in, bool firstPerson) native global

; sets the node's local position of the specified array, must be size of 3
bool Function SetNodeLocalPosition(ObjectReference ref, string node, float[] in, bool firstPerson) native global

; Euler Rotation in DEGREES (heading, attitude, bank)
; returns the euler rotation of the node into the specified array, must be size of 3
bool Function GetNodeWorldRotationEuler(ObjectReference ref, string node, float[] in, bool firstPerson) native global

; returns the euler rotation of the node into the specified array, must be size of 3
bool Function GetNodeLocalRotationEuler(ObjectReference ref, string node, float[] in, bool firstPerson) native global

; sets the euler rotation for the node of the specified array, must be size of 3
bool Function SetNodeLocalRotationEuler(ObjectReference ref, string node, float[] in, bool firstPerson) native global

; Matrix Rotation in RADIANS
; returns the matrix rotation of the node into the specified array, must be size of 9
bool Function GetNodeWorldRotationMatrix(ObjectReference ref, string node, float[] in, bool firstPerson) native global

; returns the matrix rotation of the node into the specified array, must be size of 9
bool Function GetNodeLocalRotationMatrix(ObjectReference ref, string node, float[] in, bool firstPerson) native global

; sets the matrix rotation for the node of the specified array, must be size of 9
bool Function SetNodeLocalRotationMatrix(ObjectReference ref, string node, float[] in, bool firstPerson) native global


; DEPRECATED FUNCTIONS
Function SetNodePositionX(ObjectReference ref, string node, float x, bool firstPerson) global
    NetImmerse.SetNodeLocalPositionX(ref, node, x, firstPerson)
EndFunction
Function SetNodePositionY(ObjectReference ref, string node, float y, bool firstPerson) global
    NetImmerse.SetNodeLocalPositionY(ref, node, y, firstPerson)
EndFunction
Function SetNodePositionZ(ObjectReference ref, string node, float z, bool firstPerson) global
    NetImmerse.SetNodeLocalPositionZ(ref, node, z, firstPerson)
EndFunction

float Function GetNodePositionX(ObjectReference ref, string node, bool firstPerson) global
    return NetImmerse.GetNodeWorldPositionX(ref, node, firstPerson)
EndFunction
float Function GetNodePositionY(ObjectReference ref, string node, bool firstPerson) global
    return NetImmerse.GetNodeWorldPositionY(ref, node, firstPerson)
EndFunction
float Function GetNodePositionZ(ObjectReference ref, string node, bool firstPerson) global
    return NetImmerse.GetNodeWorldPositionZ(ref, node, firstPerson)
EndFunction   :: Add a newline between files
```

```
Scriptname ObjectReference extends Form Hidden

bool FUNCTION rampRumble(float power = 0.5, float duration = 0.25, float falloff = 1600.0)
    ; Function to shake cam/controller based on distance from player
    ; should always be called on the source of the rumble,
    ; as final intensity is relevant to player
    if power > 1.0 || power <= 0
;       debug.traceStack(self + " called rampRumble() but parameter 'power' was invalid.  Must be a non-zero float less than 1.0",1)
        ; throw the warning, but don't return false - value gets clamped anyway
    endif
    float playerDist = game.getplayer().getDistance(self)
    ; ignore if the player is too far away
    if playerDist < falloff
        float intensity = (1 - (playerDist / falloff))
        ; ramp actual intensity down based on parameter value
        intensity = intensity*power
        if intensity > 1.0
            ; clamp to prevent invalid values
;           debug.traceStack(self + " called for too much controller/camera shake.  Clamped to 1.0", 0)
            intensity = 1.0
        elseif intensity <= 0
            ; clamp to prevent invalid values
;           debug.traceStack(self + " called for too little controller/camera shake", 0)
            intensity = 0
            return false
        endif
        game.shakeCamera(game.getPlayer(), intensity)
        game.shakeController(intensity, intensity, duration)
        return true
    else
;       debug.traceStack(self + "called for rampedRumble(), but player is too far away", 0)
        return False
    endif
endFUNCTION

; Function to know if I'm near the player (whether I can be safely enabled or disabled)
bool Function IsNearPlayer()
    Actor player = Game.GetPlayer()
    Cell targetCell = self.GetParentCell()
    Cell playerCell = player.GetParentCell()

    if (targetCell != playerCell)
        ; player and target are in different cells
        if (targetCell && targetCell.IsInterior() || playerCell && playerCell.IsInterior())
            ; in different cells and at least one is an interior
            ;  -- we can safely enable or disable
            return false
        else
            ; both in an exterior -- no means of testing
            ;  worldspace at the moment, so this will do.
            if (player.GetDistance(self) > 3000)
                ; pretty darned far away -- safe
                return false
            else
                ; too close for comfort
                return true
            endif
        endif
    else
        ; in the same cell -- err on the side of caution
        return true
    endif
endFunction

```

```
;jduvall
bool Function IsInInterior()
{Returns !IsInExterior()}
  Cell parentCell = GetParentCell()
  Return parentCell && parentCell.IsInterior()
EndFunction

;kkuhlmann:
bool function MoveToIfUnloaded(ObjectReference akTarget, float afXOffset = 0.0, float afYOffset = 0.0, float afZOffset = 0.0)
{Calls MoveTo if the calling ObjectReference is currently unloaded. Doesn't do anything if it IS loaded. No waiting or while loops. Returns true if it does the moveto}
    if !Is3DLoaded()
        MoveTo(akTarget, afXOffset, afYOffset, afZOffset)
        return true
    else
        return false
    endif
endFunction

;jduvall:
function MoveToWhenUnloaded(ObjectReference akTarget, float afXOffset = 0.0, float afYOffset = 0.0, float afZOffset = 0.0)
{DEPRECATED: DO NOT USE. Calls MoveTo if both the calling ObjectReference and the akTarget ObjectReference have current locations that are not loaded.}
    while self.GetCurrentLocation().IsLoaded() || akTarget.GetCurrentLocation().IsLoaded()
        ;do nothing
;       debug.trace(self + "MoveToWhenUnloaded() waiting for current location and target location to be unloaded before moving. If called by a quest stage fragment, this may cause that quest stage to not complete until this function finishes (and if it's a startup stage, the quest will not report itself as running until the stage finishes.).", 1)
        Utility.Wait(5) ;when this function is threaded we can increase this wait time... I set it lower for testing purposes so it reevaluates faster when I need to purge cell buffers in the Civil War when calling moveto on the player between Civil War campaigns
    EndWhile
    self.MoveTo(akTarget, afXOffset, afYOffset, afZOffset)
EndFunction

;jduvall
Function DeleteWhenAble()
{This will become a native function... it will wait until the object is not persisting, then delete itself.}
    While GetParentCell() && GetParentCell().IsAttached()
        ;do nothing
;       debug.trace(self + "DeleteWhenAble() waiting for current location to be unloaded before deleting. If called by a quest stage fragment, this may cause that quest stage to not complete until this function finishes (and if it's a startup stage, the quest will not report itself as running until the stage finishes.).", 1)
        Utility.Wait(5) ;when this function is threaded we can increase this wait time... I set it lower for testing purposes so it reevaluates faster when I need to purge cell buffers in the Civil War when calling moveto on the player between Civil War campaigns
    EndWhile
    Delete()
EndFunction



;jduvall
Function AddKeyIfNeeded(ObjectReference ObjectWithNeededKey)
{Should only be called by ObjectReferences that have/are containers (ie Containers and Actors). Checks to see if self has the key to ObjectWithNeededKey, and if not, creates a copy of the key and puts it in self.}
    key NeededKey = ObjectWithNeededKey.GetKey()
    if NeededKey != None
        if GetItemCount(NeededKey) == 0
```

```
            AddItem(NeededKey)
        EndIf
    EndIf
EndFunction


; Property to obtain the current X position of the object
float Property X
  float Function get()
    return GetPositionX()
  EndFunction
EndProperty

; Property to obtain the current Y position of the object
float Property Y
  float Function get()
    return GetPositionY()
  EndFunction
EndProperty

; Property to obtain the current Z position of the object
float Property Z
  float Function get()
    return GetPositionZ()
  EndFunction
EndProperty

; Have akActivator activate this reference. If abDefaultProcessingOnly is true then any block will be bypassed
; and no OnActivate event will be sent. The function returns true if default processing ran, and succeeded. If
; default processing has been blocked, will always return false.
bool Function Activate(ObjectReference akActivator, bool abDefaultProcessingOnly = false) native

; Sets up a dependent animated object
; This function should be used only with a coder supervision.  It is left undocumented because it can cause dangling pointers as well as very broken functionality
; for the dependent object if used improperly.
bool Function AddDependentAnimatedObjectReference( ObjectReference akDependent ) native

; Add an inventory event filter to this reference. Item added/removed events matching the
; specified form (or in the specified form list) will now be let through.
Function AddInventoryEventFilter(Form akFilter) native

; Adds the specified base object or object reference to this object reference's container/inventory
; Note that you cannot add more then one copy of a reference to a container (a warning will be printed if you try)
Function AddItem(Form akItemToAdd, int aiCount = 1, bool abSilent = false) native

; Adds this reference (which is a map marker) to the map, optionally making it available for fast travel
Function AddToMap(bool abAllowFastTravel = false) native

; Apply an impulse to this reference
Function ApplyHavokImpulse(float afX, float afY, float afZ, float afMagnitude) native

; Turns on and off blocking of normal activation - OnActivate events will still be sent
Function BlockActivation(bool abBlocked = true) native

; Calculate's this references encounter level based on the requested difficulty level
; 0 - Easy
; 1 - Medium
; 2 - Hard
; 3 - Very Hard
; 4 - None
int Function CalculateEncounterLevel(int aiDifficulty = 4) native

```

```
; Can the map marker be fast traveled to?
bool Function CanFastTravelToMarker() native

; Clears all effects of destruction from this object
Function ClearDestruction() native

; Create a detection event at this reference, with the specified owner. Sound level is between 0 and 100

Function CreateDetectionEvent(Actor akOwner, int aiSoundLevel = 0 ) native

; Damages this object and advances the destruction stage - does not return until the object is damaged
Function DamageObject(float afDamage) native

; Delets this object
Function Delete() native

; Disables this object - fading out if requested
Function Disable(bool abFadeOut = false) native

; Disables this object - fading out if requested. Does NOT wait for the fade or disable to finish
Function DisableNoWait(bool abFadeOut = false) native

; Drops the specified object from this object's inventory
ObjectReference Function DropObject(Form akObject, int aiCount = 1) native

; Enables this object - fading in if requested
Function Enable(bool abFadeIn = false) native

; Enables the ability to fast travel to this marker - or disables it. Note that if you disable
; fast travel the player will see "You haven't discovered this location" as an error message
Function EnableFastTravel(bool abEnable = true) native

; Enables this object - fading in if requested. Does NOT wait for the fade or enable to finish
Function EnableNoWait(bool abFadeIn = false) native

; Forcibly adds / removes the ragdoll for a reference to the world
Function ForceAddRagdollToWorld() native
Function ForceRemoveRagdollFromWorld() native

; Gets the actor that owns this object (or None if not owned by an Actor)
ActorBase Function GetActorOwner() native

; Get the current X angle of this object
float Function GetAngleX() native

; Get the current Y angle of this object
float Function GetAngleY() native

; Get the current Z angle of this object
float Function GetAngleZ() native

; Get a variable from the reference's animation graph (if applicable). Bool version.
bool Function GetAnimationVariableBool(string arVariableName) native

; Get a variable from the reference's animation graph (if applicable). Int version.
int Function GetAnimationVariableInt(string arVariableName) native

; Get a variable from the reference's animation graph (if applicable). Float version.
float Function GetAnimationVariableFloat(string arVariableName) native

; Returns the base object this reference represents
Form Function GetBaseObject() native

```

```
; Returns the object's current destruction stage
int Function GetCurrentDestructionStage() native

; Returns this reference's current location
Location Function GetCurrentLocation() native

; Returns the scene this reference is currently in - if any
Scene Function GetCurrentScene() native

; Calculates the distance between this reference and another - both must either be in the same interior, or same worldspace
float Function GetDistance(ObjectReference akOther) native

; Returns this reference's editor location
Location Function GetEditorLocation() native

; Gets the faction that owns this object (or None if not owned by a Faction)
Faction Function GetFactionOwner() native

; Gets the angle between this object's heading and the other object in degrees - in the range from -180 to 180
float Function GetHeadingAngle(ObjectReference akOther) native

; Get the current height of the object
float Function GetHeight() native

; Returns how many of the specified item is in this object reference's inventory
int Function GetItemCount(Form akItem) native

; Returns the smithed health of this object reference (1.0 == 100%)
float Function GetItemHealthPercent() native

; Returns the key base object that will unlock this object
Key Function GetKey() native

; Get the current length of the object
float Function GetLength() native

; Get our linked reference
ObjectReference Function GetLinkedRef(Keyword apKeyword = NONE) native

; Get the level of the lock on this object
int Function GetLockLevel() native

;jtucker, jduvall
;This function  counts the number of linked refs that are in a linked Ref chain (ie object is linked to A, A is linked to B, etc. this then counts all the linked refs.)
;Often used in conjunction with GetNthLinkedRef()
;*** WARNING: Having a link ref chain that at any point loops back on itself and calling this function will result in very bad things. Don't do that!***
int Function countLinkedRefChain(keyword apKeyword = None, int maxExpectedLinkedRefs = 100)
    ;Don't use this on a loop of linked refs.
    ObjectReference CurrentLink = self
    ObjectReference NewLink
    int NumLinkedRefs = 0

    while(currentLink) && NumLinkedRefs <= maxExpectedLinkedRefs

        NewLink = currentLink.getLinkedRef(apKeyword)

        if NewLink != self
            currentLink = NewLink
            NumLinkedRefs = NumLinkedRefs + 1
        Else
            currentLink = None
;           debug.trace( self + "countLinkedRefs() found itself. This suggests it was linked back to itself. This will create an infinite loop, so we are killing the function now. NumLinkedRefs =" + NumLinkedRefs)
```

```
        EndIf

    endWhile

    if NumLinkedRefs >= maxExpectedLinkedRefs
;       debug.trace( self + "countLinkedRefs() bailing out early because it found more linked refs than maxExpectedLinkRefs (suggesting an infinite loop). LinkedRefs found:" + NumLinkedRefs + ", maxExpectedLinkedRefs:" + maxExpectedLinkedRefs)
    EndIf


    return NumLinkedRefs

endFunction


; Returns the Nth linked ref from this reference (0 = self, 1 = GetLinkedRef, 2 = GetLinkedRef.GetLinkedRef, etc)
ObjectReference Function GetNthLinkedRef(int aiLinkedRef) native


; Enables all of the references that are linked, in a chain, to this one.
Function EnableLinkChain(Keyword apKeyword = None)
    ObjectReference CurrentLink = GetLinkedRef(apKeyword)
    While CurrentLink
        CurrentLink.Enable()
        CurrentLink = CurrentLink.GetLinkedRef(apKeyword)
    endWhile
endFunction


; Disables all of the references that are linked, in a chain, to this one.
Function DisableLinkChain(Keyword apKeyword = None, bool abFadeOut = false)
    ObjectReference CurrentLink = GetLinkedRef(apKeyword)
    While CurrentLink
        CurrentLink.Disable(abFadeOut)
        CurrentLink = CurrentLink.GetLinkedRef(apKeyword)
    endWhile
endFunction


; Get this object's mass
float Function GetMass() native

; Gets the open state of this object. Which can be one of the following:
; 0 - None
; 1 - Open
; 2 - Opening
; 3 - Closed
; 4 - Closing
int Function GetOpenState() native

; Gets the cell this object is in
Cell Function GetParentCell() native

; Get the current X position of the object
float Function GetPositionX() native

; Get the current Y position of the object
float Function GetPositionY() native

; Get the current Z position of the object
float Function GetPositionZ() native

; Get the current scale of the object
float Function GetScale() native

; Get the number of objects inside this trigger (throws warning if not a triggger)
int Function GetTriggerObjectCount() native
```

```

; Gets the voice type for this reference. Will return None if not an actor or a talking activator
VoiceType Function GetVoiceType() native

; Get the current width of the object
float Function GetWidth() native

; Get this objects worldspace
WorldSpace Function GetWorldSpace() native

; Returns self cast as an actor
actor Function GetSelfAsActor()
    return self as Actor
endFunction

; Returns if this reference has an active effect coming from a magic effect with the specified keyword attached
bool Function HasEffectKeyword(Keyword akKeyword) native

; Returns whether the reference has the given node
bool Function HasNode(string asNodeName) native

; Returns if this reference has the specified location ref type
bool Function HasRefType(LocationRefType akRefType) native

; Flags this reference as ignoring (or not ignoring) friendly hits
Function IgnoreFriendlyHits(bool abIgnore = true) native

; Interrupts any spell-casting this object may be doing
Function InterruptCast() native

; Checks to see if the passed in reference is the activate child of this one
bool Function IsActivateChild(ObjectReference akChild) native

; Checks to see if activation is currently blocked on this object
bool Function IsActivationBlocked() native

; Returns if the 3d for this object is loaded or not
bool Function Is3DLoaded() native

; Is this object currently flagged for delete?
bool Function IsDeleted() native

; Is this object currently disabled?
bool Function IsDisabled() native

; Because Shane got tired of remembering which way to call this
bool Function IsEnabled()
    return !IsDisabled()
EndFunction

; Is any marker on this furniture in use?
bool Function IsFurnitureInUse(bool abIgnoreReserved = false) native

; Is a particular marker on this furniture in use?
bool Function IsFurnitureMarkerInUse(int aiMarker, bool abIgnoreReserved = false) native

; Is this object ignoring friendly hits?
bool Function IsIgnoringFriendlyHits() native

; Is this actor or talking activator currently talking to the player?
bool Function IsInDialogueWithPlayer() native

; Is the lock on this object broken?
bool Function IsLockBroken() native

; Is the lock on this object locked?
bool Function IsLocked() native
```

```

; Is the map marker visible?
bool Function IsMapMarkerVisible() native

; Executes a knock effect to an area
Function KnockAreaEffect(float afMagnitude, float afRadius) native

; Lock/unlock this object. If told to lock it, it will add a lock if it doesn't have one. If locked/unlocked as the owner on a door,
; the adjoining cell will be made public/private as appropriate
Function Lock(bool abLock = true, bool abAsOwner = false) native

; Moves this object to the position of the specified object, with an offset, and optionally matching its rotation
Function MoveTo(ObjectReference akTarget, float afXOffset = 0.0, float afYOffset = 0.0, float afZOffset = 0.0, bool abMatchRotation = true) native

; Moves this object to the position (and rotation) of the specified object's interaction position
Function MoveToInteractionLocation(ObjectReference akTarget) native

; Moves this object to its editor location
Function MoveToMyEditorLocation() native

; Moves this object to the position (and rotation) of the specified node on the specified object's 3D
Function MoveToNode(ObjectReference akTarget, string asNodeName) native

; Create x copies of the passed in form (forcing them to persist if desired) and place them at our location, returning the last object created
ObjectReference Function PlaceAtMe(Form akFormToPlace, int aiCount = 1, bool abForcePersist = false, bool abInitiallyDisabled = false) native

; Create an actor at this object's location. Level mod is one of the following:
; 0 - Easy
; 1 - Medium
; 2 - Hard
; 3 - Boss
; 4 - None
Actor Function PlaceActorAtMe(ActorBase akActorToPlace, int aiLevelMod = 4, EncounterZone akZone = None) native

; Start the specified animation playing - returns true if it succeeds
bool Function PlayAnimation(string asAnimation) native

; Start the specified animation playing and wait for the specified event - returns true if succeeds
bool Function PlayAnimationAndWait(string asAnimation, string asEventName) native

; Start the specified Gamebryo animation playing - returns true if it succeeds
bool Function PlayGamebryoAnimation(string asAnimation, bool abStartOver = false, float afEaseInTime = 0.0) native

; Play the specified impact effect - returns true if it succeeds
bool Function PlayImpactEffect(ImpactDataSet akImpactEffect, string asNodeName = "", float afPickDirX = 0.0, float afPickDirY = 0.0, float afPickDirZ = -1.0, float afPickLength = 512.0, bool abApplyNodeRotation = false, bool abUseNodeLocalRotation = false) native

; Play two animations at once - one on this object, one on another object
bool Function PlaySyncedAnimationSS(string asAnimation1, ObjectReference akObj2, string asAnimation2) native

; Play two animations at once - one on this object, one on another object - and wait for both
bool Function PlaySyncedAnimationAndWaitSS(string asAnimation1, string asEvent1, ObjectReference akObj2, string asAnimation2, string asEvent2) native

; Play a terrain effect that is attached to the specified bone of this object.
```

```
Function PlayTerrainEffect(string asEffectModelName, string asAttachBoneName) native

; Tells this object to process a trap hitting it
Function ProcessTrapHit(ObjectReference akTrap, float afDamage, float afPushback, float afXVel, float afYVel, float afZVel, float afXPos, float afYPos, float afZPos, int aeMaterial, float afStagger) native

; Pushes the passed-in actor away from this object, using the passed in knockback force to determine the speed
Function PushActorAway(Actor akActorToPush, float aiKnockbackForce) native

; Remove all inventory event filters from this reference - all item added/removed events will now be received
Function RemoveAllInventoryEventFilters() native

; Removes all items from this container, transferring it to the other object if passed
Function RemoveAllItems(ObjectReference akTransferTo = None, bool abKeepOwnership = false, bool abRemoveQuestItems = false) native

; Remove an inventory event filter from this reference. Item added/removed events matching the
; specified form (or in the specified form list) will no longer be let through.
Function RemoveInventoryEventFilter(Form akFilter) native

; Removes the specified item from this object reference's inventory
Function RemoveItem(Form akItemToRemove, int aiCount = 1, bool abSilent = false, ObjectReference akOtherContainer = None) native

; Removes a previously added dependent object
; This function should be used only with a coder supervision.  It is left undocumented because it can cause dangling pointers as well as very broken functionality
; for the dependent object if used improperly.
bool Function RemoveDependentAnimatedObjectReference( ObjectReference akDependent ) native

; Resets this object, optional place the object at the new target
Function Reset(ObjectReference akTarget = None) native

; Has this object "say" the specified topic, as if spoken by the specified actor (if one is
; provided, and potentially "speaking" in the player's head.
Function Say(Topic akTopicToSay, Actor akActorToSpeakAs = None, bool abSpeakInPlayersHead = false) native

; Has this object behave as if the specified actor attempted to steal it
Function SendStealAlarm(Actor akThief) native

; Sets this object's actor cause to the specified actor
Function SetActorCause(Actor akActor) native

; Sets this object's owner to the specified actor base - None means to remove ownership
Function SetActorOwner(ActorBase akActorBase) native

; Set the orientation of the object (angles are in degrees)
Function SetAngle(float afXAngle, float afYAngle, float afZAngle) native

; Set a variable on the reference's animation graph (if applicable). Bool version.
Function SetAnimationVariableBool(string arVariableName, bool abNewValue) native

; Set a variable on the reference's animation graph (if applicable). Int version.
Function SetAnimationVariableInt(string arVariableName, int aiNewValue) native

; Set a variable on the reference's animation graph (if applicable). Float version.
Function SetAnimationVariableFloat(string arVariableName, float afNewValue) native

; Sets this object as destroyed or not
Function SetDestroyed(bool abDestroyed = true) native

; Sets this object's owner to the specified faction
```

```
Function SetFactionOwner(Faction akFaction) native

; Sets the lock level on this object. Will add an unlocked lock to it if it doesn't have one
Function SetLockLevel(int aiLockLevel) native

; Sets the motion type of the reference
; aeMotionType: The type of motion (see properties at end of file)
; abAllowActivate: When setting to a dynamic type, allows the simulation to be activated
Function SetMotionType(int aeMotionType, bool abAllowActivate = true) native

; Sets this object reference as one that teammates will refuse to do favors on
Function SetNoFavorAllowed(bool abNoFavor = true) native

; Opens/closes this object
Function SetOpen(bool abOpen = true) native

; Set the position of the object
Function SetPosition(float afX, float afY, float afZ) native

; Set the current scale of the object
Function SetScale(float afScale) native

; Makes the reference translate to the given position/orientation
; Note: Rotation speed is entirely dependent on the length of the path and the movement speed
; that is, the rotation will happen such that the reference reaches the goal orientation at the end
; of the translation.
Function TranslateTo(float afX, float afY, float afZ, float afXAngle, float afYAngle, float afZAngle, float afSpeed, float afMaxRotationSpeed = 0.0) native

; Makes the reference translate to the given position/orientation on a spline
Function SplineTranslateTo(float afX, float afY, float afZ, float afXAngle, float afYAngle, float afZAngle, float afTangentMagnitude, float afSpeed, float afMaxRotationSpeed = 0.0) native

; Makes the reference translate to the target node's ref/orient on a spline at the given speed
Function SplineTranslateToRefNode(ObjectReference arTarget, string arNodeName, float afTangentMagnitude, float afSpeed, float afMaxRotationSpeed = 0.0) native

; Stops the reference from moving
Function StopTranslation() native

; Makes the reference translate to the target ref position/orient at the given speed
Function TranslateToRef(ObjectReference arTarget, float afSpeed, float afMaxRotationSpeed = 0.0)
    TranslateTo(arTarget.X, arTarget.Y, arTarget.Z, arTarget.GetAngleX(), arTarget.GetAngleY(), arTarget.GetAngleZ(), afSpeed, afMaxRotationSpeed)
endFunction

; Makes the reference translate to the target ref position/orient on a spline at the given speed
Function SplineTranslateToRef(ObjectReference arTarget, float afTangentMagnitude, float afSpeed, float afMaxRotationSpeed = 0.0)
    SplineTranslateTo(arTarget.X, arTarget.Y, arTarget.Z, arTarget.GetAngleX(), arTarget.GetAngleY(), arTarget.GetAngleZ(), afTangentMagnitude, afSpeed, afMaxRotationSpeed)
endFunction

; Tether a prisoner cart to the given horse.
Function TetherToHorse(ObjectReference akHorse) native

; Waits for the animation graph to send the specified event
bool Function WaitForAnimationEvent(string asEventName) native

; Convenience function to check if I'm in a location or any of its children
bool Function IsInLocation(Location akLocation)
```

```
    ; cache current location to avoid changing location while this function is running
    (surprisingly that seems to be happening occasionally)
    Location currLoc = GetCurrentLocation()
    if currLoc == None
        return false
    else
        return akLocation.IsChild(currLoc) || currLoc == akLocation
    endif
endFunction

; Event received when this reference is activated
Event OnActivate(ObjectReference akActionRef)
EndEvent

; Event received when this object has moved to an attached cell from a detached one
Event OnAttachedToCell()
EndEvent

; Event received when this object's parent cell is attached
Event OnCellAttach()
EndEvent

; Event received when this object's parent cell is detached
Event OnCellDetach()
EndEvent

; Event received when every object in this object's parent cell is loaded (TODO: Find
restrictions)
Event OnCellLoad()
EndEvent

; Event received when this object is closed
Event OnClose(ObjectReference akActionRef)
EndEvent

; Event received when this object enters, exits, or changes containers
Event OnContainerChanged(ObjectReference akNewContainer, ObjectReference akOldContainer)
EndEvent

; Event received when this reference's destruction stage has changed
Event OnDestructionStageChanged(int aiOldStage, int aiCurrentStage)
EndEvent

; Event recieved when this object moves to a detached cell from an attached one
Event OnDetachedFromCell()
EndEvent

; Event received when this object is equipped by an actor
Event OnEquipped(Actor akActor)
EndEvent

; Event received when this object is grabbed by the player
Event OnGrab()
EndEvent

; Event received when a this trigger is tripped
Event OnTrigger(ObjectReference akActionRef)
EndEvent

; Event received when this trigger volume is entered
Event OnTriggerEnter(ObjectReference akActionRef)
EndEvent

; Event received when this trigger volume is left
Event OnTriggerLeave(ObjectReference akActionRef)
EndEvent

; Event received when this object is hit by a source (weapon, spell, explosion) or
projectile attack
```

```
Event OnHit(ObjectReference akAggressor, Form akSource, Projectile akProjectile, bool abPowerAttack, bool abSneakAttack, bool abBashAttack, bool abHitBlocked)
EndEvent

; Event received when an item is added to this object's inventory. If the item is a persistant reference, akItemReference will
; point at it - otherwise the parameter will be None
Event OnItemAdded(Form akBaseItem, int aiItemCount, ObjectReference akItemReference, ObjectReference akSourceContainer)
EndEvent

; Event received when an item is removed from this object's inventory. If the item is a persistant reference, akItemReference
; will point at it - otherwise the parameter will be None
Event OnItemRemoved(Form akBaseItem, int aiItemCount, ObjectReference akItemReference, ObjectReference akDestContainer)
EndEvent

; Event recieved when this object is completely loaded - will be fired every time this object is loaded
Event OnLoad()
EndEvent

; Event received when the lock on this object changes
Event OnLockStateChanged()
EndEvent

; Event received when a magic affect is being applied to this object
Event OnMagicEffectApply(ObjectReference akCaster, MagicEffect akEffect)
EndEvent

; Event received when this object is opened
Event OnOpen(ObjectReference akActionRef)
EndEvent

; Event received when this object, if a book, is read
Event OnRead()
EndEvent

; Event received when this object is released by the player
Event OnRelease()
EndEvent

; Event received when this reference is reset
Event OnReset()
EndEvent

; Event received when this reference is sold by an actor
Event OnSell(Actor akSeller)
EndEvent

; Event received when a spell is cast by this object
Event OnSpellCast(Form akSpell)
EndEvent

; Event received when translation is almost complete (from a call to TranslateTo, "almost" is determined by a gamesetting, default is 90% of the way)
Event OnTranslationAlmostComplete()
EndEvent

; Event received when translation is complete (from a call to TranslateTo)
Event OnTranslationComplete()
EndEvent

; Event received when translation is aborted (from a call to StopTranslateTo)
Event OnTranslationFailed()
EndEvent

; Event recieved when this reference hits a target
```

```
Event OnTrapHit(ObjectReference akTarget, float afXVel, float afYVel, float afZVel, float afXPos, float afYPos, float afZPos, \
    int aeMaterial, bool abInitialHit, int aeMotionType)
EndEvent

; Event recieved when this starts hitting a target
Event OnTrapHitStart(ObjectReference akTarget, float afXVel, float afYVel, float afZVel, float afXPos, float afYPos, float afZPos, \
    int aeMaterial, bool abInitialHit, int aeMotionType)
EndEvent

; Event recieved when this stops hitting a target
Event OnTrapHitStop(ObjectReference akTarget)
EndEvent

; Event received when this object is unequipped by an actor
Event OnUnequipped(Actor akActor)
EndEvent

; Event recieved when this object is being unloaded - will be fired every time this object is unloaded
Event OnUnload()
EndEvent

; Event received when this object's Ward is hit by a spell
Event OnWardHit(ObjectReference akCaster, Spell akSpell, int aiStatus)
EndEvent

; Set of read-only properties to essentually make a fake enum for motion types passed in to the trap hit
int Property Motion_Dynamic = 1 AutoReadOnly
int Property Motion_SphereIntertia = 2 AutoReadOnly
int Property Motion_BoxIntertia = 3 AutoReadOnly
int Property Motion_Keyframed = 4 AutoReadOnly
int Property Motion_Fixed = 5 AutoReadOnly
int Property Motion_ThinBoxIntertia = 6 AutoReadOnly
int Property Motion_Character = 7 AutoReadOnly

; added in 1.6.1126
Bool Function IsContainerEmpty() Native
Function RemoveAllStolenItems(ObjectReference akTransferTo) Native
Function SetContainerAllowStolenItems(Bool setAllowStolenItems) Native
Int Function GetAllItemsCount() Native


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

; Container-only functions
int Function GetNumItems() native
Form Function GetNthForm(int index) native
float Function GetTotalItemWeight() native
float Function GetTotalArmorWeight() native

; Tree and Flora only functions
bool Function IsHarvested() native
Function SetHarvested(bool harvested) native

; Tempering
Function SetItemHealthPercent(float health) native

; Charges

; Only works on ObjectReferences that have user-enchants
Function SetItemMaxCharge(float maxCharge) native
; Works on any enchanted item
float Function GetItemMaxCharge() native

float Function GetItemCharge() native
Function SetItemCharge(float charge) native

Function ResetInventory() native

bool Function IsOffLimits() native

; Returns the name of this reference
; this is the name that is displayed
string Function GetDisplayName() native

; Sets a reference's display name
; returns false if force is false and the reference
; is held by an alias using 'Stored Text' or 'Uses Stored Text'
; Text Replacement does not use this name and may be lost if forced
bool Function SetDisplayName(string name, bool force = false) native

; Returns the enable parent object
ObjectReference Function GetEnableParent() native

; Returns the player-made enchantment if there is one
Enchantment Function GetEnchantment() native

; Changes an item's player-made enchantment to something else
; None enchantment will remove the existing enchantment
; does not delete the custom enchantment, only removes it
Function SetEnchantment(Enchantment source, float maxCharge) native

; Creates a new enchantment on the item given the specified parameters
; all arrays must be the same size
; created enchantments are not purged from the save when removed or overwritten
; exact same enchantments are re-used by the game
Function CreateEnchantment(float maxCharge, MagicEffect[] effects, float[] magnitudes,
int[] areas, int[] durations) native

; Returns the number of ref aliases holding this reference
int Function GetNumReferenceAliases() native

; Returns the nth ReferenceAlias holding this reference
ReferenceAlias Function GetNthReferenceAlias(int n) native

; Returns the poison applied to the weapon
Potion Function GetPoison() native

; Returns all base forms in the inventory/container into the specified FormList
Function GetAllForms(FormList toFill) native

; Returns all base forms from the container into a new array
Form[] Function GetContainerForms() native

; Returns all of the aliases holding this reference
ReferenceAlias[] Function GetReferenceAliases() native   :: Add a newline between files
Scriptname Outfit extends Form Hidden


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

int Function GetNumParts() native
Form Function GetNthPart(int n) native   :: Add a newline between files
Scriptname Perk extends Form Hidden


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
Perk Function GetNextPerk() native

int Function GetNumEntries() native

int Function GetNthEntryRank(int n) native
bool Function SetNthEntryRank(int n, int rank) native

int Function GetNthEntryPriority(int n) native
```

```
bool Function SetNthEntryPriority(int n, int priority) native

Quest Function GetNthEntryQuest(int n) native
bool Function SetNthEntryQuest(int n, Quest newQuest) native

int Function GetNthEntryStage(int n) native
bool Function SetNthEntryStage(int n, int stage) native

Spell Function GetNthEntrySpell(int n) native
bool Function SetNthEntrySpell(int n, Spell newSpell) native

LeveledItem Function GetNthEntryLeveledList(int n) native
bool Function SetNthEntryLeveledList(int n, LeveledItem lList) native

string Function GetNthEntryText(int n) native
bool Function SetNthEntryText(int n, string newText) native

float Function GetNthEntryValue(int n, int i) native
bool Function SetNthEntryValue(int n, int i, float value) native   :: Add a newline
between files
Scriptname Potion extends Form

; Is this postion classified as hostile?
bool Function IsHostile() native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; Is this potion classified as Food?
bool Function IsFood() native

; Is this potion classified as Poison?
bool Function IsPoison() native

; return the number of the effects
int Function GetNumEffects() native

; return the magnitude of the specified effect
float Function GetNthEffectMagnitude(int index) native

; return the area of the specified effect
int Function GetNthEffectArea(int index) native

; return the duration of the specified effect
int Function GetNthEffectDuration(int index) native

; return the magic effect of the specified effect
MagicEffect Function GetNthEffectMagicEffect(int index) native

; return the index of the costliest effect
int Function GetCostliestEffectIndex() native

; sets the magnitude of the specified effect
Function SetNthEffectMagnitude(int index, float value) native

; sets the area of the specified effect
Function SetNthEffectArea(int index, int value) native

; sets the duration of the specified effect
Function SetNthEffectDuration(int index, int value) native

; gets the use sound of this potion
SoundDescriptor Function GetUseSound() native

; Returns all the magnitudes of this object in order
float[] Function GetEffectMagnitudes() native

; Returns all the areas of this object in order
int[] Function GetEffectAreas() native

; Returns all the durations of this object in order
```

```
int[] Function GetEffectDurations() native

; Returns all the magic effects of this object in order
MagicEffect[] Function GetMagicEffects() native   :: Add a newline between files
Scriptname Quest extends Form Hidden

; non-native functions

; thread-safe way to modify a global value
; optional parameters:
; aiObjectiveID = objective ID to redisplay
; afTargetValue = value you're counting up (or down) towards -- if included, function
will return TRUE when the global reaches the target value
; abCountingUp = by default, function assumes you're counting up towards the target
value; make this false to count DOWN towards target value
; abCompleteObjective = by default, function assumes you're completing the objective
once you reach the target value; make this false to FAIL the objective
; abRedisplayObjective = by default, function asssume you want to redisplay the
objective every time the global is incremeneted; make this FALSE to only display the
objectives on complete or failure
bool Function ModObjectiveGlobal(float afModValue, GlobalVariable aModGlobal, int
aiObjectiveID = -1, float afTargetValue = -1.0, bool abCountingUp = true, bool
abCompleteObjective = true, bool abRedisplayObjective = true)
    aModGlobal.Mod(afModValue)
    UpdateCurrentInstanceGlobal(aModGlobal)
    if aiObjectiveID >= 0
        ; display/complete objectives automatically
        if afTargetValue > -1
            if (abCountingUp && aModGlobal.value >= afTargetValue) || (!abCountingUp &&
            aModGlobal.value <= afTargetValue)
                if (abCompleteObjective)
                    ; complete objective
                    SetObjectiveCompleted(aiObjectiveID)
                    return true
                Else
                    ; fail objective
                    SetObjectiveFailed(aiObjectiveID)
                    return true
                Endif
            elseIf (abRedisplayObjective)
                ; redisplay objective
                SetObjectiveDisplayed(aiObjectiveID, true, true)
            Else
                SetObjectiveDisplayed(aiObjectiveID, true, false)
            endif
        elseIf (abRedisplayObjective)
            ; no target value, always redisplay objective
            SetObjectiveDisplayed(aiObjectiveID, true, true)
        Else
            SetObjectiveDisplayed(aiObjectiveID, true, false)
        endif
    endif
    return false
endFunction


; native functions

; Flags all objectives as complete
Function CompleteAllObjectives() native

; Flags this quest as completed
Function CompleteQuest() native

; Flags all objectives as failed
Function FailAllObjectives() native

; Obtains the specified alias on the quest
Alias Function GetAlias(int aiAliasID) native
```

```

; Obtains the id of the highest completed stage on this quest
int Function GetCurrentStageID() native

; Alias for GetCurrentStage - obtains the highest completed stage on this quest
int Function GetStage()
  return GetCurrentStageID()
EndFunction

; Alias for IsStageDone - checks to see whether the given stage is done or not
bool Function GetStageDone(int aiStage)
  return IsStageDone(aiStage)
EndFunction

; Is this quest "active" (tracked by the player)?
bool Function IsActive() native

; Checks to see if the quest is completed
bool Function IsCompleted() native

; Checks to see if the specified objective is completed
bool Function IsObjectiveCompleted(int aiObjective) native

; Checks to see if the specified objective is displayed
bool Function IsObjectiveDisplayed(int aiObjective) native

; Checks to see if the specified objective is failed
bool Function IsObjectiveFailed(int aiObjective) native

; Checks to see if the quest is running
bool Function IsRunning() native

; Obtains whether the specified stage is done or not
bool Function IsStageDone(int aiStage) native

; Checks to see if the quest is enabled but not running yet
bool Function IsStarting() native

; Checks to see if the quest is not enabled anymore but still shutting down
bool Function IsStopping() native

; Checks to see if the quest is no longer enabled or running
bool Function IsStopped() native

; Resets the quest
Function Reset() native

; Flags this quest as "active" (tracked by the player)
Function SetActive(bool abActive = true) native

; Set the quest to the requested stage ID - returns true if stage exists and was set.
; This function is latent and will wait for the quest to start up before returning (if
it needed to be started)
bool Function SetCurrentStageID(int aiStageID) native

; Sets the specified objective to completed or not
Function SetObjectiveCompleted(int aiObjective, bool abCompleted = true) native

; Sets the specified objective to displayed or hidden - if abForce is true, will display
the objective even if it has already been displayed
Function SetObjectiveDisplayed(int aiObjective, bool abDisplayed = true, bool abForce =
false) native

; Sets the specified objective to failed or not
Function SetObjectiveFailed(int aiObjective, bool abFailed = true) native

; Alias of SetCurrentStage - Set the quest to the requested stage
; This function is latent and will wait for the quest to start up before returning (if
it needed to be started)
```

```
bool Function SetStage(int aiStage)
  return SetCurrentStageID(aiStage)
EndFunction

; Starts the quest - returns whether the quest was able to be started or not
; This function is latent and will wait for the quest to start up before returning
bool Function Start() native

; Stops the quest
Function Stop() native

; Updates current instance's value for the given global
bool Function UpdateCurrentInstanceGlobal( GlobalVariable aUpdateGlobal ) native

; Story manager events - fired in parallel with the quest startup stage

Event OnStoryAddToPlayer(ObjectReference akOwner, ObjectReference akContainer, \
    Location akLocation, Form akItemBase, int aiAcquireType)
EndEvent

Event OnStoryArrest(ObjectReference akArrestingGuard, ObjectReference akCriminal, \
    Location akLocation, int aiCrime)
EndEvent

Event OnStoryAssaultActor(ObjectReference akVictim, ObjectReference akAttacker, \
    Location akLocation, int aiCrime)
EndEvent

Event OnStoryBribeNPC(ObjectReference akActor)
EndEvent

Event OnStoryCastMagic(ObjectReference akCastingActor, ObjectReference akSpellTarget, \
    Location akLocation, Form akSpell)
EndEvent

Event OnStoryChangeLocation(ObjectReference akActor, Location akOldLocation, \
    Location akNewLocation)
EndEvent

Event OnStoryCrimeGold(ObjectReference akVictim, ObjectReference akCriminal, \
    Form akFaction, int aiGoldAmount, int aiCrime)
EndEvent

Event OnStoryCure(Form akInfection)
EndEvent

Event OnStoryDialogue(Location akLocation, ObjectReference akActor1, ObjectReference akActor2)
EndEvent

Event OnStoryDiscoverDeadBody(ObjectReference akActor, ObjectReference akDeadActor, \
    Location akLocation)
EndEvent

Event OnStoryEscapeJail(Location akLocation, Form akCrimeGroup)
EndEvent

Event OnStoryActivateActor(Location akLocation, ObjectReference akActor)
EndEvent

Event OnStoryFlatterNPC(ObjectReference akActor)
EndEvent

Event OnStoryHello(Location akLocation, ObjectReference akActor1, ObjectReference akActor2)
EndEvent

Event OnStoryIncreaseLevel(int aiNewLevel)
EndEvent
```

```

Event OnStoryIncreaseSkill(string asSkill)
EndEvent

Event OnStoryInfection(ObjectReference akTransmittingActor, Form akInfection)
EndEvent

Event OnStoryIntimidateNPC(ObjectReference akActor)
EndEvent

Event OnStoryJail(ObjectReference akGuard, Form akCrimeGroup, Location akLocation, \
    int aiCrimeGold)
EndEvent

Event OnStoryKillActor(ObjectReference akVictim, ObjectReference akKiller, \
    Location akLocation, int aiCrimeStatus, int aiRelationshipRank)
EndEvent

Event OnStoryCraftItem(ObjectReference akBench, Location akLocation, Form akCreatedItem)
EndEvent

Event OnStoryNewVoicePower(ObjectReference akActor, Form akVoicePower)
EndEvent

Event OnStoryPickLock(ObjectReference akActor, ObjectReference akLock)
EndEvent

Event OnStoryPayFine(ObjectReference akCriminal, ObjectReference akGuard, \
    Form akCrimeGroup, int aiCrimeGold)
EndEvent

Event OnStoryPlayerGetsFavor(ObjectReference akActor)
EndEvent

Event OnStoryRelationshipChange(ObjectReference akActor1, ObjectReference akActor2, \
    int aiOldRelationship, int aiNewRelationship)
EndEvent

Event OnStoryRemoveFromPlayer(ObjectReference akOwner, ObjectReference akItem, \
    Location akLocation, Form akItemBase, int aiRemoveType)
EndEvent

Event OnStoryScript(Keyword akKeyword, Location akLocation, ObjectReference akRef1, \
    ObjectReference akRef2, int aiValue1, int aiValue2)
EndEvent

Event OnStoryServedTime(Location akLocation, Form akCrimeGroup, int aiCrimeGold, \
    int aiDaysJail)
EndEvent

Event OnStoryTrespass(ObjectReference akVictim, ObjectReference akTrespasser, \
    Location akLocation, int aiCrime)
EndEvent

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

; returns the quest with the specified editor id
Quest Function GetQuest(string editorId) global native

; returns the editor ID of the quest
string Function GetID() native

; returns the priority of the quest
int Function GetPriority() native

; returns the number of aliases associated with the quest
int Function GetNumAliases() native

; returns the specified alias associated with the queest
```

```
Alias Function GetNthAlias(int index) native

; returns the alias associated with the quest by name
Alias Function GetAliasByName(string name) native

; returns the alias by AlisID
Alias Function GetAliasById(int aliasId) native

; Returns all the aliases of this quest
Alias[] Function GetAliases() native   :: Add a newline between files
Scriptname Race extends Form Hidden

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; returns the number of spells for the race
int Function GetSpellCount() native

; returns the specified spell from the race
Spell Function GetNthSpell(int n) native

; returns whether the specified race flag is set
bool Function IsRaceFlagSet(int n) native

; sets the specified race flag
Function SetRaceFlag(int n) native

; clears the specified race flag
Function ClearRaceFlag(int n) native

; Returns the races default voice type
VoiceType Function GetDefaultVoiceType(bool female) native

; Sets the races default voice type
Function SetDefaultVoiceType(bool female, VoiceType voice) native

; Gets/sets the skin of the race
Armor Function GetSkin() native
Function SetSkin(Armor skin) native

; Returns the number of playable races
int Function GetNumPlayableRaces() native global

; Returns the nth playable race
Race Function GetNthPlayableRace(int n) native global

; Returns a race by it's editorId name
Race Function GetRace(string editorId) native global

; race flags for previous functions
int property kRace_Playable                      = 0x00000001 AutoReadOnly
int property kRace_FaceGenHead                   = 0x00000002 AutoReadOnly
int property kRace_Child                         = 0x00000004 AutoReadOnly
int property kRace_TiltFrontBack                 = 0x00000008 AutoReadOnly
int property kRace_TiltLeftRight                 = 0x00000010 AutoReadOnly
int property kRace_NoShadow                      = 0x00000020 AutoReadOnly
int property kRace_Swims                         = 0x00000040 AutoReadOnly
int property kRace_Flies                         = 0x00000080 AutoReadOnly
int property kRace_Walks                         = 0x00000100 AutoReadOnly
int property kRace_Immobile                      = 0x00000200 AutoReadOnly
int property kRace_NotPushable                   = 0x00000400 AutoReadOnly
int property kRace_NoCombatInWater               = 0x00000800 AutoReadOnly
int property kRace_NoRotatingToHeadTrack         = 0x00001000 AutoReadOnly
int property kRace_UseHeadTrackAnim              = 0x00008000 AutoReadOnly
int property kRace_SpellsAlignWithMagicNode      = 0x00010000 AutoReadOnly
int property kRace_UseWorldRaycasts              = 0x00020000 AutoReadOnly
int property kRace_AllowRagdollCollision         = 0x00040000 AutoReadOnly
int property kRace_CantOpenDoors                 = 0x00100000 AutoReadOnly
int property kRace_AllowPCDialogue               = 0x00200000 AutoReadOnly
int property kRace_NoKnockdowns                  = 0x00400000 AutoReadOnly
int property kRace_AllowPickpocket               = 0x00800000 AutoReadOnly
int property kRace_AlwaysUseProxyController     = 0x01000000 AutoReadOnly
int property kRace_AllowMultipleMembraneShaders = 0x20000000 AutoReadOnly
int property kRace_AvoidsRoads                  = 0x80000000 AutoReadOnly

bool Function IsPlayable()
    return IsRaceFlagSet(self.kRace_Playable)
endFunction

Function MakePlayable()
    SetRaceFlag(self.kRace_Playable)
endFunction

Function MakeUnplayable()
    ClearRaceFlag(self.kRace_Playable)
endFunction

bool Function IsChildRace()
    return IsRaceFlagSet(self.kRace_Child)
endFunction

Function MakeChildRace()
    SetRaceFlag(self.kRace_Child)
endFunction

Function MakeNonChildRace()
    ClearRaceFlag(self.kRace_Child)
endFunction

bool Function CanFly()
    return IsRaceFlagSet(self.kRace_Flies)
endFunction

Function MakeCanFly()
    SetRaceFlag(self.kRace_Flies)
endFunction

Function MakeNonFlying()
    ClearRaceFlag(self.kRace_Flies)
endFunction

bool Function CanSwim()
    return IsRaceFlagSet(self.kRace_Swims)
endFunction

Function MakeCanSwim()
    SetRaceFlag(self.kRace_Swims)
endFunction

Function MakeNonSwimming()
    ClearRaceFlag(self.kRace_Swims)
endFunction

bool Function CanWalk()
    return IsRaceFlagSet(self.kRace_Walks)
endFunction

Function MakeCanWalk()
    SetRaceFlag(self.kRace_Walks)
endFunction

Function MakeNonWalking()
    ClearRaceFlag(self.kRace_Walks)
endFunction

bool Function IsImmobile()
    return IsRaceFlagSet(self.kRace_Immobile)
endFunction

Function MakeImmobile()
```

```
        SetRaceFlag(self.kRace_Immobile)
    endFunction

    Function MakeMobile()
        ClearRaceFlag(self.kRace_Immobile)
    endFunction

    bool Function IsNotPushable()
        return IsRaceFlagSet(self.kRace_NotPushable)
    endFunction

    Function MakeNotPushable()
        SetRaceFlag(self.kRace_NotPushable)
    endFunction

    Function MakePushable()
        ClearRaceFlag(self.kRace_NotPushable)
    endFunction

    bool Function NoKnockdowns()
        return IsRaceFlagSet(self.kRace_AllowPickpocket)
    endFunction

    Function MakeNoKnockdowns()
        SetRaceFlag(self.kRace_AllowPickpocket)
    endFunction

    Function ClearNoKNockdowns()
        ClearRaceFlag(self.kRace_AllowPickpocket)
    endFunction

    bool Function NoCombatInWater()
        return IsRaceFlagSet(self.kRace_NoCombatInWater)
    endFunction

    Function SetNoCombatInWater()
        SetRaceFlag(self.kRace_NoCombatInWater)
    endFunction

    Function ClearNoCombatInWater()
        ClearRaceFlag(self.kRace_NoCombatInWater)
    endFunction

    bool Function AvoidsRoads()
        return IsRaceFlagSet(self.kRace_AvoidsRoads)
    endFunction

    Function SetAvoidsRoads()
        SetRaceFlag(self.kRace_AvoidsRoads)
    endFunction

    Function ClearAvoidsRoads()
        ClearRaceFlag(self.kRace_AvoidsRoads)
    endFunction

    bool Function AllowPickpocket()
        return IsRaceFlagSet(self.kRace_AllowPickpocket)
    endFunction

    Function SetAllowPickpocket()
        SetRaceFlag(self.kRace_AllowPickpocket)
    endFunction

    Function ClearAllowPickpocket()
        ClearRaceFlag(self.kRace_AllowPickpocket)
    endFunction

    bool Function AllowPCDialogue()
        return IsRaceFlagSet(self.kRace_AllowPCDialogue)
```

```
endFunction

Function SetAllowPCDialogue()
    SetRaceFlag(self.kRace_AllowPCDialogue)
endFunction

Function ClearAllowPCDialogue()
    ClearRaceFlag(self.kRace_AllowPCDialogue)
endFunction

bool Function CantOpenDoors()
    return IsRaceFlagSet(self.kRace_CantOpenDoors)
endFunction

Function SetCantOpenDoors()
    SetRaceFlag(self.kRace_CantOpenDoors)
endFunction

Function ClearCantOpenDoors()
    ClearRaceFlag(self.kRace_CantOpenDoors)
endFunction

bool Function NoShadow()
    return IsRaceFlagSet(self.kRace_NoShadow)
endFunction

Function SetNoShadow()
    SetRaceFlag(self.kRace_NoShadow)
endFunction

Function ClearNoShadow()
    ClearRaceFlag(self.kRace_NoShadow)
endFunction

   :: Add a newline between files
Scriptname Scroll extends Form Hidden

; Cast this scroll from an ObjectReference, optionally toward another.
Function Cast(ObjectReference akSource, ObjectReference akTarget=NONE) native

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; return the casting time
float Function GetCastTime() native

; return the perk associated with the spell
Perk Function GetPerk() native

; return the number of the effects
int Function GetNumEffects() native

; return the magnitude of the specified effect
float Function GetNthEffectMagnitude(int index) native

; return the area of the specified effect
int Function GetNthEffectArea(int index) native

; return the duration of the specified effect
int Function GetNthEffectDuration(int index) native

; return the magic effect of the specified effect
MagicEffect Function GetNthEffectMagicEffect(int index) native

; return the index of the costliest effect
int Function GetCostliestEffectIndex() native

; sets the magnitude of the specified effect
Function SetNthEffectMagnitude(int index, float value) native

; sets the area of the specified effect
```

```
Function SetNthEffectArea(int index, int value) native

; sets the duration of the specified effect
Function SetNthEffectDuration(int index, int value) native

; Returns the particular equipslot type
EquipSlot Function GetEquipType() native
Function SetEquipType(EquipSlot type) native

; Returns all the magnitudes of this object in order
float[] Function GetEffectMagnitudes() native

; Returns all the areas of this object in order
int[] Function GetEffectAreas() native

; Returns all the durations of this object in order
int[] Function GetEffectDurations() native

; Returns all the magic effects of this object in order
MagicEffect[] Function GetMagicEffects() native   :: Add a newline between files
Scriptname Shout extends Form Hidden

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
WordOfPower Function GetNthWordOfPower(int n) native
Spell Function GetNthSpell(int n) native
float Function GetNthRecoveryTime(int n) native

Function SetNthWordOfPower(int n, WordOfPower aWoop) native
Function SetNthSpell(int n, Spell aSpell) native
Function SetNthRecoveryTime(int n, float time) native   :: Add a newline between files
Scriptname SKSE Hidden
; General SKSE-specific information

; get the major version of SKSE
int Function GetVersion() global native
; get the minor version of SKSE
int Function GetVersionMinor() global native
; get the beta version of SKSE
int Function GetVersionBeta() global native
; get the release index of SKSE.  This number is incremented every time
; SKSE is released outside of the development team
int Function GetVersionRelease() global native
; get the release index of this script file.
; Can be used to detect a script/runtime version mismatch
int Function GetScriptVersionRelease() global
    return 72
endFunction

; get a plugins version number, -1 if the plugin is not loaded
int Function GetPluginVersion(string name) global native   :: Add a newline between files
Scriptname SoulGem extends MiscObject Hidden

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

int Function GetSoulSize() native
int Function GetGemSize() native   :: Add a newline between files
Scriptname Sound extends Form Hidden
import ObjectReference

; Play this sound base object from the specified source
int Function Play(ObjectReference akSource) native

; Play this sound from the specified source, and wait for it to finish
bool Function PlayAndWait(ObjectReference akSource) native

; Stops a given playback instance of a sound
Function StopInstance(int aiPlaybackInstance) native global

; Set the volume of a given playback instance of a sound. Clamped between 0 and 1.
```

```
Function SetInstanceVolume(int aiPlaybackInstance, float afVolume) native global



; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
SoundDescriptor Function GetDescriptor() native   :: Add a newline between files
Scriptname SoundDescriptor extends Form Hidden

float Function GetDecibelAttenuation() native
Function SetDecibelAttenuation(float dbAttenuation) native

int Function GetDecibelVariance() native
Function SetDecibelVariance(int dbVariance) native

int Function GetFrequencyVariance() native
Function SetFrequencyVariance(int frequencyVariance) native

int Function GetFrequencyShift() native
Function SetFrequencyShift(int frequencyShift) native   :: Add a newline between files
Scriptname SpawnerTask Hidden

; SpawnerTask allows to spawn and position an arbitrary number references in game world.
; It's effectively a batch combination of PlaceAtMe and SetPosition/MoveTo that smoothly executes in a single frame.
;
; Example:
;
;       ObjectReference player   = ...
;       Form            chair    = ...
;       float[]         offset   = new float[3]
;       float[]         rotation = new float[3]
;
;       ; Allocate new task
;       int taskId = SpawnerTask.Create()
;
;       ; No rotation
;       rotation[0] = 0
;       rotation[1] = 0
;       rotation[2] = 0
;
;       ; Spawn 100 chairs in a grid above the player
;       int i = 0
;       while i < 100
;           offset[0] = -250 + (i / 10) * 50
;           offset[1] = -250 + (i % 10) * 50
;           offset[2] = 200
;
;           SpawnerTask.AddSpawn(taskId, chair, player, offset, rotation)
;           i += 1
;       endWhile
;
;       ; Run the task and return all placed references in an array
;       ObjectReference[] spawned = SpawnerTask.Run(taskId)


; Creates a new SpawnerTask and returns a handle, which is an identifier for the created task.
; The task handle is valid until the task has been run or canceled, or until the calling stack has exited.
;   (Function type: non-delayed)
;
int Function Create() global native

; Adds a spawn to the task identified by the given handle.
; Running the task places a new instance of formToPlace at target reference with given rotation and position offset. Parameters are analogously defined to PlaceAtMe.
; Multiple spawns can be added to the same task to be executed in a batch (which is the purpose).
;   (Function type: non-delayed)
```

```
;
Function AddSpawn(int handle, Form formToPlace, ObjectReference target, float[] positionOffset, float[] rotation, int count = 1, bool bForcePersist = false, bool bInitiallyDisabled = false) global native

; Runs the task and returns the spawned references in an array. May return arrays with a size larger than 128.
; The resources allocated to the task are freed in the process, so the same task handle cannot be run twice.
;   (Function type: latent)
;
ObjectReference[] Function Run(int handle) global native

; Cancels a task before running it and frees its allocated resources.
; Tasks cannot be canceled once they have been started with Run, and vice versa.
;
Function Cancel(int handle) global native
   :: Add a newline between files
Scriptname Spell extends Form Hidden

; Cast this spell from an ObjectReference, optionally toward another.
Function Cast(ObjectReference akSource, ObjectReference akTarget=NONE) native

; Cast this spell from an ObjectReference, optionally toward another, and blame it on a particular actor.
Function RemoteCast(ObjectReference akSource, Actor akBlameActor, ObjectReference akTarget=NONE) native

; Is this spell classified as hostile?
bool Function IsHostile() native

; Preload the art for this spell. Useful for spells you equip & unequip on the player.
; Warning: Misuse of this function can lead to erroneous behavior as well as excessive
; memory consumption. It's best to avoid using this. This function will likely be
; deprecated in the future.
Function Preload() native

; Unload the art for this spell. Call this only if you've previously called Preload.
; Warning: Misuse of this function can lead to erroneous behavior including spell art
; being unloaded while in use, and excessive memory consumption. It's best to avoid using this.
; This function will likely be deprecated in the future.
Function Unload() native


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC
; return the casting time
float Function GetCastTime() native

; return the perk associated with the spell
Perk Function GetPerk() native

; return the number of the effects
int Function GetNumEffects() native

; return the magnitude of the specified effect
float Function GetNthEffectMagnitude(int index) native

; return the area of the specified effect
int Function GetNthEffectArea(int index) native

; return the duration of the specified effect
int Function GetNthEffectDuration(int index) native

; return the magic effect of the specified effect
MagicEffect Function GetNthEffectMagicEffect(int index) native

; return the index of the costliest effect
int Function GetCostliestEffectIndex() native
```

```

; return the base magicka cost of the spell
int Function GetMagickaCost() native

; return the effective magicka cost of the spell for given caster
int Function GetEffectiveMagickaCost(Actor caster) native

; sets the magnitude of the specified effect
Function SetNthEffectMagnitude(int index, float value) native

; sets the area of the specified effect
Function SetNthEffectArea(int index, int value) native

; sets the duration of the specified effect
Function SetNthEffectDuration(int index, int value) native

; Returns the particular equipslot type
EquipSlot Function GetEquipType() native
Function SetEquipType(EquipSlot type) native

; Returns all the magnitudes of this object in order
float[] Function GetEffectMagnitudes() native

; Returns all the areas of this object in order
int[] Function GetEffectAreas() native

; Returns all the durations of this object in order
int[] Function GetEffectDurations() native

; Returns all the magic effects of this object in order
MagicEffect[] Function GetMagicEffects() native   :: Add a newline between files
Scriptname StringUtil Hidden

; Note about the internal Skyrim implementation of the string classes used for scripting:
; the strings are case-insensitive.  Each BSFixedString is managed in a cache and reused
; everywhere it is needed.  This means that strings like "O" and "o" are technically equivalent;
; Which string is used depends greatly on which version is found first.  We are investigating
; how to manage this, but for the time being be aware that the distinction between uppercase
; and lowercase may not exist.  It also means that functions below returning an integer
; for the character may not correspond exactly.  Also GetNthChar("Hello Skyrim!", 4) will
; return a string with either "O" or "o" depnding on which might be registered first.  All
; my tests so far have it return the uppercase, eventhough in the string it is lowercase.
; We may solve this problem by switching back to returning an integer rather than a string
; for GetNthChar, but this will still have problems.

; return the length of the string
int Function GetLength(string s) global native

; returns a single character string with the character at index
string Function GetNthChar(string s, int index) global native

; Functions to work on Chars
; returns information about a specific character
; assumes a single character string.  If a multicharacter string is passed
; the information about the first character is returned
bool Function IsLetter(string c) global native
bool Function IsDigit(string c) global native
bool Function IsPunctuation(string c) global native
bool Function IsPrintable(string c) global native

; returns the index of the first character of toFind inside string s
; returns -1 if toFind is not part of the string or if startIndex is invalid
int Function Find(string s, string toFind, int startIndex = 0) global native

```

```
; returns a substring of the specified string starting at startIndex and going for len characters
; or until the end of the string.  Default len of 0 means for the entire string
string Function Substring(string s, int startIndex, int len = 0) global native

; returns the numeric value of the first character as an int
int Function AsOrd(string c) global native

; returns a single character string interpreting c as a character
string Function AsChar(int c) global native

; returns array of strings separated by the specified delimiter
string[] Function Split(string s, string delim) global native   :: Add a newline between files
Scriptname TextureSet extends Form Hidden


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

; Returns the number of texture paths
int Function GetNumTexturePaths() native

; Returns the path of the texture
string Function GetNthTexturePath(int n) native

; Sets the path of the texture
Function SetNthTexturePath(int n, string texturePath) native   :: Add a newline between files
Scriptname TreeObject extends Form Hidden

SoundDescriptor Function GetHarvestSound() native
Function SetHarvestSound(SoundDescriptor akSoundDescriptor) native

Form Function GetIngredient() native
Function SetIngredient(Form akIngredient) native   :: Add a newline between files
Scriptname UI Hidden

; For functions that require a menuName, potential values are
;   "InventoryMenu"
;   "Console"
;   "Dialogue Menu"
;   "HUD Menu"
;   "Main Menu"
;   "MessageBoxMenu"
;   "Cursor Menu"
;   "Fader Menu"
;   "MagicMenu"
;   "Top Menu"
;   "Overlay Menu"
;   "Overlay Interaction Menu"
;   "Loading Menu"
;   "TweenMenu"
;   "BarterMenu"
;   "GiftMenu"
;   "Debug Text Menu"
;   "MapMenu"
;   "Lockpicking Menu"
;   "Quantity Menu"
;   "StatsMenu"
;   "ContainerMenu"
;   "Sleep/Wait Menu"
;   "LevelUp Menu"
;   "Journal Menu"
;   "Book Menu"
;   "FavoritesMenu"
;   "RaceSex Menu"
;   "Crafting Menu"
;   "Training Menu"
;   "Mist Menu"
;   "Tutorial Menu"
;   "Credits Menu"
;   "TitleSequence Menu"
;   "Console Native UI Menu"
;   "Kinect Menu"
;
; The target parameter requires one the following prefixes:
;   _global     , for the default namespace;
;   _root       , for the movie root.


; Returns if the menu is currently open.
bool Function IsMenuOpen(string menuName) global native


; Sets bool/number/string value at target location.
; Target value must already exist.
;
;   Examples:
;       UI.SetBool("InventoryMenu", "_root.Menu_mc._visible", false)
;       UI.SetString("FavoritesMenu", "_root.Menu_mc.panel.message.text", "My Text")
;
Function SetBool(string menuName, string target, bool value) global native
Function SetInt(string menuName, string target, int value) global native
Function SetFloat(string menuName, string target, float value) global native
Function SetString(string menuName, string target, string value) global native
Function SetNumber(string menuName, string target, float value) global ; DEPRECIATED
    SetFloat(menuName, target, value)
EndFunction

; Gets bool/number/string from target location, or false/0/none if the value doesn't
exist.
;
;   Examples:
;       bool    visible = UI.GetBool("Inventory Menu", "_root.Menu_mc._visible")
;       float   height  = UI.GetNumber("Magic Menu", "_root.Menu_mc._height")
;
bool    Function GetBool(string menuName, string target) global native
int     Function GetInt(string menuName, string target) global native
float   Function GetFloat(string menuName, string target) global native
string  Function GetString(string menuName, string target) global native
float   Function GetNumber(string menuName, string target) global ; DEPRECIATED
    return GetFloat(menuName, target)
EndFunction


; Invokes the ActionScript function at given target location.
;
;   Examples:
;       UI.InvokeString("InventoryMenu", "_global.skse.Log", "Printed to logfile")
;       UI.InvokeStringA("InventoryMenu", "_global.myFunction", myArray)
;
Function Invoke(string menuName, string target) global
    InvokeBool(menuName, target, false)
EndFunction

Function InvokeBool(string menuName, string target, bool arg) global native
Function InvokeInt(string menuName, string target, int arg) global native
Function InvokeFloat(string menuName, string target, float arg) global native
Function InvokeString(string menuName, string target, string arg) global native
Function InvokeNumber(string menuName, string target, float arg) global ; DEPRECIATED
    InvokeFloat(menuName, target, arg)
EndFunction

Function InvokeBoolA(string menuName, string target, bool[] args) global native
Function InvokeIntA(string menuName, string target, int[] args) global native
Function InvokeFloatA(string menuName, string target, float[] args) global native
Function InvokeStringA(string menuName, string target, string[] args) global native
Function InvokeNumberA(string menuName, string target, float[] args) global ; DEPRECIATED
```

```
    InvokeFloatA(menuName, target, args)
EndFunction

; Sends Form data to Scaleform as a Flash object, FormLists included.
Function InvokeForm(string menuName, string target, Form arg) global native

; returns if scaleform is in 'text input' mode
; this is useful for ignoring keys that should get swallowed by an editable text box
bool Function IsTextInputEnabled() global native

; open a custom menu named "CustomMenu" by loading the given swf from the interface folder
; (filename without extension)
; there can only be a single custom menu open at the same time
Function OpenCustomMenu(string swfPath, int flags = 0) global native

; close the custom menu if it's currently open.
Function CloseCustomMenu() global native   :: Add a newline between files
Scriptname UICallback Hidden

; UICallback allows passing arguments of different types to UI functions, unlike UI.Invoke*
;
; Example:
;   int handle = UICallback.Create("InventoryMenu", "_global.MyClass.initData")
;   if (handle)
;       UICallback.PushBool(handle, true)
;       UICallback.PushInt(handle, 1000)
;       UICallback.PushString(handle, "Hello")
;       UICallback.PushFloat(handle, 1.234)
;       UIDelegate.Send(handle)
;   endIf
;
; Any UICallback allocated by Create must be released later.
; That happens automatically when passing it to send.
; Otherwise the handle must be manually released by passing it to Release.
;
; Internally, UICallback objects only persist for the duration of the current
; game session. They are also cleared after each reload.
;
; This means that in very rare cases, the execution sequence of several operations
; on one UICallback might get interrupted, the handle turns invalid and the final Send
; will fail. If its necessary to detect this, check the return value of Send.

; Creates a new UICallback and returns the handle.
int Function Create(string menuName, string target) global native

; Invokes the UICallback and releases it.
; Returns true, if it was executed, false if an error happened.
bool Function Send(int handle) global native

; Releases the UICallback without sending it.
Function Release(int handle) global native

; Push single parameter. Maximum number of parameters per callback is 128.
Function PushBool(int handle, bool value) global native
Function PushInt(int handle, int value) global native
Function PushFloat(int handle, float value) global native
Function PushString(int handle, string value) global native

; Push parameters from array. Maximum number of parameters per callback is 128.
Function PushBoolA(int handle, bool[] args) global native
Function PushIntA(int handle, int[] args) global native
Function PushFloatA(int handle, float[] args) global native
Function PushStringA(int handle, string[] args) global native   :: Add a newline between files
Scriptname Utility Hidden

; Converts a float game time (in terms of game days passed) to a string detailing the
```

```
date
; and time it represents in "MM/DD/YYYY HH:MM" format. A 24-hour clock is used, and the function
; is latent (due to issues in the current architecture with returning strings from code)
string Function GameTimeToString(float afGameTime) native global

; Obtains the current game time in terms of game days passed (same as the global variable)
float Function GetCurrentGameTime() native global

; Obtains the number of seconds since the application started (the same timer that WaitMenuMode uses)
; Does not take into account menu-mode, or VM frozen time
; Most useful for determining how long something took to run
float Function GetCurrentRealTime() native global

; Returns whether the game is currently in menu mode or not
bool Function IsInMenuMode() native global

; Generates a random integer between aiMin and aiMax (inclusive)
int Function RandomInt(int aiMin = 0, int aiMax = 100) native global

; Generates a random floating point number between afMin and afMax (inclusive)
float Function RandomFloat(float afMin = 0.0, float afMax = 1.0) native global

; Set the given INI by type
function SetINIFloat(string ini, float value) native global
function SetINIInt(string ini, int value) native global
function SetINIBool(string ini, bool value) native global
function SetINIString(string ini, string value) native global

; Waits for the specified amount of time (latent). Timer will not run during menu mode
Function Wait(float afSeconds) native global

; Waits for the specified amount of game time (latent)
Function WaitGameTime(float afHours) native global

; Waits for the specified amount of time (latent) - Timer WILL run during menu mode
Function WaitMenuMode(float afSeconds) native global

; Frame rate capture functions only available in beta version

; Gets you a string describing the frame rate for a certain number of frames
; (String will be no longer than 1K characters long, separated by commas)
string Function CaptureFrameRate(int numFrames) native global

; Starts or ends a frame rate capture -- then you can get the min or max since
; frame capture started at any time
Function StartFrameRateCapture() native global
Function EndFrameRateCapture() native global
float Function GetAverageFrameRate() native global
float Function GetMinFrameRate() native global
float Function GetMaxFrameRate() native global

; Memory tracking functions - only available if memory tracking is turned on
int Function GetCurrentMemory() native global ; Must be called first, it sets up the memory stats used by the other functions
int Function GetBudgetCount() native global
int Function GetCurrentBudget(int aiBudgetNumber) native global
bool Function OverBudget(int aiBudgetNumber) native global
string Function GetBudgetName(int aiBudgetNumber) native global

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

float Function GetINIFloat(string ini) global native
int Function GetINIInt(string ini) global native
bool Function GetINIBool(string ini) global native
string Function GetINIString(string ini) global native

```

```

; Size is treated as unsigned, negative numbers will result
; extremely large positive numbers, USE WITH CARE
float[] Function CreateFloatArray(int size, float fill = 0.0) global native
int[] Function CreateIntArray(int size, int fill = 0) global native
bool[] Function CreateBoolArray(int size, bool fill = false) global native
string[] Function CreateStringArray(int size, string fill = "") global native
Form[] Function CreateFormArray(int size, Form fill = None) global native
Alias[] Function CreateAliasArray(int size, Alias fill = None) global native

float[] Function ResizeFloatArray(float[] source, int size, float fill = 0.0) global
native
int[] Function ResizeIntArray(int[] source, int size, int fill = 0) global native
bool[] Function ResizeBoolArray(bool[] source, int size, bool fill = false) global native
string[] Function ResizeStringArray(string[] source, int size, string fill = "") global
native
Form[] Function ResizeFormArray(Form[] source, int size, Form fill = None) global native
Alias[] Function ResizeAliasArray(Alias[] source, int size, Alias fill = None) global
native   :: Add a newline between files
Scriptname Weapon extends Form Hidden

; Fire this weapon base object from the specified source
Function Fire(ObjectReference akSource, Ammo akAmmo = None) native


; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

int Function GetBaseDamage() native
Function SetBaseDamage(int damage) native

int Function GetCritDamage() native
Function SetCritDamage(int damage) native

float Function GetReach() native
Function SetReach(float reach) native

float Function GetMinRange() native
Function SetMinRange(float minRange) native

float Function GetMaxRange() native
Function SetMaxRange(float maxRange) native

float Function GetSpeed() native
Function SetSpeed(float speed) native

float Function GetStagger() native
Function SetStagger(float stagger) native

int Function GetWeaponType() native
Function SetWeaponType(int type) native

; works on the path to the nif file representing the in-game model of the weapon
string Function GetModelPath() native
Function SetModelPath(string path) native

; works on the path to the nif file representing the icon for the weapon in the inventory
string Function GetIconPath() native
Function SetIconPath(string path) native

; works on the path to the file representing the message icon for the weapon
string Function GetMessageIconPath() native
Function SetMessageIconPath(string path) native

; works on the enchantment associated with the weapon
Enchantment Function GetEnchantment() native
Function SetEnchantment(Enchantment e) native

; works on the enchantment value of the associated weapon
int Function GetEnchantmentValue() native
```

```
Function SetEnchantmentValue(int value) native

; works on the weapon model when equipped of the associated weapon
Static Function GetEquippedModel() native
Function SetEquippedModel(Static model) native

; Returns the particular equipslot type
EquipSlot Function GetEquipType() native
Function SetEquipType(EquipSlot type) native

string Function GetSkill() native
Function SetSkill(string skill) native

; DamageResist
; ElectricResist
; FireResist
; FrostResist
; MagicResist
; PoisonResist
string Function GetResist() native
Function SetResist(string resist) native

; works on the spell that applies when critting
Spell Function GetCritEffect() native
Function SetCritEffect(Spell ce) native

; Gets, sets or unsets whether the the crit effect should only occur on death
bool Function GetCritEffectOnDeath() native
Function SetCritEffectOnDeath(bool ceod) native

; Gets/sets the weapons crit multiplier
float Function GetCritMultiplier() native
Function SetCritMultiplier(float crit) native

; returns the weapon template of this weapon
Weapon Function GetTemplate() native

bool Function IsBattleaxe()
    return HasKeywordString("WeapTypeBattleaxe")
endFunction

bool Function IsBow()
    return HasKeywordString("WeapTypeBow")
endFunction

bool Function IsDagger()
    return HasKeywordString("WeapTypeDagger")
endFunction

bool Function IsGreatsword()
    return HasKeywordString("WeapTypeGreatsword")
endFunction

bool Function IsMace()
    return HasKeywordString("WeapTypeMace")
endFunction

bool Function IsStaff()
    return HasKeywordString("WeapTypeStaff")
endFunction

bool Function IsSword()
    return HasKeywordString("WeapTypeSword")
endFunction

bool Function IsWarhammer()
    return HasKeywordString("WeapTypeWarhammer")
endFunction

```

```
bool Function IsWarAxe()
    return HasKeywordString("WeapTypeWarAxe")
endFunction
   :: Add a newline between files
Scriptname Weather extends Form Hidden

; Tells the sky to release its overriding weather.
function ReleaseOverride() native global

; Forces the active weather on the sky to be this weather.
function ForceActive( bool abOverride=false ) native

; Sets the active weather on the sky to be this weather.
function SetActive( bool abOverride=false, bool abAccelerate=false ) native

; Finds a weather from the current region/climate whose classification matches the given one.
; 0 - Pleasant
; 1 - Cloudy
; 2 - Rainy
; 3 - Snow
Weather function FindWeather( int auiType ) native global

; Gets this weather's classification
; -1 - No classification
;  0 - Pleasant
;  1 - Cloudy
;  2 - Rainy
;  3 - Snow
int function GetClassification() native

; Gets the sky's current weather
Weather function GetCurrentWeather() native global

; Gets the sky's outgoing weather
Weather function GetOutgoingWeather() native global

; Gets the transition percentage of the current weather
float function GetCurrentWeatherTransition() native global

; Gets the sky's current mode
; 0 - No sky (SM_NONE)
; 1 - Interior (SM_INTERIOR)
; 2 - Skydome only (SM_SKYDOME_ONLY)
; 3 - Full sky (SM_FULL)
int function GetSkyMode() native global

; SKSE64 additions built 2024-01-17 20:01:40.731000 UTC

; Returns the sun glare percentage
float Function GetSunGlare() native

; Returns the sun damage percentage
float Function GetSunDamage() native

; Returns the wind direction in degrees (0-360)
float Function GetWindDirection() native

; Returns the wind direction range in degrees (0-180)
float Function GetWindDirectionRange() native

; 0 - Near
; 1 - Far
; 2 - Power
; 3 - Max
float Function GetFogDistance(bool day, int type) native
   :: Add a newline between files
Scriptname WornObject Hidden

```

```
; These functions operate directly on
; worn items within the inventory
; Valid Hand Slot:
; 0 - Left
; 1 - Right
; Valid Slot Masks:
; See Armor.psc
; Use zero when using hand slot

; Tempering
float Function GetItemHealthPercent(Actor akActor, int handSlot, int slotMask) global native
Function SetItemHealthPercent(Actor akActor, int handSlot, int slotMask, float health) global native

; Charges
; Only works on items that have user-enchants
Function SetItemMaxCharge(Actor akActor, int handSlot, int slotMask, float maxCharge) global native

; Works on any enchanted item
float Function GetItemMaxCharge(Actor akActor, int handSlot, int slotMask) global native

float Function GetItemCharge(Actor akActor, int handSlot, int slotMask) global native

; Use LeftItemCharge/RightItemCharge ActorValues instead
;Function SetItemCharge(Actor akActor, int handSlot, int slotMask, float charge) global native

; Returns the name of this reference
; this is the name that is displayed
string Function GetDisplayName(Actor akActor, int handSlot, int slotMask) global native

; Sets a reference's display name
; returns false if force is false and the reference
; is held by an alias using 'Stored Text' or 'Uses Stored Text'
; Text Replacement does not use this name and may be lost if forced
bool Function SetDisplayName(Actor akActor, int handSlot, int slotMask, string name, bool force = false) global native

; Returns the player-made enchantment if there is one
Enchantment Function GetEnchantment(Actor akActor, int handSlot, int slotMask) global native

; Changes an item's player-made enchantment to something else
; None enchantment will remove the existing enchantment
; does not delete the custom enchantment, only removes it
Function SetEnchantment(Actor akActor, int handSlot, int slotMask, Enchantment source, float maxCharge) global native

; Creates a new enchantment on the item given the specified parameters
; all arrays must be the same size
; created enchantments are not purged from the save when removed or overwritten
; exact same enchantments are re-used by the game
Function CreateEnchantment(Actor akActor, int handSlot, int slotMask, float maxCharge, MagicEffect[] effects, float[] magnitudes, int[] areas, int[] durations) global native

; Returns the number of ref aliases holding this reference
int Function GetNumReferenceAliases(Actor akActor, int handSlot, int slotMask) global native

; Returns the nth ReferenceAlias holding this reference
ReferenceAlias Function GetNthReferenceAlias(Actor akActor, int handSlot, int slotMask, int n) global native

; Returns the poison on the specified item
Potion Function GetPoison(Actor akActor, int handSlot, int slotMask) global native

; Returns all of the ReferenceAlias holding this reference
```

```
ReferenceAlias[] Function GetReferenceAliases(Actor akActor, int handSlot, int slotMask)
global native   :: Add a newline between files

```