Creating Custom Couriers

This tutorial outlines the steps to make a custom courier that (sort of) works in a manner similar to the existing ones, but will track you down anywhere.

It is fairly complicated and, like the existing couriers, driven by a quest.

Contents

Components

NPCs

First, make a copy of the NPC you want to use and strip off the AI information so it has no packages and no agro radius behavior. Make it cowardly and "helps nobody" as well. Make as many variations as you want, and give them a name easy to find (Like prefixing it with WICC (World Interactions Custom Courier))

Containers

Then, make a copy of any container in the game and empty out the loot options. Give it an easy to find name as above, perhaps WICCC prefix (WI Custom Courier Container)

Global

Make a custom global variable for the Item Count (the number of items in the current container)

The Cell

- Next, Make a copy of the courier cell, or create a new cell of your own if you prefer.
- Put an instance of each of your stripped down NPCs in it, and set them all to "Starts Disabled".
- Make one or more Custom Container instances in there as well (You'll see why in a moment)
- You will also need one Xmarker with a distinctive name (WICCXMarker for example)

That's all you need in the cell.

Create a copy of a small hard to see object (A charcoal stick works well) and give it a memorable name (WICCDropPoint for example).

Create a Keyword like "WICCSpawnPointKeyword"

Create an XMarkerActivator type that has the afforementioned Keyword attached. Place at least 3 of these in hard to see spots in major cities that have their own cells like Solitude, Markarth, Windhelm, etc. The reason for this is, you don't want the courier to spawn outside the walls of the town if the player is inside, since this will make it so the courier cannot reach the player.

The Quest

Build a quest that starts game enabled and allows repeated stages,

- Make an alias for each of your Custom Courier types, picking them in the render window.
- Make an alias for each of the containers, also picking them in the window.
- Make an alias "Courier" and an alias "Container" and pick the courier and container you want to use by default.
- The last alias is one for the cell marker.

Packages

The Courier alias should have to following packages attached to it, in this order:

1. A travel package, targeted on the player, with the conditions:

   GetDistance (Courier to Player) > 1000.0
   GetGlobalValue(Your global item count variable) >= 1.0

2. A follow player package with just the Item Count test.
3. A "Flee from target" package set to the player with no conditions.

This will make the courier travel to the player and then follow him so long as there is something in the container. Once the container is empty, the courier will then leave.

Quest Stages

The Quest stages and their associated fragments should be:

Stage 0

```
; Startup Stage - Return here when Deliveries complete.
;Debug.Notification("Custom Courier in his home cell")
        UnregisterForUpdate() ; Make sure to cancel previous delivery's updates, so
we're not registered twice.
        RegisterForUpdate(20)
        CourierScript.StateChange("Waiting")
```

Stage 100

```
; Item added to courier box. Look for a place to start.
; Check optional references now.
; Debug.Notification("Custom Courier has a letter!")

CourierScript.StateChange("Seeking")
```

Stage 200

```
; Courier found the player. Make Delivery

CourierScript.StateChange("Delivering")
DeliverScene.Start()
```

Stage 300

```
; Delivery Complete. Courier will run away now.

CourierScript.StateChange("Departing")
```

Delivery Scene

Create a scene to define the sort of behavior you want your couriers to have when they reach the player.

In the delivery scene, be sure to create an appropriate dialog topic, and have the scene itself transfer the objects with this fragment when the scene ends:

```
CourierScript.GiveItemsToPlayer()
```

Quest Script

Here is the Courier Script I used:

```
Scriptname WICustomCourierScript extends Quest

import Utility

ReferenceAlias Property CourierCellMarker Auto

ReferenceAlias Property Courier Auto ; The courier currently in use. Alternate
versions below.
ReferenceAlias Property WolfCourier Auto
ReferenceAlias Property SpiderCourier Auto
ReferenceAlias Property WerewolfCourier Auto
ReferenceAlias Property DeerCourier Auto
ReferenceAlias Property FoxCourier Auto
ReferenceAlias Property FlameCourier Auto
ReferenceAlias Property DwarfSpiderCourier Auto

ReferenceAlias Property ContainerAlias Auto
GlobalVariable Property ItemCount  Auto ; Current container and item count. Alternate
versions below.
ReferenceAlias Property DwLgContainer Auto
int property DwLgCount Auto
ReferenceAlias Property DwMedContainer Auto
int Property DwMedCount Auto
ReferenceAlias Property DwSmContainer Auto
int Property DwSmCount Auto
ReferenceAlias Property HiveContainer Auto
int property HiveCount Auto
ReferenceAlias Property BarrelContainer Auto
int property BarrelCount Auto
ReferenceAlias Property CoffinContainer Auto
int property CoffinCount Auto
```

```
Bool Function DisableDeliveries(Bool DisableMe = True) ; Allows you to turn off
deliveries, in case
; you want to fiddle with the containers and courier settings without interruption.
    If DisableMe
        While GetState() != "Waiting" ; wait until the courier is finished with any
deliveries.
; Just want to make sure it stays disabled once it is.
        EndWhile
        GoToState("Disabled")
        return True
    else
;Debug.Notification("Enabling Courier")
        GoToState("Waiting") ; Courier will enable itself when it has something to
deliver.
        Return False
    endif
EndFunction

Function SetRandomContainer() ; Picks a random container
    int Which = RandomInt(1,6)
    if Which == 1
        ChangeContainer("Hive")
    elseif Which == 2
        ChangeContainer("Coffin")
    elseif Which == 3
        ChangeContainer("Barrel")
    elseif Which == 4
        ChangeContainer("Small")
    elseif Which == 5
        ChangeContainer("Medium")
    else
        ChangeContainer("Large")
    endif
EndFunction

Function SetRandomCourier() ; Picks a random courier.
    int Which = RandomInt(1,7)
    if Which == 1
        ChangeCourier("Wolf")
    elseif Which == 2
        ChangeCourier("Spider")
    elseif Which == 3
        ChangeCourier("Werewolf")
    elseif Which == 4
        ChangeCourier("Deer")
    elseif Which == 5
        ChangeCourier("Fox")
    elseif Which == 6
        ChangeCourier("Flame")
    else
        ChangeCourier("Dwemer")
    endif
EndFunction

Function StoreContainerQuantity()
; Saves quantity of the current container if changing containers, in case there is
something
; still in there.  When we change back, it will restore the value.
    ObjectReference Current = ContainerAlias.GetRef()
    If Current == DwLgContainer.GetRef()
        DwLgCount = (ItemCount.GetValue() as int)
    elseIf Current == DwSmContainer.GetRef()
        DwSmCount = (ItemCount.GetValue() as int)
    elseIf Current == DwMedContainer.GetRef()
        DwMedCount = (ItemCount.GetValue() as int)
    elseIf Current == HiveContainer.GetRef()
        HiveCount = (ItemCount.GetValue() as int)
    elseIf Current == BarrelContainer.GetRef()
        BarrelCount =( ItemCount.GetValue() as int)
    else ; Only the coffin is left.
        CoffinCount = (ItemCount.GetValue() as int)
    endif
EndFunction

Function ChangeCourier(String Which)

;Debug.Notification("Changing Courier: " + Which)

    While Courier.GetRef().IsEnabled() || Updating
; If a courier is currently trying to deliver something, wait til it's done.
        Wait(1.0)
    endWhile
    Updating = True ; Don't spawn a new courier or change containers while changing
courier types.
    if Which == "Wolf"
        Courier.ForceRefTo(WolfCourier.GetRef())
    elseif Which == "Spider"
        Courier.ForceRefTo(SpiderCourier.GetRef())
    elseif Which == "Werewolf"
        Courier.ForceRefTo(WereWolfCourier.GetRef())
    elseif Which == "Deer"
        Courier.ForceRefTo(DeerCourier.GetRef())
    elseif Which == "Fox"
        Courier.ForceRefTo(FoxCourier.GetRef())
    elseif Which == "Flame"
        Courier.ForceRefTo(FlameCourier.GetRef())
    elseif Which == "Dwemer"
        Courier.ForceRefTo(DwarfSpiderCourier.GetRef())
    else ; Default to Dwarven Spider courier
```

```
        Courier.ForceRefTo(DwarfSpiderCourier.GetRef())
    endif
    Updating = False
EndFunction

Function ChangeContainer(String Which)
; Change to the specified container, and update the Item Count to match it's
contents.

;Debug.Notification("Changing Container: " + Which)

    While Courier.GetRef().IsEnabled() || Updating
; If a courier is currently trying to deliver something, wait til it's done.
        Wait(1.0)
    endWhile
    Updating = True
; Don't spawn a new courier or change courier types while we're switching
containers.
    StoreContainerQuantity()
; Saves the quantity in the current container, in case a delivery failed.
; We can go back to it later, but that's the responsibility of the mod writer to
check.
    if Which == "Hive"
        ContainerAlias.ForceRefTo(HiveContainer.GetRef())
        ItemCount.SetValue(HiveCount)
    elseif Which == "Coffin"
        ContainerAlias.ForceRefTo(CoffinContainer.GetRef())
        ItemCount.SetValue(CoffinCount)
    elseif Which == "Barrel"
        ContainerAlias.ForceRefTo(BarrelContainer.GetRef())
        ItemCount.SetValue(BarrelCount)
    elseif Which == "Small"
        ContainerAlias.ForceRefTo(DwSmContainer.GetRef())
        ItemCount.SetValue(DwSmCount)
    elseif Which == "Medium"
        ContainerAlias.ForceRefTo(DwMedContainer.GetRef())
        ItemCount.SetValue(DwMedCount)
    elseif Which == "Large"
        ContainerAlias.ForceRefTo(DwLgContainer.GetRef())
        ItemCount.SetValue(DwLgCount)
    else ; Default to large Dwarven Box container
        ContainerAlias.ForceRefTo(DwLgContainer.GetRef())
        ItemCount.SetValue(DwLgCount)
    endif
    Updating = False
EndFunction

function addItemToContainer(form FormToAdd, int countToAdd = 1)
    ContainerAlias.GetRef().addItem(FormToAdd, countToAdd) ;add parameter object to
container
```

```
    ItemCount.Value += 1
endFunction

function addRefToContainer(objectReference objectRefToAdd)
    ContainerAlias.GetRef().addItem(objectRefToAdd) ;add parameter object to
container
    ItemCount.Value += 1
endFunction

function addAliasToContainer(ReferenceAlias refAliasToAdd)
    addRefToContainer(( refAliasToAdd.getRef() as ObjectReference))
EndFunction

function GiveItemsToPlayer()
    ItemCount.SetValue(0)
    ContainerAlias.GetRef().RemoveAllItems(Game.GetPlayer())
    Debug.Notification("Item(s) Added.")
EndFunction

ObjectReference Function FindBeaminLocation()

; Used if we are not in an area with preset courier spawn points. Release 3 bouncers

    ObjectReference PlaceTarget = Game.GetPlayer().PlaceAtMe(MarkerType,1)
    ObjectReference[] Bouncer = new ObjectReference[3]
    PlaceTarget.MoveTo(Game.GetPlayer(),5000.0,5000.0,5000.0)
    Bouncer[0] = PlaceTarget.PlaceAtme(BouncerType,1)
    PlaceTarget.MoveTo(Game.GetPlayer(),-5000.0,-5000.0,5000.0)
    Bouncer[1] = PlaceTarget.PlaceAtme(BouncerType,1)
    PlaceTarget.MoveTo(Game.GetPlayer(),-5000.0,5000.0,5000.0) ; Place them at 3
corners of a square.
    Bouncer[2] = PlaceTarget.PlaceAtme(BouncerType,1)
    Wait(5.0) ; Let them fall to the ground. We don't want to bounce our poor
courier around too much.
    ObjectReference sorthold
    if Game.GetPlayer().GetDistance(Bouncer[0]) >
Game.GetPlayer().GetDistance(Bouncer[2])
        SortHold = Bouncer[2]
        Bouncer[2] = Bouncer[0]
        Bouncer[0] = SortHold
    endif
    if Game.GetPlayer().GetDistance(Bouncer[0]) >
Game.GetPlayer().GetDistance(Bouncer[1])
        SortHold = Bouncer[0]
        Bouncer[0] = Bouncer[1]
        Bouncer[1] = SortHold
    endif
    if Game.GetPlayer().GetDistance(Bouncer[1]) >
Game.GetPlayer().GetDistance(Bouncer[2])
        SortHold = Bouncer[2]
        Bouncer[2] = Bouncer[1]
        Bouncer[1] = SortHold
    endif
    ObjectReference Loc1 = Bouncer[0]
    ObjectReference Loc2 = Bouncer[1]
    ObjectReference Loc3 = Bouncer[2]
    ObjectReference BILoc

    if Loc3 && Loc3.GetDistance(Game.GetPlayer()) <= 10000.0 ; Didn't fall through
the world
                BiLoc = Loc3
    endif
    if !BiLoc && Loc2 && Loc2.GetDistance(Game.GetPlayer()) <= 10000.0 ; Ditto
previous comment
                BiLoc = Loc2
    endif
    if !BiLoc
            BiLoc = Loc1
    endif
    return BiLoc
EndFunction

Function StateChange(String Which)
    GoToState(Which)
EndFunction

Bool Updating = False

Event OnUpdate()
    GoToState("Waiting") ; Starts us in the waiting state, in case we somehow got in
the empty state.
EndEvent

State Waiting
    Event OnUpdate() ; Make sure all variables are reset.
    Updating = False
    WhereToGo = None
           Courier.Getref().MoveTo(CourierCellMarker.GetRef())
    Courier.GetRef().Disable()
            If ItemCount.value >= 1.0
        SetStage(100)
            endif
    EndEvent
EndState

State Seeking
   Event OnUpdate() ; First, find a map marker near the player.

    if Courier.GetRef().GetDistance(Game.GetPlayer()) <= 500.0
        WhereToGo = None ; Clear out the property for the next delivery.
        SetStage(200)
    elseif Courier.GetRef().IsEnabled() &&
Courier.GetRef().GetDistance(Game.GetPlayer()) > 30000.0
; Player has eluded the courier. Put it away.
        WhereToGo = None ; If at first you don't succeed...
            Courier.Getref().MoveTo(CourierCellMarker.GetRef())
            Courier.Getref().Disable() ; Go to sleep until the next update cycle.
    endif
    if !Updating && Courier.Getref().IsDisabled()
; Courier is not in the world yet, and we're not already trying to place one.
        Updating = True
        If !WhereToGo ; If we haven't updated yet this delivery
            if ReloadOptionals.IsRunning()
                ReloadOptionals.Stop()
                Wait (1.0)
            Endif
            ReloadOptionals.Start()
; Force a reload of the optional aliases in case we're in a city cell.
            ReloadOptionals.SetStage(0)
            Wait(2.0)
        endif
        ObjectReference Loc1 = Gvar.Near
        ObjectReference Loc2 = Gvar.Mid
        ObjectReference Loc3 = Gvar.Far
        if Loc3
            WhereToGo = Loc3
        endif
        if Loc2 && (!Loc3 || (Loc3 && Game.GetPlayer().GetDistance(Loc3) > 10000.0))
            WhereToGo = Loc2
        endif
        if Loc1 && (!Loc2 || (Loc2 && Game.GetPlayer().GetDistance(Loc2) > 10000.0))
            WhereToGo = Loc1
        Endif

        if !WhereToGo || Game.GetPlayer().GetDistance(WhereToGo) < 900.0
; No location found yet, or it's too close. (Player is in a cell with no viable
markers)
            WhereToGo = FindBeaminLocation()
            if WhereToGo ; Found a spot!
                Courier.Getref().MoveTo(WhereToGo)
                Courier.Getref().Enable()
            endif
        else
            Courier.Getref().MoveTo(WhereToGo)
            Courier.Getref().Enable()
        endif
        Updating = False
    endif

;float DeltaX = Game.GetPlayer().X - Courier.GetRef().X
```

```
;float DeltaY = Game.GetPlayer().Y - Courier.GetRef().Y
;float DeltaZ = Game.GetPlayer().Z - Courier.GetRef().Z

;Debug.Notification("Courier Seeking: "+DeltaX+","+DeltaY+","+DeltaZ)
; Show the player where the courier is for testing.

   EndEvent
EndState



State Delivering
    Event OnUpdate()
            If ItemCount.value == 0
                    SetStage(300)
            endif
    EndEvent
EndState



State Departing
    Event OnUpdate()
; Debug.Notification("Courier Leaving")
            If !Game.GetPlayer().HasLOS(Courier.GetRef())
               Courier.Getref().MoveTo(CourierCellMarker.GetRef())
               Courier.Getref().Disable()
               Updating = False
               SetStage(0)
            endif
    EndEvent
EndState

State Disabled

    Event OnUpdate() ; Courier is out. Come back later.
;        Debug.Notification("Courier Disabled")
    EndEvent

EndState

ObjectReference Property WhereToGo Auto
Activator Property BouncerType Auto
Activator property MarkerType Auto
Quest Property ReloadOptionals  Auto

CourierSpawnPointScript Property Gvar Auto
```

Loader Quest

Since there is no way apparently to make an alias recheck it's conditionals and reload on an "always running" quest, We will need to make a second quest to load the spawn points if the player is in a town.

Note the Quest property ReloadOptionals - That will be a quest that has 3 aliases for the closest, next closest, and third closest spawn points in the nearby areas. ("GVAR" is the same quest, but cast to CourierSpawnPointScript for easy access to the properties.) They are all "Find Matching", "Closest", and "In Loaded Area". Their conditions are:

- Nearest:

  HasKeyword (the keyword we defined above) == 1.0

- NextNearest:

  HasKeyword (the keyword we defined above) == 1.0
  GetIsAliasRef (Nearest) != 1.0

- ThirdNearest:

  HasKeyword (the keyword we defined above) == 1.0
  GetIsAliasRef (Nearest) != 1.0
  GetIsAliasRef (NextNearest) != 1.0

The quest loads these aliases into three properties, Near, Mid, and Far with a script like this:

```
Scriptname CourierSpawnPointScript extends Quest

ObjectReference Property Near Auto
ReferenceAlias Property aNear Auto

ObjectReference Property Mid Auto
ReferenceAlias Property aMid Auto

ObjectReference Property Far Auto
ReferenceAlias Property aFar Auto

Event OnInit()
 Utility.Wait(5.0)
 Near = aNear.GetRef()
 Mid = aMid.GetRef()
 far = aFar.GetRef()
EndEvent
```

Assign ANear, AMid and AFar to the above mentioned aliases.

To use the custom courier, create a property for your script type in the quest and call one of the add functions, depending on whether you're using an alias or an objectreference or something else. You can change the courier type by setting up the Change courier function and you can switch containers if you want for a variety of different behaviors.

Uses

In addition to simple couriers, this could be adapted to spawn encounters with hostile mobs, by simply leaving in the hostile AI, adding a Global WICCPOD {"Place objects on NPC on death"} that kills the delivery scene, and adding a script to the KillableCourier alias that catches the OnDeath event, and places the items from the chest into the "Courier" instead of giving it to the player. (You would also need to make another alias "KillableCourier" that only loads if the WICCPOD property is 1.0, and put code in the Delivery state code that clears the Courier alias once the

KillableCourier finds the player, making the Essential property fall off so it can be killed.

See Also

- Simple custom courier
- Using the Vanilla Courier