Map Object Messaging, Puzzles And Events

This document details how to use the map object and event system, to set up gameplay situations.

Object Classes And The Object Database

All objects require a class ID (type) to tell them what they are, and how to behave. Object classes are set up in the object database. There are already a large number of object classes defined, and more will be added as and when they are needed.

Messaging And Event Setup In WorldEdit

The accompanying document ‘Map Messaging Tools In WorldEdit’ explains how to use WorldEdit to set up map events.

Map Node Messaging

All map 'nodes' (characters, players, triggers, objects etc) send and receive messages which tell them how to react based on certain game circumstances:-

Examples:-
Every frame all nodes are sent as "NM_UPDATE" message.

If Lara (or Kurtis) collides with an object, the object is sent an "NM_PLAYERCOLLIDE" message, Lara is sent an "NM_OBJCOLLIDE" message.

When a triggers activation conditions are met, the trigger is sent an "NM_ACTORCOLLIDE" message. This in turn causes the trigger to dispatch the messages to the targets set up in the triggers "PARAM LIST"

Map Event Binding

Map messaging is a simple and flexible method of controlling basic node behavior. Event binding provides a mechanism for complex behaviors and 'Chain Reaction' style events.

Event binding is a system for dispatching multiple messages to multiple targets (known as an event group) when another event (message) occurs. This enables any node to act like a trigger (the triggering system uses the event binding mechanism behind the scenes).

The following examples were used for the initial tests of the system:-

Example 1 – Lara touches valuable vase, causing large rocks to fall on her.

Sequence of events:-

Lara touches vase – game sends "NM_PLAYERCOLLIDE" to vase.

Vase has "TriggerHeavyRocksOnVase" event group bound to "NM_PLAYERCOLLIDE" message.

"TriggerHeavyRocksOnVase" sends messages to heavy rocks causing them to fall.

Breakdown of components for example 1:-

Nb:- Don't worry about the references to object flags- these will be explained fully later in this document.

Vase – Set up in the database with the flag OF_PLAYERCOLLIDE set – this means it will check for collision with Lara.

Heavy Rocks – Set up in the database with flags OF_GRAVITY, OF_PLAYERCOLLIDE and OF_STATIC.

This means that they will fall with gravity and collide with Lara. The OF_STATIC flag overrides everything and stops the object from updating.

Objects placed in Maya and tagged using the mesh attribute buddy.

"TriggerHeavyRocksOnVase" – Event group – set up in World Edit from Window → event editor.
Then "Create new group" and enter name of group.
Then select group and "Edit group" – This brings up the "Messaging Editor" (as used for the triggers). The maessages are set up as "NM_CLEAROBJECTFLAGS", targets as Big Rock 1, 2 and 3 parameter is the Static flag – use the flag calculator to work out the value to use.

To set up the event binding, select the vase object in WorldEdit, go into the attribute editor and select "Edit event bindings". This will bring up the event binding editor. Select the message you wish to 'bind' an event group (in this case "NM_PLAYERCOLLIDE"), and the event group you wish to use ("TriggerHeavyRocksOnVase") and click "Create New Entry".

When this binding is setup, when Lara collides with the vase object (vase receives "NM_PLAYERCOLLIDE), the messages in "TriggerHeavyRocksOnVase" will be sent, causing the rocks to fall (Hopefully on Lara).

Breakdown Of Game Side Events for Example 1:-

Frame 1. – Rocks receive NM_UPDATE, but do nothing as their OF_STATIC flags are sent.
Vase receives NM_UPDATE message.
Vase checks to see if it has collided with Lara.
Vase has collided with Lara, so sends itself "NM_PLAYERCOLLIDE"
Vase receives "NM_PLAYERCOLLIDE" – When a node receives a message it checks to see if it has anything bound to the message. In this case it will find that "TriggerHeavyRocksOnVase" is bound to "NM_PLAYERCOLLIDE" and dispatches the messages in "TriggerHeavyRocksOnVase"
Heavy Rock 1, 2 and 3 all receive NM_CLEAROBJECTFLAGS messages with a parameter telling them to clear OF_STATIC.

Frame 2. – Rocks receive "NM_UPDATE" – Now that "OF_STATIC" is cleared, they begin to fall as "OF_GRAVITY" is set.

Event Binding Example 2:-

Give Lara a medipack pickup when she searches a drawer.

Components:
Medipack Pickup – Placed in map with the 'ghost' box ticked in the attribute editor (Worldedit). This means the object is not displayed or updated.

Drawer – Tagged as a searchable drawer type from the database. Has "GiveMedipack" event group bound to "NM_SEARCHFINISH".

"GiveMedipack" event group – Sends "NM_GIVEITEM" to Lara with the parameter set to the hash value of the Medipack pickup node name.

***Sequence of Events* :-**

When Lara opens a searchable object (the drawer) she goes through a full open → search → close sequence. When each of these actions finish the object receives an "NM_OPENFINISH / NM_SEARCHFINISH / NM_CLOSEFINISH" message. As the drawer has "GiveMedipack" bound to NM_SEARCHFINISH, when Lara finishes searching the drawer , "GiveMediPack" events will be dispatched and Lara will receive a Medipack.

More Notes on 'Giving Items'

Once an item to 'give' has been setup in the map along with the event group to "Give" that item, it can be used in multiple places in the level and on multiple object types. Just bind the event group to the appropriate message.

Puzzles And Puzzle Events:-

Simple puzzles are setup in the database as normal objects with a 'puzzle count' and a set of 'auto puzzle items' (up to four). Each time one of the 'auto puzzle items' is used the 'puzzle count' is decremented and "NM_USEPUZZLE A – D" is sent to the object. When the puzzle count hits 0 "NM_PUZZLECLEAR" is sent to the object.

By binding event groups to the NM_PUZZLECLEAR message, reactions to completing puzzles can be setup. As puzzles usually require an animation from the player they are setup as doors, and use door triggers.

Puzzle Example 1:-

Padlocked wardrobe. The padlock can be opened with the crowbar (As used in Paris 1)

Components:

Wardrobe – Tagged with the appropriate wardrobe class from the database.

Wardrobe Door Trigger – Setup as a manual door trigger. Trigger attributes edited so active is not ticked, leaving the trigger inactive.

Padlock – Tagged with the 'Lara crowbar padlock' class from the database. This has a puzzle count of 1, is set as locked (OF_LOCKED) and has a single 'auto puzzle item' – the crowbar. Padlock has an event group "ActivateWardrobeTrigger" bound to event "NM_PUZZLECLEAR".

Padlock Trigger – A standard manual door trigger.

Event Group "ActivateWardroveTrigger" – Setup to send "NM_ACTIVATE" to the wardrobe door trigger and "NM_DEACTIVATE" to the padlock trigger.

Game Side Events for Puzzle Example 1 :-

Lara steps into padlock trigger and presses 'action' to open padlock. Padlock is 'locked' and has a puzzle count set on it, so the game invokes the inventory. If Lara uses the correct puzzle item (in this case the crowbar) the padlock puzzle count hits 0 so the padlock receives "NM_PUZZLECLEAR". As the padlock has the "ActivateWardrobeTrigger" event group bound to "NM_PUZZLECLEAR", the events in "ActivateWardrobeTrigger" are dispatched:-

"NM_ACTIVATE" is sent to the wardrobe trigger making it active, thus making the wardrobe openable.

"NM_DEACTIVATE" is sent to the padlock trigger making it inactive – it's no longer needed and deactivating it stops it interfering with the (now active) wardrobe trigger.

Puzzle example 2:-

Locked door and swipe card reader

This puzzle can be setup as a simple 'one shot' puzzle:-
Lara swipes card, door opens and remains open.

The puzzle can also be setup as more complex puzzle:-
Lara swipes card, door opens and then closes after Lara and requires swipe card to open it again.

Components for simple case:
Door – Tagged from database as locked (OF_LOCKED) with no player open anims (Lara cannot open it by hand)

Door Trigger – Set up as manual open and manual close

Swipe Card Reader – Tagged in the database with a puzzle count of 1, and a puzzle item of the swipe card. Also tagged with a door / switch class to give it the Lara anims required. Has the "OpenSwipeCardDoor" event group bound to "NM_PUZZLECLEAR"

Swipe Card Trigger – Setup as a switch trigger

Event group "OpenSwipeCardDoor" – Sends NM_UNLOCK to the door
Sends NM_OPEN to the door.

Sequence of events for puzzle example 2 (Simple case)

Lara uses swipe card on swipe card reader trigger. Swipe card reader receives "NM_PUZZLECLEAR" which has "OpenSwipeCardDoor" bound to it.
"OpenSwipeCardDoor" events are dispatched:-

NM_UNLOCK – Unlocks door
NM_OPEN – Opens the door

Component alterations for the more complex case:-

Door Trigger – Set up as manual open and auto close.

Additional event groups – "ResetSwipeCardPuzzle"
Sends NM_SETPUZZLECOUNT (Parameter of 1) to the swipe card reader.
Sends NM_LOCK to the door.

Extra Bindings – Door has "ResetSwipeCardPuzzle" bound to NM_CLOSEFINISH (NM_CLOSEFINISH is sent to door when 'close' anim finishes)

Sequence of events for complex case (Continuing from simple case above)
When Lara leaves the door trigger the door closes (this is done using event bindings but they are automatically setup by Worldedit – clicking ‘edit event bindings’ on the door will show them). When the door finishes closing it sends NM_CLOSEFINISH to itself. “ResetSwipeCardPuzzle” is bound to NM_CLOSEFINISH so the puzzle resets itself.

Breakdown Of User / Bindable Messages:-

NM_INIT
Received by a node on level startup.

NM_DAMAGE - NM_DAMAGESTAGE
Received by a node when it takes damage. NM_DAMAGESTAGE is sent when an object changes its mesh to a 'more broken' one.

NM_DESTROY
Received by a node when it is destroyed.

NM_UPDATE
Received by a node every game loop

NM_CHARENTRY – NM_CHAREXIT
Received by a trigger node (and it's targets) when a character / player enters / exits the trigger box. Only if trigger has TF_CHARTRACK set ('Track character' tick box in Worldedit).

NM_FXTRIG
Sent to / received by 'global_effectsmanager' to trigger effects. The parameter is the hash value of the name of the effect template to trigger.

NM_LADDER – NM_WINDOW
Sent to Lara / Kurtis by a trigger to indicate ladders / windows.

NM_OPEN
Send it to an object, watch it open (if it can)

NM_CAMERAENTRY – NM_CAMERAEXIT
As NM_CHARENTRY / EXIT but for the camera.

NM_OBJCOLLIDE
Sent to a node when it collides with an object.

NM_PICKUP
Sent to Lara / Kurtis when they receive a pickup.

NM_REMOVE
Send it to a node to remove it from the world.

NM_SETOBJECTFLAGS – NM_CLEAROBJECTFLAGS
Set / clear object flags. Parameter is flags to set / clear – See flag calculator in Worldedit for calculating parameter.

NM_PLAYERCOLLIDE
Sent to an object when Lara / Kurtis collides with it.

NM_BGCOLLIDE
Sent to an object when it collides with the background.

NM_STARTTIMER
Starts a countdown timer on a node – Parameter is value to countdown from.

NM_STARTTIMERRANGE
Starts the timer with a random value within a range. Parameter is the packed 2 digit range. See 'messaging editor' for information on creating packed parameters in Worldedit.

NM_TIMEREVENT
Sent to a node when its timer hits 0.

NM_PIPESWING – NM_WALLCLIMB
Like NM_LADDER / NM_WINDOW.

NM_DISPATCHEVENT
Used by triggers to dispatch an event group. Parameter is the hash value of the event group name.

NM_CLOSE
Send it to an object, watch it close

NM_ANIMFINISH
Sent to an object when it's current animation finishes.

NM_PUZZLECLEAR
Sent to an object when it's puzzle count hits 0.

NM_USEPUZZLE A,B,C,D
Sent to an object when auto puzzle item a,b,c, or d is used on it.

NM_UNLOCK
Clears an objects lock (OF_LOCKED). Same as sending NM_CLEAROBJECTFLAGS with OF_LOCK as the parameter.

NM_UPGRADE_UPPERBODY / LOWERBODY / GRIP / JUMP
Sent to Lara / Kurtis to do body upgrades. Parameter is the upgrade level.

NM_SEARCHFINISH
Sent to an object when search anim finishes playing.

NM_GIVEITEM
Send to Lara / Kutis to give a pickup item. Parameter is hash value of the pickup node name.

NM_DEACTIVATE
Set node as inactive

NM_OPENFINISH – NM_CLOSEFINISH
Sent to an object when it's open / close anim finishes.

NM_VOICEPROMPT
Presumably something to do with voice prompt stuff.

NM_STARTWATERRAISE / NM_STARTWATERLOWER
Starts the ***surface*** of a water volume raising / lowering – user parameter is the velocity

NM_STOPWATER
Stops the surface of a water volume (when raising / lowering)

NM_SHOWWATER
Shows a hidden water volume

NM_HIDEWATER
Hides a visible water volume

NM_SETPUZZLECOUNT
Sets an object's puzzle count

NM_FLIPROOM_RESET
Set a flippable room back to its original state

NM_FLIPROOM0 – NM_FLIPROOM3
Set a flippable room to one of its 4 flipmaps

NM_PLAYANIM
Play an animation on an animated object. User parameter is the hash value of the animation to play.

NM_SETROTX
NM_SETROTY
NM_SETROTZ
Set an objects x,y or z angles. User parameter is the angle to set, in degrees.

NM_ROCK
Send it to an item to make it rock

NM_LOCK
Sets an object's lock flag.

NM_FLIPROOM_RESET
NM_FLIPROOM0
NM_FLIPROOM1
NM_FLIPROOM2
NM_FLIPROOM3

Flip room messages. Can be sent to either 'global__zonemanager', with the hash value of the room aggregate name as the parameter, or (and this is easier) directly to the room itself (not yet implemented!).
NM_FLIPROOM_RESET will reset the room to its initial state, NM_FLIPROOM0-3 flip to the relevant flipmap (if set up).

NM_USER0
NM_USER1
NM_USER2
NM_USER3
NM_USER4
NM_USER5
NM_USER6
NM_USER7

User defined messages that have no specific effect – you can use them to indicate anything you like….

NM_DAMAGESTAGE

Gets sent to an object when it changes to one of its broken mesh stages – bind groups that trigger effects to it (dust etc).

NM_LIFTPLATFORM

Use on triggers that are associated with lift platforms. Send this message to Lara/Kurtis.

NM_TIMEDACTIVATE

Sends NM_ACTIVATE to a node, sets up a timer on the node for the number of ticks specified in user parameter (60 ticks = 1 second), then deactivates the node when the timer reaches 0. Useful for camera cut triggers…

NM_PLAYANIM_SLOT0
NM_PLAYANIM_SLOT1
NM_PLAYANIM_SLOT2
NM_PLAYANIM_SLOT3

Plays one of the four predefined anim slots (set up in the database for object classes) on an object. Easier to use (if set up) than NM_PLAYANIM

NM_DEFLECT
Deflects a character / noderigger by the amount specified in Param. Use 'create packed parameter', and enter the lower and upper deflection amouts.

NM_ENDLEVEL
Send to 'global__zonemanager' to end the level. No need to worry about telling it which level to go to next – the game knows……. (probably doesn't work yet, mind..)

NM_CUEMUSIC
Start a lovely tune. Use the hash calculator, it has a built in list of music id's…

NM_ANIMFINISH_SLOT0
NM_ANIMFINISH_SLOT1
NM_ANIMFINISH_SLOT2
NM_ANIMFINISH_SLOT3
Sent to an object when the animation playing from the relevant slot (NM_PLAYANIM_SLOT0-3) finishes. Bind to it to create chained anim events (for instance the crane in the Strahov – ask Dave…)

NM_TIMEDREMOVE
Like NM_TIMEDACTIVATE, but removes the node totally from the world.

NM_ACTIVATEONTIMER
Activates the node AFTER the timer reaches 0, leaves the node activated.

NM_CUTSCENE

Play a cutscene. Hash calculator has a list of cutscene ID's.

NM_HEADLOOK

Send to an object to force Lara to look at it

NM_LEDGESHIMMY

Indicates ledge shimmy – Phil knows more…

NM_STARTAUDIOEVENT
NM_STOPAUDIOEVENT
NM_AUDIOEVENTFADE

Something to do with Nigel…

NM_LOCKPLAYER
NM_UNLOCKPLAYER

Lock and unlock player controls

NM_HARMPLAYER

Harm Lara. Left hand (2 value) packed parameter is %damage from maximum health x100 (so a value of 100 does 1% damage, 10000 will kill Lara). Right value is method of harming.

NM_ZONESWITCH

Sent to global__zonemanager to handle zone switching. Two value packed parameter are zone ID's to switch between.

NM_ZONESWITCH_DOOR

As above, but bound to NM_OPEN on a door. Will fade and zone load as door is opening…

NM_DISABLETRIGGER
NM_ENABLETRIGGER

Disable / enable trigger. When an active trigger is disabled, it is still scanned, but never triggered. Action icons will still be displayed, but grayed out…

NM_HIDEOBJECT

Hide an object (make invisible)

NM_SHOWOBJECT

Unhide an object

NM_CUTSCENE_END

Sent back to the trigger that triggered a cutscene, when the cutscene finishes.

NM_SETXVEL
NM_SETYVEL
NM_SETZVEL

Set X Y or Z velocities on an object

NM_SETVELDECAY

Set the velocity decay on an object, as a percentage of the current velocity per frame (so a param of 100 gives no decay, a param of 0 gives instant decay). Only used when the OF_SCALEVELOCITY flag is set on the object (either through the database, or with a NM_SETOBJECTFLAGS message)

NM_LIFTSIGNAL

Send to Lara to indicate that she is within a lift trigger, but don't update her position. Used to indicate lifting areas that she can freeclimb / hang / money swing on.

NM_INCREMENT_UVAR

Add 1 to a user variable on the node. All nodes have 16 user variables, a param of 0-15 is used to select the one to increment. NB: Be very careful with this message is the node is running a script – the scripts use some of the user variables for, er, variables…. Check first if in doubt…

NM_BLAST

Send it to an object to create a blast wave of 'param' radius around that object. Will affect Lara, Enemies and any breakable or rockable objects within the radius. This is only a logical operation, and will not create any visual effects. If you want those, ask Dan…

NM_PLAYANIMSLOT_UVAR

Play the animation slot referenced by userVariable[param]. As mentioned before, there are 16 user variables on each node. The number in 'param' indicates which user variable to use. The number contained in that user variable indicates which animation slot to play. This means that multiple anim sequences (eg the crane in Prague2) can be created without multiply linked event groups. Instead, try something like this, to sequence and play 8 anims:-
[assuming using userVariable 5 – all user variables contain 0 on init]

initial action:-
NM_PLAYANIMSLOT_UVAR , 5
NM_INCREMENT_UVAR , 5

Bind to NM_ANIMFINISH:-
NM_PLAYANIMSLOT_UVAR , 5
NM_INCREMENT_UVAR , 5

Bind to NM_ANIMFINISH_SLOT7 (so it's played 8 slots, slot 0 to slot7 inclusive)
NM_CLEARUSERVAR , 5

Which will cause it to wrap round to slot 0 again – alternatively to stop the sequence use
NM_INVALIDATE_USERVAR , 5

NM_PLAYANIMSLOT

Play animslot reference by param. So, NM_PLAYANIMSLOT , 0 will play anim slot 0, NM_PLAYANIMSLOT , 5 will play anim slot 5.
This supercedes NM_PLAYANIMSLOT0 – NM_PLAYANIMSLOT3. These will still work (for backwards compatability), but this method should be used in future.

NM_ANIMFINISH_SLOT4
NM_ANIMFINISH_SLOT5
NM_ANIMFINISH_SLOT6
NM_ANIMFINISH_SLOT7
NM_ANIMFINISH_SLOT8
NM_ANIMFINISH_SLOT9
NM_ANIMFINISH_SLOT10
NM_ANIMFINISH_SLOT11

Anim finish messages for the extra 8 anim slots….

NM_CLEARUSERVAR
Set the user var referenced by Param to 0. So, NM_CLEARUSERVAR , 3 will set user var 3 to 0. Remember, all user vars are numbered from 0 – 15. Be careful with this message – see notes above about scripts….

NM_INVALIDATE_USERVAR
Sets user var referenced by Param to –1, animations trying to reference it will not be played.

The following messages are not yet implemented, but will be very soon…

NM_BGCOLLIDE_FLOOR
NM_BGCOLLIDE_CEILING
NM_BGCOLLIDE_WALL
Sent along with NM_BGCOLLIDE to an object to indicate which part of the background it has collided with.

<u>More Event Binding Examples:-</u>

Create event groups to trigger effects (dust, debris etc) then bind them to damage events on breakable objects:-

Event group "CreateSmallDust" – Bind to NM_DAMAGE
"CreateSmallDebris" – Bind to NM_DAMAGE
"CreateLargeDust" – Bind to NM_DAMAGESTAGE

"CreateLargeDebris" – Bind to NM_DAMAGESTAGE

Create an event group to 'rock' the current object (use 'this' as the target – see notes on messaging editor) and bind it to NM_PLAYERCOLLIDE so when Lara knocks into it it rocks.

Create an event group to enable a zone trigger (send NM_ACTIVATE to trigger) and bind it to the NM_OPEN event of a door sat over the trigger. Do the reverese as well (event group sends NM_DEACTIVATE to the trigger bound to NM_CLOSE) and the door will enable the zone trigger as it starts to open and disable the zone trigger as it starts to close.