

Bethesda Tutorial Radiant Quests

 creationkit.com/index.php

Contents

Overview

This chapter will explain the more flexible systems we now have for creating dynamic quest content in various degrees. We collectively refer to these variable systems as "Radiant" as a kind of loose descriptor.

The reader will learn:

- How to fill an alias with conditions
- How to dynamically fill an alias based on other data

The Spice of Life

The content we've written so far, well, it's not so great. Overly simple, basic fetch quest kind of things, with an unresolved love plotline to boot. On top of that, it's always the same. We won't address the first set of problems here (that's left as an exercise), but we can at least add some variety by changing things up.

This is where the power of aliases as overlaid roles really becomes apparent -- the more of the quest functionality that is placed on the alias instead of base actors, the more we can take that functionality and overlay it on any actor. This is a very powerful idea, and gives us the ability to tell stories in Skyrim in a much more procedural manner.

Imagine how we could potentially roll the dice with this quest:

- **Dungeon Change:** Pick the boss of a random dungeon and place the amulet on them. Since the selection system has a bias for places the player has never visited, this is likely to send him to someplace new!
- **Different Victim:** We could pick anyone in the world to have lost their amulet -- or only Dark Elves who live alone, or only someone from Eastmarch, etc. We can be as specific or general as we want here, limited really just by how many different voices we want to record.

Et cetera, et cetera, et cetera.

For this example, we'll only be changing the dungeon location, but it should be enough to give you the idea.

Conditional Aliases

An important notion when constructing a radiant quest is that of the "seed alias". Remember, it's not just the thief we'll be finding -- we also need to know what map marker to add to the player's map, which means we have to know the dungeon itself as well. So there are three aliases we'll need to worry about here, but they all feed off that initial selection of the thief. For technical reasons, however, we usually want to start with the location (telling the game to pick a location that has a boss, then assigning the thief as that boss). In this case, the location becomes our "seed alias". When tracking down bugs in radiant content, knowing which alias is the seed is very important.

Go over to the alias tab, and right-click to make a new Location alias and call it "ThiefLocation". Location aliases are somewhat abstract bits of data that represent places and the collection of "stuff" within them. A location might know, for instance, that it contains certain types of creatures, or that it is part of a city, or that it is in a particular hold, or

that it has a front door, or that it has an actor designated as its "boss". We can test this data with conditional statements to find a Location that matches our needs.

The first condition we'll add is "LocationHasRefType;" choose "Boss" from the pull-down menu. This condition will be true for any location that contains an actor who has been assigned a "Boss" keyword.

The second condition we want to add is "LocationHasKeyword," which will test the data on the location itself (as opposed to the actors it contains). The keyword we want in this case is "LocTypeBanditCamp".



If you're interested in how to define this data, you should investigate [Locations](#) and [Keywords](#).

As you might expect, these two conditions will filter out all locations *except* for bandit camps that have bosses. The game will then pick randomly from the set of valid locations.

Finally, click the "Reserves Location" reference at the top of the window -- that will just prevent any other radiant content from using this location while our quest is running. Your new location alias should look like this:

Location Alias

Alias NameThiefLocation

☒Reserves Location☐Optional☐Stores Text

Force Into Alias When FilledNONE

☐Allow Reuse in Quest☐Allow Reserved☐Allow Cleared

Fill Type

☐Specific LocationNONE

☐Reference Alias LocationNONE

☐External Alias LocationNONE

☒Find Matching Location

☐From Event

KeywordNONE

AliasNONE

Event DataNONE

Target	Function Name	Function Info	Comp	Value		
S	LocationHasRefType	Location Ref Type: 'Boss'	==	1.00	AND	
S	LocationHasKeyword	Keyword: 'LocTypeBanditCamp'	==	1.00	AND	

<<>>

New

OK

Cancel

Close this window. **This part is important:** with the ThiefLocation alias highlighted, use the left arrow key to move it so that it's above the Thief alias. Because the aliases will get filled from the top down, order becomes important, and since the Thief is going to be based on this alias, it needs to come after it.

Dependent Aliases

Open up the Thief alias. Previously this was pointing to a Specific Reference, but we're going to change that to be keyed off the location that we found. Click the radio button for "Location Alias Reference;" this will let us pick any of the references that a location potentially contains. Select "ThiefLocation" from the first pull-down menu, and "Boss" from the Ref Type pull-down.

- ◆ Note that if we had not checked that our Location actually contained a Boss, this alias could fail to fill. If any alias fails to fill, the quest will never start, so look here for problems if your quest is never starting. You can flag an alias as "Optional" if you need to leave it empty for some reason. Here, ThiefLocation can be checked as "Optional" so that code conditional on it being empty can run at Stage 0 in an attempt to refill it.
-

Now we just need an alias for the map marker. Make one more Reference Alias, and call it "LocationMarker". This will also be conditional -- it's two conditions should be:

- GetInCurrentLocAlias: ThiefLocation
- GetIsId: MapMarker

Finally, we just need to change our script for stage 10 to add this map marker instead of the hard-coded reference we put before. So edit the stage 10 script to now read:

```
SetObjectiveDisplayed(10)
```

```
Alias_Thief.GetReference().Enable()
```

```
Alias_LocationMarker.GetReference().AddToMap()
```

The Spice Must Flow

Congratulations, you've now added a radiant element to your quest. You should do a pass to remove any specific mentions of the now deprecated Reachwind Eyrie-- in light of the fact that it isn't one of the 31 objects tagged by LocTypeBanditCamp, less with Bosses, (Bendu probably just doesn't need to mention it by name, for instance). Otherwise, the quest functions normally!

Don't mention the location

If you followed the [Basic Quest Objectives tutorial](#) to the letter, you will notice that none of the Objectives actually do make reference to Reachwind Eyrie, so there's nothing to change there, but what if we wanted the Quest Objective to tell the player where the thief is, when even we don't know? This is actually very simple to achieve now that we've set everything up. First, go back to your ThiefLocation alias and check the 'Stores Text' checkbox. Then, go back to your Quest Objectives tab, and change the Display Text for Objective Index 10 from:-

"Kill the thief" to:-

"Kill the thief at <Alias=ThiefLocation>".

Slotting aliases into text is just this easy. We don't need to check if the alias is filled, because (assuming we're not using an alias with the 'Optional' checkbox checked), the quest simply won't start if it isn't. For more advanced information on what you can do with this, see the page on [Text Replacement](#).

The time you want to be more careful not to mention a location (or any other alias that will be filled at runtime) by name is anything that has to be voice-acted. Even then, you can, if you want, voice-act several lines to give a rough location, e.g. narrowing it down to a specific hold or a specific city, but that is left as an exercise for the reader.

Whoops

One problem we still have is that we left the Thief script on the thief's base object, instead of having it on the appropriate alias. Now you see how much easier things are if you make your quest data alias-centric from the start! Do you know how to clean this up? By now you should, and don't forget the amulet either!

Almost there...

Only one more stop before you have a basic grasp of all the Creation Kit quest systems! Onwards to the STORY MANAGER...