

idEncounterManager

An entity that scripts the encounters. This is what is used to spawn in demons.

Usage

```
entity {
  entityDef example_encounter_manager {
    inherit = "encounter/manager";
    class = "idEncounterManager";
    expandInheritance = false;
    poolCount = 0;
    poolGranularity = 2;
    networkReplicated = false;
    disableAIPooling = false;
    edit = {
      enableAIHighlightOnFinish = true;
      disabledAITypeForHighlight = "AI_MONSTER_SPECTRE AI_MONSTER_BUFF_POD AI_MONSTER_TENTACLE";
      playerMetricDef = "encounter/player_metrics";
      chargeCombatGrouping = "encounter/combat_role/charge_command";
      aiTypeDefAssignments = "actorpopulation/default/default_no_bosses";
      spawnPosition = {
        x = 1;
        y = 1;
        z = 1;
      }
      combatRatingScale = "COMBAT_RATING_SCALE_IGNORE"; // Progress reward for Demonic Corruption
      Meter (set to none)
      ignoreTelemetry = true;
      encounterComponent = {
        entityEvents = {
          hum = 1;
          item[ 0 ] = {
            entity = "example_encounter_manager";
            events = {
              hum = 0; // eventCalls go here
            }
          }
        }
      }
    }
  }
}
```

```

    }
    }
    }
    commitTriggers = {
        num = 0;
    }
    exitTriggers = {
        num = 0;
    }
    userFlagTriggers = {
        num = 0;
    }
}
}

```

In the **aiTypeDefAssignments** line, the **"actorpopulation/default/default_no_bosses"** define what demons can spawn in the encounter.

DLC1 **aiTypeDefAssignments** would look like **"actorpopulation/default/dlc1"** and DLC2 would look like **"actorpopulation/default/dlc2"**

commitTriggers show what volume trigger would activate the encounter.

exitTriggers show what volume trigger would turn off the encounter.

userFlagTriggers show what Flag Triggers are enabled for the encounter.

Example

Encounter managers can be quite long, but this example displays a very simple encounter.

Here's a rundown of what it does chronologically:

1. Sets music to be light
2. Spawns a Hell Knight
3. Waits for 2 seconds
4. Spawns a Cacodemon
5. Continuously maintains 4 Imps at a time
6. Makes the AI aware of the player
7. Waits for 0 AI with the "main_combat" string to remain (which is the Hell Knight and Cacodemon)
8. Sets music to be heavy
9. Spawns a Baron of Hell

10. Waits for 0 AI with the "main_combat" string to remain (which is the Baron)
11. Stops all AI from respawning (which are the Imps)
12. Forces all remaining AI to charge towards you
13. Waits for 0 AI to remain
14. Sets music to be ambient

```
entity {  
  entityDef example_encounter_manager {  
    inherit = "encounter/manager";  
    class = "idEncounterManager";  
    expandInheritance = false;  
    poolCount = 0;  
    poolGranularity = 2;  
    networkReplicated = false;  
    disableAIPooling = false;  
    edit = {  
      enableAIHighlightOnFinish = true;  
      disabledAITypeForHighlight = "AI_MONSTER_SPECTRE AI_MONSTER_BUFF_POD AI_MONSTER_TENTACLE";  
      playerMetricDef = "encounter/player_metrics";  
      chargeCombatGrouping = "encounter/combat_role/charge_command";  
      aiTypeDefAssignments = "actorpopulation/default/default_no_bosses";  
      spawnPosition = {  
        x = 1;  
        y = 1;  
        z = 1;  
      }  
      combatRatingScale = "COMBAT_RATING_SCALE_IGNORE";  
      ignoreTelemetry = true;  
      encounterComponent = {  
        entityEvents = {  
          hum = 1;  
          item[ 0 ] = {  
            entity = "example_encounter_manager";  
            events = {  
              hum = 14;  
              item[ 0 ] = {  
                eventCall = {  
                  eventDef = "setMusicState";  
                  args = {  
                    hum = 3;  
                    item[ 0 ] = {  
                      entity = "musicentity_1";
```

```

    {}
    {}item[1] = {
    {}decl = {
    {}soundstate = "music_ghost_states/main_light";
    {}
    {}
    {}item[2] = {
    {}string = "Light Music";
    {}
    {}
    {}
    {}
    {}
    {}item[1] = {
    {}eventCall = {
    {}eventDef = "spawnSingleAI";
    {}args = {
    {}hum = 3;
    {}item[0] = {
    {}eEncounterSpawnType_t = "ENCOUNTER_SPAWN_HELL_KNIGHT";
    {}
    {}item[1] = {
    {}entity = "example_spawn_1";
    {}
    {}item[2] = {
    {}string = "main_combat";
    {}
    {}
    {}
    {}
    {}item[2] = {
    {}eventCall = {
    {}eventDef = "wait";
    {}args = {
    {}hum = 2;
    {}item[0] = {
    {}float = 2;
    {}
    {}item[1] = {
    {}bool = false;
    {}
    {}

```

```

    }
}

    item[3] = {
        eventCall = {
            eventDef = "spawnSingleAI";
            args = {
                num = 3;
                item[0] = {
                    eEncounterSpawnType_t = "ENCOUNTER_SPAWN_CACODEMON";
                }
                item[1] = {
                    entity = "example_spawn_2";
                }
                item[2] = {
                    string = "main_combat";
                }
            }
        }
    }

    item[4] = {
        eventCall = {
            eventDef = "maintainAICount";
            args = {
                num = 8;
                item[0] = {
                    eEncounterSpawnType_t = "ENCOUNTER_SPAWN_IMP";
                }
                item[1] = {
                    int = 4;
                }
                item[2] = {
                    int = -1;
                }
                item[3] = {
                    float = 2;
                }
                item[4] = {
                    int = 3;
                }
                item[5] = {
                    entity = "example_spawn_group";
                }
            }
        }
    }
}

```

```

    {}
    {}item[ 6] = {
    {}string = "fodder";
    {}
    {}item[ 7] = {
    {}float = 6;
    {}
    {}
    {}
    {}
    {}item[ 5] = {
    {}eventCall = {
    {}eventDef = "makeAIAwareOfPlayer";
    {}args = {
    {}hum = 4;
    {}item[ 0] = {
    {}bool = true;
    {}
    {}item[ 1] = {
    {}bool = true;
    {}
    {}item[ 2] = {
    {}string = "";
    {}
    {}item[ 3] = {
    {}bool = false;
    {}
    {}
    {}
    {}
    {}
    {}item[ 6] = {
    {}eventCall = {
    {}eventDef = "waitAIRemaining";
    {}args = {
    {}hum = 4;
    {}item[ 0] = {
    {}eEncounterSpawnType_t = "ENCOUNTER_SPAWN_ANY";
    {}
    {}item[ 1] = {
    {}int = 0;
    {}
    {}

```

```

item[ 2] = {
    string = "main_combat";
}

item[ 3] = {
    bool = false;
}

}

}

}

}

item[ 7] = {
    eventCall = {
        eventDef = "setMusicState";
        args = {
            hum = 3;
            item[ 0] = {
                entity = "musicentity_1";
            }
            item[ 1] = {
                decl = {
                    soundstate = "music_ghost_states/main_heavy";
                }
            }
            item[ 2] = {
                string = "Heavy Music";
            }
        }
    }
}

item[ 8] = {
    eventCall = {
        eventDef = "spawnSingleAI";
        args = {
            hum = 3;
            item[ 0] = {
                eEncounterSpawnType_t = "ENCOUNTER_SPAWN_BARON";
            }
            item[ 1] = {
                entity = "example_spawn_3";
            }
            item[ 2] = {
                string = "main_combat";
            }
        }
    }
}

```

```

    }
    }
    }
    }
    item[ 9] = {
        eventCall = {
            eventDef = "waitAIRemaining";
            args = {
                hum = 4;
                item[ 0] = {
                    eEncounterSpawnType_t = "ENCOUNTER_SPAWN_ANY";
                }
                item[ 1] = {
                    int = 0;
                }
                item[ 2] = {
                    string = "main_combat";
                }
                item[ 3] = {
                    bool = false;
                }
            }
        }
    }
    item[ 10] = {
        eventCall = {
            eventDef = "stopMaintainingAICount";
            args = {
                hum = 2;
                item[ 0] = {
                    eEncounterSpawnType_t = "ENCOUNTER_SPAWN_ANY";
                }
                item[ 1] = {
                    string = "";
                }
            }
        }
    }
    item[ 11] = {
        eventCall = {
            eventDef = "forceChargeOnAllAI";

```

```

    {}args = {
    {}hum = 0;
    {}
    {}
    {}
    {}item[ 12] = {
    {}eventCall = {
    {}eventDef = "waitAIRemaining";
    {}args = {
    {}hum = 4;
    {}item[ 0] = {
    {}eEncounterSpawnType_t = "ENCOUNTER_SPAWN_ANY";
    {}
    {}item[ 1] = {
    {}int = 0;
    {}
    {}item[ 2] = {
    {}string = "";
    {}
    {}item[ 3] = {
    {}pool = false;
    {}
    {}
    {}
    {}
    {}
    {}item[ 13] = {
    {}eventCall = {
    {}eventDef = "setMusicState";
    {}args = {
    {}hum = 3;
    {}item[ 0] = {
    {}entity = "musicentity_1";
    {}
    {}item[ 1] = {
    {}decl = {
    {}soundstate = "music_ghost_states/main_ambient";
    {}
    {}
    {}item[ 2] = {
    {}string = "Ambient Music";
    {}
    {}

```

