

DEATH MARKERS

Minimum steps to add Death Markers to your own ruleset.

COMBAT TRACKER

ct_entry

Add an idelete button and set as invisible

Set it visible when damage is greater than or equal to health – use a script like this on damage field

```
function onInit()
    super.onInit();
    onValueChanged();
end

function onValueChanged()
    super.onValueChanged();

    if getValue() > 0 and getValue() >= window.health.getValue() then
        window.idelete.setVisible(true);
    else
        window.idelete.setVisible(false);
    end
end

end
```

set onPress or onClickRelease on idelete to run a CoreRPG function

ActorHealthManager.isDyingOrDead

```
function onClickRelease(button, x, y)
    ActorHealthManager.isDyingOrDead(window.getDatabaseNode());
end
```

Update the delete() function (ct_entry.lua)

```
function delete()
    CombatManager.deleteCombatant(getDatabaseNode());
end
```

CoreRPG and DnD expects hptotal and wounds, your ruleset may use different fieldnames

GAMESYSTEM

Add the following to onInit

```
function onInit()
    registerStandardDeathMarkersSotDL();
end
```

Add a new function

```
function registerStandardDeathMarkersSotDL()
    ImageDeathMarkerManager.setEnabled(true);
    ImageDeathMarkerManager.registerGetCreatureTypeFunction(CombatManagerSotDL.myCustomCreatureType);
end
```

The first line is required for showing the Death Marker settings in Options and other Death Marker functions.

The second line overwrites the Death Marker function that checks for the Creature Type. In my example I don't have Creature Types. See next section.

CREATURE TYPES

As my ruleset doesn't have Creature Types coded in I need to return "" every time the function is run.

```
function myCustomCreatureType()
  return "";
end
```

GET WOUND PERCENT

The ActorHealthManager.isDeadOrDying uses the ActorHealthManager.getWoundPercent to determine if the target is dead/dying. In the default CoreRPG function it always returns Healthy. We need it to return "Dead" or "Dying" if we want blood splatters to replace the token.

Create a new named Script ManagerActorSDL

```
function onInit()
  initActorHealth();
end

function initActorHealth()
  ActorHealthManager.getWoundPercent = getWoundPercent;
end

function getWoundPercent(v)
  local rActor = ActorManager.resolveActor(v);

  local nHP = 0;
  local nWounds = 0;
  local nDeathSaveFail = 0;

  local nodeCT = ActorManager.getCTNode(rActor);
  if nodeCT then
    nHP = math.max(DB.getValue(nodeCT, "health", 0), 0);
    nWounds = math.max(DB.getValue(nodeCT, "damage", 0), 0);
  elseif ActorManager.isPC(rActor) then
    local nodePC = ActorManager.getCreatureNode(rActor);
    if nodePC then
      nHP = math.max(DB.getValue(nodePC, "health", 0), 0);
      nWounds = math.max(DB.getValue(nodePC, "damage", 0), 0);
    end
  end

  local nPercentWounded = 0;
  if nHP > 0 then
    nPercentWounded = nWounds / nHP;
  end

  local sStatus;
  if nPercentWounded >= 1 then
    sStatus = ActorHealthManager.STATUS_DYING;
  else
    sStatus =
  ActorHealthManager.getDefaultStatusFromWoundPercent(nPercentWounded);
  end

  return nPercentWounded, sStatus;
end
```

The second function replaces the CoreRPG function with the 3rd function in this script. This calculates the Dead/Dying status using the particular field names in my ruleset – health and damage vs hptotal and wounds