

AmbientAICallouts Visual API Documentation - Create your own AiCallout

A structure guide for good AiC's.

```
public class YourAiCalloutName : AmbientAICallouts.API.AiCallout
```

In any case: Returning Setup()/Process/End() with **false** will indicate an Error and result in not proceeding the next function

```
public override bool
```

Setup ()

Initialize your Scene before dispatch is calling the unit
This needs a true as return in order to execute the Process().

Needs:

string SceneInfo => Defines the what happend at the Scene. Examples: Robbery
Vector3 location => Sets a location for the AiCallout - **INFO**: AiCallout wont go futher than Setup if the location is not inside the minimum/maximumCalloutDistance
string calloutDetailsString => set the name of the file that shall be played when dispatch tells the unit what happend
Example: "EMERGENCY_CALL"

Optional:

float arrivalDistanceThreshold => edit for this specific AiCallout the blip size and the within the range where the unit stops

Strongly Recommended: **Keep a structure like this to prevent most difficulties with the AmbientAICallout System and other**

```
public override bool
```

Process()

Execution of the AiCallout.
Beginns after the the Unit starts Code3 response

```
try {
    if (!IsUnitInTime(100f, 130)) //if vehicle is never reaching its location
    { Disregard(); }
    else { //if vehicle is reaching its location
        GameFiber.SleepUntil(() => location.DistanceTo(Unit.Position) < arrivalDistanceThreshold + 5f, 30000);
        OfficersLeaveVehicle(true);
        if (playerRespondingInAdditon) { //if the player responds as a additional unit to the AiCallout
            // when player is also responding
        } else { //if the player is not responding as additional
            if (IsAiTakingCare()) { //do we escalate into an PlayerCallout or let the Ai handle it itself
                //do something without calling the player for backup
            } else { //do something and call the player for backup
                switch (new Random().Next(0, 5)) { //decision tree what callout shall get triggered
                    case 0:
                        UnitCallsForBackup("AAIC-OfficerDown"); break;
                    case 1:
                        UnitCallsForBackup("AAIC-OfficerInPursuit"); break;
                    default:
                        UnitCallsForBackup("AAIC-OfficerRequiringAssistance"); break;
                }
            }
        }
    }
    return true;
} catch (Exception e) {
    LogTrivial_withAiC("ERROR: in AiCallout object: At Process(): " + e);
    return false;
}
```

Every AiCallout has these objects/values.
USE them to make AiCallouts compatible to built in PlayerCallouts and to get cleaned up in case of an Error (only for Objects listed below).
You have to set your Objects as these

Values:

```
///String SceneInfo
///Vector3 location
///String callSign
///String calloutDetailsString
///Voicelines rndVI
```

Objects:

```
///Vehicle unit
///List<Ped> unitOfficers
///List<Ped> suspects <= fill this one with your suspects
///List<Vehicle> suspectsVehicle <= fill this one with your suspectVehicles
```

```
public override bool
```

End()

Clean up all of your created objects.
This is IMPORTANT

Recommended:

```
void EnterAndDismiss()
```

=> cleans up objects, officers will get into their vehicle, peds will dismiss, **all objects from above** get either dismissed or set as not persistent (depending on the lspdfr interaction)

AmbientAICallouts.API.Functions

The functions in here are all Static and needs an AiCalloutManager object which addresses an running AiCallout

Functions in here are made to interact with another running AiCallout in case you have the AiCalloutManager **ManagerObject**

```
## Name
public static string Get_SceneInfo(AiCalloutManager given_mO)

## Location get&set
public static void Set_location(AiCalloutManager given_mO, Vector3 location)
public static Vector3 Get_Location(AiCalloutManager given_mO)

## RadioSettings
public static String Get_callSign(AiCalloutManager given_mO)
public static void Set_calloutDetailsString(AiCalloutManager given_mO, String calloutDetailsString)
public static String Get_calloutDetailsString(AiCalloutManager given_mO)

## Decicion Functions
public static void Disregard(AiCalloutManager given_mO)
public static void AbortCode(AiCalloutManager given_mO)
public static void UnitCallsForBackup(AiCalloutManager given_mO, string PlayerCalloutName)

## Objects
public static Vehicle Unit(AiCalloutManager given_mO)
public static List<Ped> UnitOfficers(AiCalloutManager given_mO)
public static List<Ped> Suspects(AiCalloutManager given_mO)
public static List<Vehicle> SuspectVehicles(AiCalloutManager given_mO)
public static void SetupSuspects(AiCalloutManager given_mO, int numberOfSuspects)
public static void SetupRandomSuspects(AiCalloutManager given_mO, int numberOfSuspects, Vector3 location)

## HeliAssistance
public static void AiCandHA_RequestHelicopterTo(AiCalloutManager given_mO, Vector3 pos)
public static void AiCandHA_DismissHelicopter(AiCalloutManager given_mO)
public static void AiCandHA_AddHelicopterToPursuit(AiCalloutManager given_mO, LHandle lhandle)

## AiCalloutDistance getter
public static float minimumAiCalloutDistance
public static float maximumAiCalloutDistance

## Player direct response
public static bool Get_playerRespondingInAdditon(AiCalloutManager give_mO)
public static void AAIC_Response()

## others
public static void CleanArea(Vector3 Area, float areaRadius)
```

Please still read the API-Dokumentation.md to make sure you followed all the needed and

There are more AmbientAICallouts.API.AiCallout functions which can be found at the API-Dokumentation.md file