

## ISXEQ2 - How to determine group leader

### Introduction

To determine the leader of the current group, a script will need to iterate through the GroupMembers Window eq2ui elements associated with the names of players. Then, when you come across the one which is not a Mercenary and has the text color in yellow (or faded yellow), then this indicates that the player is the group leader. Since the character of the current player is not listed in the groupwindow, then check to see if that character is the group leader first (using `Me.IsGroupLeader`.)

### Sample

```
function main()
{
    variable int GroupCounter = 1
    variable int NumChildren
    variable string GMName
    variable string GMType
    variable string RBGAColor

    if (Me.IsGroupLeader)
        echo "0. {Me.Name} (GROUP LEADER) "
    else
        echo "0. {Me.Name}"

    do
    {
        ; If the MemberInfo page doesn't have 8 children, then it's not valid (i.e., there is no
        group member in this slot or the structure of the file has changed.)
        NumChildren:Set [EQ2UIPage[MainHUD,GroupMembers].Child[Page,GroupMember${GroupCounter}.MemberInfo
        Page.MemberInfo].NumChildren]
        if ({NumChildren} < 8)
            continue

        ; Sanity Check (if this fails, then the structure of the xml file has changed)
        if
        (!{EQ2UIPage[MainHUD,GroupMembers].Child[Page,GroupMember1.MemberInfoPage.MemberInfo].ChildType[2
        ].Equal["Text"]})
        {
            echo "CRITICAL ERROR - the eq2ui_mainhud_groupmembers.xml file must have changed."
            return
        }

        ; We should ignore mercenaries
        GMType:Set [{Me.Group[${GroupCounter}].ToActor.Type}]
        if ({GMType.Equal["Mercenary"]})
            continue

        GMName:Set [{EQ2UIPage[MainHUD,GroupMembers].Child[Page,GroupMember${GroupCounter}.MemberInfoPage.
        MemberInfo].Child[Text,2].GetProperty[Text]]}

        RBGAColor:Set [{EQ2UIPage[MainHUD,GroupMembers].Child[Page,GroupMember${GroupCounter}.MemberInfoPa
        ge.MemberInfo].Child[Text,2].GetProperty[TextColor].Right[6]}]

        ; if the TextColor is ffff (bright yellow) or 7f7f (faded yellow or olive) then the player
        is the group leader
        if ({RBGAColor.Equal["ffff00"]} || {RBGAColor.Equal["7f7f00"]})
            echo "${GroupCounter}. {GMName} is a PC with a TextColor of {RBGAColor} (GROUP
        LEADER) "
        else
            echo "${GroupCounter}. {GMName} is a PC with a TextColor of {RBGAColor}"
        }
        while ${GroupCounter:Inc} <= 5
    }
}
```

```
}
```

## Technical Notes

The most challenging thing about the snippet above is this call:

`#{EQ2UIPage[MainHUD,GroupMembers].Child[Page,GroupMember#{GroupCounter}.MemberInfoPage.MemberInfo]}`. It comes from analyzing the top portion of `eq2ui_mainhud_groupmembers.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
...
```

```
      :4bb9327fd43f489d:GroupMember
```

```
...
```

The first arguments, `[MainHUD,GroupMembers]`, comes from the file name. Then, we want to get down to the "MemberInfo" Page that has the Name text element. So, to do this, we simply follow the xml down level by level using `[Page,GroupMember5.MemberInfoPage.MemberInfo]`. (Of course, in the snippet, we use `#{GroupCounter}` instead of the actual number.)

Once you've gotten a handle on the "MemberInfo" page, then you can see that it has 8 children. The second child is a Text element named "Name" -- and that's the one we're looking for. Due to ISXEQ2 limitations, our script snippet needs to ensure that the design/structure of this xml file matches what we expect (i.e., that there are 8 children and the second one is a text element.) If this changes, then the script snippet would need to change as well.