Créer une fenêtre externe pour chopper les informations du chat :

**1) Ajout des variables et de la fonction de finalisation**

Placez ce bloc de code **en haut** de votre Guild.lua (par exemple juste après local Guild = TrinityAdmin:GetModule("Guild")), afin qu’il soit accessible partout :

-------------------------------------------------------------

-- Variables et fonctions pour la capture du .guild info

-------------------------------------------------------------

local capturingGuildInfo = false

local guildInfoCollected = {}

local guildInfoTimer = nil

-- Fonction appelée quand on arrête la capture

local function FinishGuildInfoCapture()

capturingGuildInfo = false

if #guildInfoCollected > 0 then

-- Concatène toutes les lignes

local fullText = table.concat(guildInfoCollected, "\n")

-- Affiche dans la popup

GuildInfoPopup\_SetText(fullText)

GuildInfoPopup:Show()

else

-- Aucun message capturé

TrinityAdmin:Print("No guild info was captured.")

end

end

**2) Fenêtre popup pour afficher le texte**

Toujours **hors** de la fonction CreateGuildPanel(), créez la **fenêtre popup** (similaire à ce que vous aviez fait pour le .pinfo) :

-------------------------------------------------------------

-- Fenêtre popup GuildInfoPopup pour afficher le .guild info

-------------------------------------------------------------

local GuildInfoPopup = CreateFrame("Frame", "GuildInfoPopup", UIParent, "BackdropTemplate")

GuildInfoPopup:SetSize(400, 300)

--GuildInfoPopup:SetPoint("CENTER")

GuildInfoPopup:SetPoint("TOPRIGHT", UIParent, "TOPRIGHT", -100, -100)

GuildInfoPopup:SetMovable(true)

GuildInfoPopup:EnableMouse(true)

GuildInfoPopup:RegisterForDrag("LeftButton")

GuildInfoPopup:SetScript("OnDragStart", GuidInfoPopup.StartMoving)

GuildInfoPopup:SetScript("OnDragStop", GuidInfoPopup.StopMovingOrSizing)

GuildInfoPopup:SetBackdrop({

bgFile = "Interface/DialogFrame/UI-DialogBox-Background",

edgeFile = "Interface/DialogFrame/UI-DialogBox-Border",

tile = true, tileSize = 32, edgeSize = 32,

insets = { left=8, right=8, top=8, bottom=8 }

})

GuildInfoPopup:Hide() -- Caché par défaut

-- Titre de la fenêtre

local title = GuildInfoPopup:CreateFontString(nil, "OVERLAY", "GameFontNormalLarge")

title:SetPoint("TOP", 0, -15)

title:SetText("Guild Info")

-- Bouton Close

local closeButton = CreateFrame("Button", nil, GuildInfoPopup, "UIPanelCloseButton")

closeButton:SetPoint("TOPRIGHT", GuildInfoPopup, "TOPRIGHT")

-- ScrollFrame

local scrollFrame = CreateFrame("ScrollFrame", "GuildInfoScrollFrame", GuildInfoPopup, "UIPanelScrollFrameTemplate")

scrollFrame:SetPoint("TOPLEFT", 15, -50)

scrollFrame:SetSize(370, 230)

-- Conteneur du texte

local content = CreateFrame("Frame", nil, scrollFrame)

content:SetSize(370, 230)

scrollFrame:SetScrollChild(content)

-- FontString pour le texte

local infoText = content:CreateFontString(nil, "OVERLAY", "GameFontNormal")

infoText:SetPoint("TOPLEFT")

infoText:SetWidth(350) -- un peu moins que 370 pour la marge

infoText:SetJustifyH("LEFT")

infoText:SetJustifyV("TOP")

-- Fonction pour régler le texte et ajuster la taille

function GuildInfoPopup\_SetText(text)

infoText:SetText(text or "")

local textHeight = infoText:GetStringHeight()

content:SetHeight(textHeight + 5)

scrollFrame:SetVerticalScroll(0) -- revient en haut

end

**3) Frame pour intercepter les messages du chat**

Toujours dans votre code (mais **en dehors** de CreateGuildPanel()), créez le **captureFrame** :

-------------------------------------------------------------

-- CaptureFrame pour écouter CHAT\_MSG\_SYSTEM

-------------------------------------------------------------

local guildCaptureFrame = CreateFrame("Frame")

guildCaptureFrame:RegisterEvent("CHAT\_MSG\_SYSTEM")

guildCaptureFrame:SetScript("OnEvent", function(self, event, msg)

if not capturingGuildInfo then

return

end

-- Nettoyage éventuel des codes couleur, etc.

local cleanMsg = msg

cleanMsg = cleanMsg:gsub("|c%x%x%x%x%x%x%x%x", "") -- Retire codes couleurs

cleanMsg = cleanMsg:gsub("|r", "")

cleanMsg = cleanMsg:gsub("|H.-|h(.-)|h", "%1") -- Retire liens

cleanMsg = cleanMsg:gsub("|T.-|t", "") -- Retire textures

-- Retire quelques caractères "box drawing"

cleanMsg = cleanMsg:gsub("\226[\148-\149][\128-\191]", "")

-- Ajoute la ligne au tableau

table.insert(guildInfoCollected, cleanMsg)

-- On redémarre le timer (1 seconde sans nouveau message => fin capture)

if guildInfoTimer then guildInfoTimer:Cancel() end

guildInfoTimer = C\_Timer.NewTimer(1, FinishGuildInfoCapture)

end)

## 4) Modifier le bouton « Info » pour lancer la capture

Enfin, dans **votre code existant** de la fonction Guild:CreateGuildPanel() (là où vous avez infoButton), **remplacez** le script OnClick par quelque chose qui :

1. Videra guildInfoCollected.
2. Mettra capturingGuildInfo = true.
3. Enverra la commande .guild info <ID> via SendCommand.
4. Démarrera (ou relancera) le timer pour la fin de capture.

Par exemple :

infoButton:SetScript("OnClick", function()

local gName = guildInfoNameEB:GetText()

if (not gName or gName == "" or gName == "Guild ID") then

TrinityAdmin:Print("Please specify a valid Guild ID for info!")

return

end

-- 1) On efface l'ancien texte et vide la table

GuildInfoPopup\_SetText("")

guildInfoCollected = {}

-- 2) Active le mode capture

capturingGuildInfo = true

-- 3) Envoie la commande

TrinityAdmin:SendCommand('.guild info ' .. gName)

-- 4) Lance un timer pour clôturer la capture s'il n'y a plus de message

if guildInfoTimer then

guildInfoTimer:Cancel()

end

guildInfoTimer = C\_Timer.NewTimer(1, FinishGuildInfoCapture)

-- Facultatif : on peut afficher immédiatement la popup vide.

-- GuildInfoPopup:Show()

end)