Scripts

Admin Menu v2.0

Projet

? Admin Menu v2.1 - Altis Life

Auteur: ProtoMehka **Version**: 2.1 **Date**: 2025-10-10 **Framework**: Altis Life 5.0+

? Description

Interface admin moderne et épurée pour serveurs Altis Life, avec système d'onglets, mini-map interactive, recherche de joueurs et permissions configurables via fichier de configuration centralisé.

Version Framework Status

? Fonctionnalités

? Interface Moderne

- Design épuré avec thème sombre et accents bleus
- Interface responsive avec SafeZone
- **4 onglets**: **Joueurs** / **Actions** / **Objets** / **Serveur**
- Navigation intuitive avec mise en surbrillance
- **Layout optimisé**: Liste joueurs compacte, grande carte, zone actions avec dropdown

? Recherche

- Barre de recherche par nom de joueur
- Tri alphabétique automatique
- Code couleur par faction (Bleu=Cop, Vert=Civ, Orange=EMS)

?? Mini-Map Interactive

- Affichage temps réel de la position du joueur sélectionné
- Centrage automatique sur le joueur
- Marqueur rouge avec nom
- Mise à jour toutes les 2 secondes
- **Grande taille** pour meilleure visibilité

? Informations Joueur

- Nom et Steam ID
- Faction avec code couleur
- Véhicule actuel
- Position GPS (coordonnées)
- Distance par rapport à l'admin
- **Argent (Cash et Banque)** - Récupération depuis la BDD via requête SQL

?? Système de Permissions

- **Configuration centralisée** dans `config/Config_Admin.hpp`
- 5 niveaux admin (Support → Owner)
- **Menu déroulant dynamique** avec bouton EXÉCUTER
- Couleurs par niveau de permission
- Boutons affichés uniquement selon le niveau admin

? Gestionnaire d'Objets

- **Détection d'items au sol** (WeaponHolder) dans un rayon personnalisable (1-50m)
- Affichage du contenu détaillé (armes, chargeurs, items, sacs)
- **Flèche 3D rouge** pointant vers l'item sélectionné
- **Suppression individuelle** d'items (les autres restent au sol)
- Suppression automatique du conteneur s'il devient vide

? Logging

- Toutes les actions admin sont **loguées en base de données**
- Interface de visualisation des logs **supprimée** (logging backend préservé)
- Traçabilité complète des actions administratives

? Structure

```
admin_menu.Altis/
├── config/
│   └── Config_Admin.hpp              # ⚙ Configuration centralisée
├── dialog/
│   ├── admin_menu.hpp                # ??Interface principale
│   ├── server_message.hpp            # ??Dialog message serveur
│   └── object_manager.hpp            # ??Dialog gestionnaire d'objets
└── core/admin/
    ├── fn_adminMenu.sqf              # ??Contrôleur principal
    ├── fn_adminMenuTabChange.sqf     # ??Gestion onglets + dropdown
    ├── fn_adminRefreshList.sqf       # ??Liste joueurs
    ├── fn_adminSearchPlayer.sqf      # ??Recherche
    ├── fn_adminPlayerSelected.sqf    # ?? Infos + mini-map
    ├── fn_adminQueryFinances.sqf     # ??Requête infos financières
    ├── fn_adminReceiveFinances.sqf   # ??Réception infos financières
    ├── fn_adminRepairObject.sqf      # ??Réparer objet
    ├── fn_adminDeleteObject.sqf      # ???Supprimer objet
    ├── fn_adminServerMsg.sqf         # ??Ouvrir dialog message serveur
    ├── fn_adminSendServerMsg.sqf     # ??Envoyer message serveur
    ├── fn_adminBroadcastMsg.sqf      # ??Broadcast message
    ├── fn_adminSpectate.sqf          # ???Mode spectateur (TFAR compatible)
    ├── fn_adminTpHere.sqf            # ??Téléporter joueur vers admin
    ├── fn_adminFreeze.sqf            # ❄ Geler joueur
    ├── fn_adminGodMode.sqf           # ???Mode invincibilité
    ├── fn_adminObjectManager.sqf     # ??Ouvrir gestionnaire objets
    ├── fn_adminObjectRefresh.sqf     # ??Scanner objets autour
    ├── fn_adminObjectSelect.sqf      # ??Sélectionner objet (flèche 3D)
    └── fn_adminObjectDelete.sqf      # ???Supprimer item individuel

life_server/Functions/Admin/
├── fn_getPlayerFinances.sqf          # ??Requête SQL finances
└── fn_logAdminAction.sqf             # ??Logging actions admin en BDD
```

? Installation Rapide

1?? Copier les fichiers

Client (admin_menu.Altis):

```
 config/Config_Admin.hpp
 dialog/admin_menu.hpp
 dialog/server_message.hpp
 dialog/object_manager.hpp
 core/admin/fn_admin*.sqf (28 fichiers)
```

Serveur (life_server):

```
 Functions/Admin/fn_getPlayerFinances.sqf
 Functions/Admin/fn_logAdminAction.sqf
```

2?? Modifier Config_Master.hpp

Ajouter tout en haut de la section des includes (après `class Life_Settings`):

```
#include "Config_Admin.hpp"
```

3?? Modifier Functions.hpp (Client)

```
class Admin {
    file = "core\admin";
    // ... fonctions existantes conservées ...

    // Admin Menu v2.1
    class adminMenu {};
    class adminMenuTabChange {};
    class adminRefreshList {};
    class adminSearchPlayer {};
    class adminPlayerSelected {};
    class adminRepairObject {};
    class adminDeleteObject {};
    class adminQueryFinances {};
    class adminReceiveFinances {};
    class adminServerMsg {};
    class adminSendServerMsg {};
    class adminBroadcastMsg {};
    class adminSpectate {};
```

```
    class adminTpHere {};
    class adminFreeze {};
    class adminGodMode {};
    class adminObjectManager {};
    class adminObjectRefresh {};
    class adminObjectSelect {};
    class adminObjectDelete {};
};
```

4?? Modifier MasterHandler.hpp

```
#include "admin_menu.hpp"
#include "server_message.hpp"
#include "object_manager.hpp"
```

5?? Modifier config.cpp (Serveur)

```
class Admin {
    file = "\life_server\Functions\Admin";
    class getPlayerFinances {};
    class logAdminAction {};
};
```

6?? Modifier CfgRemoteExec.hpp (Client)

Client functions:

```
F(life_fnc_adminQueryFinances,CLIENT)
F(life_fnc_adminReceiveFinances,CLIENT)
F(life_fnc_adminBroadcastMsg,CLIENT)
```

Server functions:

```
F(TON_fnc_getPlayerFinances,SERVER)
F(TON_fnc_logAdminAction,SERVER)
```

7?? Modifier player_inv.hpp (Client)

Bouton IDC 2021:

```
onButtonClick = "closeDialog 0; createDialog ""life_admin_menu"";";
```

8?? Base de données

Voir INTEGRATION_DB.md pour la création de la table `admin_logs`.

? C'est prêt !

Appuyez sur `Y` → **Admin Menu**

? Documentation

Document	Description
INTEGRATION_CLIENT.md	◫ Installation client (mission)
INTEGRATION_SERVER.md	◫ Installation serveur
INTEGRATION_DB.md	◫ Configuration base de données
CONFIG_ADMIN_GUIDE.md	◫ Guide complet de configuration
CHANGELOG.md	◫ Journal des modifications

?? Configuration

Ajouter une Commande

Éditer `config/Config_Admin.hpp` :

```
class Commands {
    list[] = {
        {"Ma Commande", "life_fnc_maFonction", 3, "players"}
        // Nom | Fonction | Niveau min (1-5) | Onglet
    };
```

```
};
```

Onglets Disponibles

- `"players"` → Onglet Joueurs
- `"actions"` → Onglet Actions
- `"objects"` → Onglet Objets
- `"server"` → Onglet Serveur

Dropdown / Boutons

Modifier dans `Config_Admin.hpp` :

```
class DropdownSettings {
    maxButtonsBeforeDropdown = 0;  // 0 = Toujours dropdown, 999 = Toujours boutons
};
```

? Niveaux Admin

Niveau	Nom	Couleur	Permissions
1	Support	◫ Vert	Infos joueurs
2	Modérateur	◫ Bleu clair	+ Compensate
3	Admin	◫ Bleu	+ Spectate, Teleport, Repair Object, **Object Manager**
4	Super Admin	◫ Orange	+ TpHere, GodMode, Freeze, Delete Object, Server Message
5	Owner	◫ Rouge	Accès total (Debug Console)

? Compatibilité

- □ Altis Life 5.0 (AsYetUntitled)
- □ Frameworks Altis Life compatibles
- □ Arma 3 (dernière version)
- ⚠ Requiert adaptation mineure pour frameworks custom

?? Troubleshooting

Le menu ne s'ouvre pas

 Vérifier `life_adminlevel >= 1` Vérifier includes dans `MasterHandler.hpp` et `Config_Master.hpp`

Liste de joueurs vide

 Vérifier logs `[ADMIN REFRESH]` dans le RPT Tester `hint str (count playableUnits);`

Aucune commande affichée

 Vérifier que `Config_Admin.hpp` est bien inclus dans `Config_Master.hpp` Vérifier les logs `[ADMIN MENU]` pour voir si les commandes sont chargées

Plus de détails: CONFIG_ADMIN_GUIDE.md

? Fonctionnalités Implémentées

- ☑ Interface moderne avec dropdown
- ☑ Configuration centralisée (Config_Admin.hpp)
- ☑ Server Message broadcast
- ☑ Repair/Delete Object
- ☑ Informations financières via SQL
- ☑ Logging actions admin en BDD
- ☑ Layout optimisé (liste compacte, grande carte)
- ☑ Textes agrandis pour meilleure lisibilité
- ☑ **Mode spectateur compatible TFAR** (détection automatique)
- ☑ **Gestionnaire d'objets** avec suppression individuelle d'items
- ☑ Corrections bugs (TP Here, Freeze, God Mode)

? Licence

Libre d'utilisation et de modification. Crédit apprécié mais non obligatoire.

? Crédits

Auteur: ProtoMehka **Framework**: Altis Life 5.0 (AsYetUntitled) **Date**: Octobre 2024 - Janvier 2025

Version 2.1 - Octobre 2025

Intégration Client

?? Guide d'Intégration Client - Admin Menu v2.1

Auteur: ProtoMehka **Version**: 2.1 **Date**: 10 Janvier 2025 **Framework**: Altis Life 5.0+

? Table des Matières

? Prérequis

Avant de commencer, assurez-vous d'avoir :

- ☐ Framework Altis Life 5.0 ou supérieur
- ☐ Accès aux fichiers de la mission (`admin_menu.Altis/`)
- ☐ Éditeur de texte (Notepad++, VSCode, etc.)
- ☐ Backup complet de votre mission

? Fichiers à Copier

Étape 1 : Nouveaux Fichiers

Copiez les fichiers suivants dans votre mission `admin_menu.Altis/` :

Configuration (1 fichier)

```
admin_menu.Altis/config/
└── Config_Admin.hpp          NOUVEAU
```

Interface (3 fichiers)

```
admin_menu.Altis/dialog/
├── admin_menu.hpp             ⚠ REMPLACER
├── server_message.hpp         ⚠ REMPLACER
└── object_manager.hpp         NOUVEAU (v2.1)
```

Fonctions Admin Core (12 fichiers)

```
admin_menu.Altis/core/admin/
├── fn_adminMenu.sqf               ⚠ REMPLACER
├── fn_adminMenuTabChange.sqf      ⚠ REMPLACER
├── fn_adminRefreshList.sqf        ⚠ REMPLACER
├── fn_adminSearchPlayer.sqf       ⚠ REMPLACER
├── fn_adminPlayerSelected.sqf     ⚠ REMPLACER
├── fn_adminQueryFinances.sqf      ⚠ REMPLACER
├── fn_adminReceiveFinances.sqf    ⚠ REMPLACER
├── fn_adminRepairObject.sqf       ⚠ REMPLACER
├── fn_adminDeleteObject.sqf       ⚠ REMPLACER
├── fn_adminServerMsg.sqf          ⚠ REMPLACER
├── fn_adminSendServerMsg.sqf      ⚠ REMPLACER
└── fn_adminBroadcastMsg.sqf       ⚠ REMPLACER
```

Fonctions Admin - Corrections v2.1 (5 fichiers)

```
admin_menu.Altis/core/admin/
├── fn_adminSpectate.sqf           ⚠ REMPLACER (TFAR compatible)
├── fn_adminTpHere.sqf             ⚠ REMPLACER (IDC corrigé)
├── fn_adminFreeze.sqf             ⚠ REMPLACER (IDC corrigé)
└── fn_adminGodMode.sqf            ⚠ REMPLACER (closeDialog supprimé)
```

Object Manager - NOUVEAU v2.1 (4 fichiers)

```
admin_menu.Altis/core/admin/
├── fn_adminObjectManager.sqf      □ NOUVEAU
├── fn_adminObjectRefresh.sqf      □ NOUVEAU
├── fn_adminObjectSelect.sqf       □ NOUVEAU
└── fn_adminObjectDelete.sqf       □ NOUVEAU
```

Autres Fonctions Admin (3 fichiers)

```
admin_menu.Altis/core/admin/
├── fn_adminCompensate.sqf         ⚠□ REMPLACER (logging ajouté)
├── fn_adminDebugCon.sqf           ⚠□ REMPLACER (logging ajouté)
└── fn_adminMarkers.sqf            ⚠□ REMPLACER (logging ajouté)
```

Fonction Teleport (1 fichier)

```
admin_menu.Altis/core/functions/
└── fn_teleport.sqf                ⚠□ REMPLACER (logging ajouté)
```

? Modifications des Fichiers de Configuration

Étape 2 : Config_Master.hpp

Fichier: `admin_menu.Altis/config/Config_Master.hpp`

Localiser la section des includes (généralement au début du fichier, après `class Life_Settings`).

Ajouter cette ligne :

```
#include "Config_Admin.hpp"
```

Exemple de placement :

```
class Life_Settings {
    /* ... autres paramètres ... */
};

#include "Config_Admin.hpp"      // ← AJOUTER ICI
```

```
#include "Config_Master.hpp"
#include "Config_Clothing.hpp"
// ... autres includes ...
```

Étape 3 : Functions.hpp

Fichier: `admin_menu.Altis/Functions.hpp`

Localiser la classe `Admin` (environ ligne 70-90).

Remplacer ou **compléter** avec les nouvelles fonctions :

```
class Admin {
    file = "core\admin";

    // Fonctions existantes (conserver si elles existent)
    class adminCompensate {};
    class adminDebugCon {};
    class adminFreeze {};
    class admingetID {};
    class adminGodMode {};
    class adminid {};
    class admininfo {};
    class adminMarkers {};
    class adminQuery {};
    class adminSpectate {};
    class adminTeleport {};
    class adminTpHere {};

    // □ NOUVELLES FONCTIONS - Admin Menu v2.1
    class adminMenu {};
    class adminMenuTabChange {};
    class adminRefreshList {};
    class adminSearchPlayer {};
    class adminPlayerSelected {};
    class adminQueryFinances {};
    class adminReceiveFinances {};
    class adminRepairObject {};
    class adminDeleteObject {};
```

```
    class adminServerMsg {};
    class adminSendServerMsg {};
    class adminBroadcastMsg {};

    // □ Object Manager v2.1
    class adminObjectManager {};
    class adminObjectRefresh {};
    class adminObjectSelect {};
    class adminObjectDelete {};
};
```

Étape 4 : MasterHandler.hpp

Fichier: `admin_menu.Altis/dialog/MasterHandler.hpp`

Localiser les includes de dialogs admin.

Modifier pour inclure les nouveaux dialogs :

```
// Admin Menu v2.1
#include "admin_menu.hpp"
#include "server_message.hpp"
#include "object_manager.hpp"
```

⚠ **Important** : Supprimez l'ancien include si présent :

```
// #include "admin_menu_new.hpp"   ← SUPPRIMER si présent
```

Étape 5 : CfgRemoteExec.hpp

Fichier: `admin_menu.Altis/CfgRemoteExec.hpp`

Ajouter les permissions pour les nouvelles fonctions client :

```
// Admin Menu v2.1 - Client Functions
F(life_fnc_adminQueryFinances,CLIENT)
F(life_fnc_adminReceiveFinances,CLIENT)
F(life_fnc_adminBroadcastMsg,CLIENT)
```

Ajouter les permissions pour les fonctions serveur :

```
// Admin Menu v2.1 - Server Functions
F(TON_fnc_getPlayerFinances,SERVER)
F(TON_fnc_logAdminAction,SERVER)
```

Exemple complet d'une section :

```
class CfgRemoteExec {
    class Functions {
        mode = 1;
        jip = 0;

        // ... autres fonctions ...

        // Admin Menu v2.0 - Client
        F(life_fnc_adminQueryFinances,CLIENT)
        F(life_fnc_adminReceiveFinances,CLIENT)
        F(life_fnc_adminBroadcastMsg,CLIENT)

        // Admin Menu v2.0 - Server
        F(TON_fnc_getPlayerFinances,SERVER)
        F(TON_fnc_logAdminAction,SERVER)
    };
};
```

Étape 6 : player_inv.hpp

Fichier: `admin_menu.Altis/dialog/player_inv.hpp`

Localiser le bouton "Admin Menu" (généralement IDC **2021**).

Modifier l'action `onButtonClick` :

```
class ButtonAdminMenu: Life_RscButtonMenu {
    idc = 2021;
    text = "$STR_PM_AdminMenu";
    colorBackground[] = {"(profilenamespace getvariable ['GUI_BCG_RGB_R',0.3843])",
"(profilenamespace getvariable ['GUI_BCG_RGB_G',0.7019])", "(profilenamespace getvariable
['GUI_BCG_RGB_B',0.8862])", 0.5};
```

```
    onButtonClick = "closeDialog 0; createDialog ""life_admin_menu"";";  // ← MODIFIER ICI
    x = 0.0204 * safezoneW + safezoneX;
    y = 0.82 * safezoneH + safezoneY;
    w = 0.0392 * safezoneW;
    h = 0.033 * safezoneH;
};
```

⚠ **Important** : L'ancien nom était potentiellement `life_admin_menu_new`, vérifiez et changez en `life_admin_menu`.

Étape 7 : common.hpp (Optionnel)

Fichier: `admin_menu.Altis/dialog/common.hpp`

Ce fichier a reçu des **modifications mineures** (+1 ligne). Si vous avez des customisations importantes dans votre `common.hpp`, vérifiez les différences.

Conseil : Si vous n'avez pas modifié ce fichier, remplacez-le par la nouvelle version.

? Nouveautés v2.1

Corrections de Bugs

- **TP Here** : IDC corrigé (2902 → 29020)
- **Freeze Player** : IDC corrigé (2902 → 29020)
- **God Mode** : Ne ferme plus le menu
- **Server Message** : Fonction et remoteExec corrigés
- **Spectate** : Touche F fonctionnelle, caméra repositionnée, compatible TFAR

Nouvelle Fonctionnalité

- **Object Manager** : Gestionnaire d'items au sol
  - Détection items dans rayon personnalisable (1-50m)
  - Affichage contenu WeaponHolder détaillé
  - Flèche 3D rouge sur item sélectionné
  - **Suppression individuelle** d'items
  - Les autres items restent au sol

Version 2.1 - 10 Janvier 2025

Intégration Serveur

? Guide d'Intégration Serveur - Admin Menu v2.1

Auteur: ProtoMehka **Version**: 2.1 **Date**: 10 Janvier 2025 **Framework**: Altis Life 5.0+

? Introduction

Ce guide détaille l'installation de la **partie serveur** du menu admin v2.1. Il couvre :

- Les fonctions serveur pour les requêtes SQL
- Le système de logging des actions admin
- La configuration du serveur
- Les permissions remoteExec

Prérequis :

- Serveur Altis Life fonctionnel avec extDB3
- Accès aux fichiers `life_server`
- Base de données MySQL/MariaDB configurée

? Structure des Fichiers Serveur

```
life_server/
└── Functions/
    └── Admin/
        ├── fn_getPlayerFinances.sqf    # Récupère cash/banque d'un joueur
        └── fn_logAdminAction.sqf        # Enregistre les actions admin en BDD
```

? Installation

1?? Créer le Dossier Admin

Si le dossier n'existe pas déjà :

```
life_server/Functions/Admin/
```

2?? Copier les Fichiers

Copiez les 2 fichiers `.sqf` dans `life_server/Functions/Admin/` :

```
 fn_getPlayerFinances.sqf
 fn_logAdminAction.sqf
```

3?? Modifier config.cpp

Chemin : `life_server/config.cpp`

Localiser la classe `TON_System` :

```
class TON_System {
    tag = "TON";

    // ... autres classes existantes ...
};
```

Ajouter cette classe **DANS** `TON_System` :

```
class Admin {
    file = "\life_server\Functions\Admin";
    class getPlayerFinances {};
    class logAdminAction {};
};
```

Résultat final :

```
class TON_System {
    tag = "TON";
```

```
    // Autres classes...
    class MySQL_Database {
        // ...
    };

    class Admin {
        file = "\life_server\Functions\Admin";
        class getPlayerFinances {};
        class logAdminAction {};
    };
};
```

4?? Modifier CfgRemoteExec.hpp (Client)

Chemin : `admin_menu.Altis/CfgRemoteExec.hpp`

Ajouter dans la section `/* Server only functions */` :

```
F(TON_fnc_getPlayerFinances,SERVER)
F(TON_fnc_logAdminAction,SERVER)
```

Exemple complet :

```
/* Server only functions */
F(DB_fnc_insertRequest,SERVER)
F(DB_fnc_queryRequest,SERVER)
F(DB_fnc_updatePartial,SERVER)
F(DB_fnc_updateRequest,SERVER)
F(TON_fnc_getPlayerFinances,SERVER)
F(TON_fnc_logAdminAction,SERVER)
F(life_fnc_jailSys,SERVER)
// ... autres fonctions serveur
```

? Détail des Fonctions

fn_getPlayerFinances.sqf

Description : Récupère le cash et la banque d'un joueur depuis la base de données.

Paramètres :

- `0: STRING` - PID du joueur cible
- `1: OBJECT` - Admin qui demande les infos

Fonctionnement :

1. Reçoit une demande du client (admin)
2. Exécute : `SELECT cash, bankacc FROM players WHERE pid='...'`
3. Envoie les résultats au client via `remoteExecCall`

Exemple de requête SQL :

```
SELECT cash, bankacc FROM players WHERE pid='76561198012345678'
```

---

fn_logAdminAction.sqf

Description : Enregistre une action admin dans la base de données.

Paramètres :

- `0: STRING` - PID de l'admin
- `1: STRING` - Nom de l'admin
- `2: STRING` - Commande exécutée
- `3: STRING` - Cible (optionnel, nom du joueur ciblé)
- `4: STRING` - Détails additionnels (optionnel)

Fonctionnement :

1. Reçoit les données d'une action admin
2. Échappe les guillemets pour la sécurité SQL
3. Insère dans la table `admin_logs`

Exemple de requête SQL :

```
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'Admin Name', 'TELEPORT', NULL, 'X:3612 Y:13138')
```

Gestion des valeurs NULL :

- Si `target` ou `details` sont vides, ils sont insérés comme `NULL` et non comme chaînes vides

? Sécurité

Protection SQL Injection

Les fonctions utilisent `regexReplace` pour échapper les guillemets :

```
_adminName = _adminName regexReplace ["'", "''"];
_target = _target regexReplace ["'", "''"];
_details = _details regexReplace ["'", "''"];
_command = _command regexReplace ["'", "''"];
```

Validation des Données

Vérifications avant exécution :

- PID et commande non vides
- Objet admin valide
- Limite de résultats respectée

? Checklist d'Installation

```
□ Dossier life_server/Functions/Admin créé
□ 2 fichiers .sqf copiés
□ config.cpp modifié (classe Admin ajoutée avec 2 fonctions)
□ CfgRemoteExec.hpp modifié (2 fonctions serveur ajoutées)
□ Table admin_logs créée en BDD (voir INTEGRATION_DB.md)
□ Serveur compilé sans erreur
□ Tests effectués avec succès
```

? Nouveautés v2.1

Fonctionnalités Utilisant le Serveur

Object Manager :

- Logging des suppressions d'items : `DELETE_ITEM | Type: weapon | Class: hgun_Rook40_F`
- Utilise `TON_fnc_logAdminAction` pour tracer les suppressions

Actions Admin avec Logging Renforcé :

- Toutes les commandes admin sont loguées via `TON_fnc_logAdminAction`
- Logs stockés dans la table `admin_logs` avec timestamp automatique

Spectate Mode :

- Compatible TFAR (détection automatique)
- Logging de l'action spectate avec nom de la cible

Pas de Modification Serveur Requise

Les fonctions serveur v2.0 sont **100% compatibles** avec v2.1 :

- ✅ `fn_getPlayerFinances.sqf` - Inchangé
- ✅ `fn_logAdminAction.sqf` - Inchangé

Aucune modification côté serveur n'est nécessaire pour migrer de v2.0 vers v2.1.

Version 2.1 - 10 Janvier 2025

Intégration Base de données

? Guide d'Intégration Base de Données - Admin Menu v2.1

Auteur: ProtoMehka **Version**: 2.1 **Date**: 10 Janvier 2025 **Framework**: Altis Life 5.0+

? Introduction

Ce guide détaille la configuration de la **base de données** pour le menu admin v2.1. Il couvre :

- La table `admin_logs` pour l'historique des actions
- Les modifications de la structure existante
- Les requêtes SQL nécessaires

Prérequis :

- Base de données MySQL/MariaDB fonctionnelle
- Accès phpMyAdmin ou ligne de commande MySQL
- Table `players` existante

?? Structure de la Base de Données

Table: `admin_logs`

Cette table enregistre toutes les actions effectuées par les admins.

Schéma Complet

```
CREATE TABLE IF NOT EXISTS `admin_logs` (
    `id`             INT NOT NULL AUTO_INCREMENT,
```

```
    `pid`            VARCHAR(17) NOT NULL,
    `player_name`    VARCHAR(32) NOT NULL,
    `target`         VARCHAR(64) DEFAULT NULL,
    `execution_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `command`        VARCHAR(64) NOT NULL,
    `details`        TEXT,

    PRIMARY KEY (`id`),
    INDEX `index_pid` (`pid`),
    INDEX `index_command` (`command`),
    INDEX `index_execution_time` (`execution_time`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Description des Colonnes

Colonne	Type	Nullable	Description
`id`	INT	Non	ID unique auto-incrémenté
`pid`	VARCHAR(17)	Non	Steam ID de l'admin (76561198...)
`player_name`	VARCHAR(32)	Non	Nom du joueur admin
`target`	VARCHAR(64)	Oui	Nom du joueur ciblé (si applicable)
`execution_time`	TIMESTAMP	Non	Date/heure de l'action (auto)
`command`	VARCHAR(64)	Non	Type d'action (TELEPORT, GODMODE, etc.)
`details`	TEXT	Oui	Informations additionnelles (coordonnées, montant, etc.)

Index

- **PRIMARY KEY** sur `id` : Identifiant unique
- **INDEX** sur `pid` : Recherche rapide par admin
- **INDEX** sur `command` : Filtrage par type d'action
- **INDEX** sur `execution_time` : Tri chronologique optimisé

? Installation

Méthode 1: Script SQL Complet

Fichier : `altislife.sql` (fourni)

Exécutez le fichier SQL complet qui contient la table `admin_logs` avec toutes les autres tables du serveur.

Via phpMyAdmin :

1. Sélectionner la base de données
2. Onglet "Importer"
3. Choisir `altislife.sql`
4. Cliquer "Exécuter"

Via ligne de commande :

```
mysql -u username -p database_name < altislife.sql
```

Méthode 2: Ajout Manuel de la Table

Si vous avez déjà une base de données existante :

```
CREATE TABLE IF NOT EXISTS `admin_logs` (
    `id`             INT NOT NULL AUTO_INCREMENT,
    `pid`            VARCHAR(17) NOT NULL,
    `player_name`    VARCHAR(32) NOT NULL,
    `target`         VARCHAR(64) DEFAULT NULL,
    `execution_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `command`        VARCHAR(64) NOT NULL,
    `details`        TEXT,

    PRIMARY KEY (`id`),
    INDEX `index_pid` (`pid`),
    INDEX `index_command` (`command`),
    INDEX `index_execution_time` (`execution_time`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

? Types d'Actions Loggées

Liste des Commandes

Commande	Cible	Détails	Exemple
`COMPENSATE`	Nom joueur	Montant	`Amount: $50000`
`TELEPORT`	-	Coordonnées	`X:3612 Y:13138`
`TP_HERE`	Nom joueur	-	-
`GODMODE`	-	État	`ON` ou `OFF`
`FREEZE`	Nom joueur	-	-
`SPECTATE`	Nom joueur	-	-
`REPAIR_OBJECT`	-	Type objet	`Type: C_Offroad_01_F`
`DELETE_OBJECT`	-	Type objet	`Type: C_Offroad_01_F`
`DELETE_ITEM`	-	Type et classe	`Type: weapon | Class: hgun_Rook40_F`
`SERVER_MESSAGE`	-	Message	Texte du message
`DEBUG_CONSOLE`	-	-	-
`MARKERS`	-	État	`ON` ou `OFF`

Exemples de Données

```
-- Teleport
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'AdminName', 'TELEPORT', NULL, 'X:3612 Y:13138');

-- Compensation
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'AdminName', 'COMPENSATE', 'PlayerName', 'Amount: $50000');

-- GodMode ON
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'AdminName', 'GODMODE', NULL, 'ON');

-- Freeze Player
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'AdminName', 'FREEZE', 'TargetPlayer', NULL);

-- Delete Item (v2.1)
INSERT INTO admin_logs (pid, player_name, command, target, details)
```

```
VALUES ('76561198012345678', 'AdminName', 'DELETE_ITEM', NULL, 'Type: weapon | Class:
hgun_Rook40_F');
```

? Requêtes Utiles

Voir les Dernières Actions

```
SELECT
    id,
    player_name AS Admin,
    command AS Action,
    target AS Cible,
    details AS Détails,
    execution_time AS Date
FROM admin_logs
ORDER BY execution_time DESC
LIMIT 50;
```

Filtrer par Admin

```
SELECT * FROM admin_logs
WHERE pid = '76561198012345678'
ORDER BY execution_time DESC;
```

Filtrer par Type d'Action

```
SELECT * FROM admin_logs
WHERE command = 'TELEPORT'
ORDER BY execution_time DESC
LIMIT 20;
```

Actions sur un Joueur Spécifique

```
SELECT * FROM admin_logs
WHERE target = 'PlayerName'
```

```
ORDER BY execution_time DESC;
```

Statistiques par Admin

```
SELECT
    player_name AS Admin,
    COUNT(*) AS Total_Actions,
    COUNT(DISTINCT command) AS Types_Actions
FROM admin_logs
GROUP BY player_name
ORDER BY Total_Actions DESC;
```

Statistiques par Type d'Action

```
SELECT
    command AS Action,
    COUNT(*) AS Nombre
FROM admin_logs
GROUP BY command
ORDER BY Nombre DESC;
```

Actions des Dernières 24h

```
SELECT * FROM admin_logs
WHERE execution_time >= NOW() - INTERVAL 24 HOUR
ORDER BY execution_time DESC;
```

?? Maintenance

Nettoyage des Vieux Logs

Pour éviter que la table ne devienne trop volumineuse :

Supprimer les logs de plus de 30 jours :

```
DELETE FROM admin_logs
WHERE execution_time < NOW() - INTERVAL 30 DAY;
```

Supprimer les logs de plus de 90 jours :

```
DELETE FROM admin_logs
WHERE execution_time < NOW() - INTERVAL 90 DAY;
```

Archivage (Recommandé)

Au lieu de supprimer, créer une table d'archive :

```
-- Créer la table d'archive
CREATE TABLE IF NOT EXISTS `admin_logs_archive` LIKE `admin_logs`;

-- Copier les vieux logs
INSERT INTO admin_logs_archive
SELECT * FROM admin_logs
WHERE execution_time < NOW() - INTERVAL 90 DAY;

-- Supprimer de la table principale
DELETE FROM admin_logs
WHERE execution_time < NOW() - INTERVAL 90 DAY;
```

Optimisation de la Table

Après de nombreux suppressions :

```
OPTIMIZE TABLE admin_logs;
```

? Surveillance

Taille de la Table

```
SELECT
    table_name AS `Table`,
```

```
    ROUND(((data_length + index_length) / 1024 / 1024), 2) AS `Size (MB)`
FROM information_schema.TABLES
WHERE table_schema = 'altislife'
    AND table_name = 'admin_logs';
```

Nombre de Logs

```
SELECT COUNT(*) AS Total_Logs FROM admin_logs;
```

Logs par Jour (7 derniers jours)

```
SELECT
    DATE(execution_time) AS Jour,
    COUNT(*) AS Nombre_Actions
FROM admin_logs
WHERE execution_time >= NOW() - INTERVAL 7 DAY
GROUP BY DATE(execution_time)
ORDER BY Jour DESC;
```

? Sécurité

Permissions Recommandées

L'utilisateur MySQL du serveur Arma devrait avoir uniquement :

```
GRANT SELECT, INSERT ON altislife.admin_logs TO 'arma_user'@'localhost';
```

Ne pas donner :

- ❌ `DELETE` - Pour éviter suppression accidentelle
- ❌ `UPDATE` - Les logs ne doivent jamais être modifiés
- ❌ `DROP` - Protection contre suppression de table

Sauvegarde

Sauvegarder régulièrement la table :

```
mysqldump -u username -p altislife admin_logs > admin_logs_backup_$(date +%Y%m%d).sql
```

Restaurer depuis une sauvegarde :

```
mysql -u username -p altislife < admin_logs_backup_20251007.sql
```

? Tests

Test 1: Création de la Table

```
SHOW CREATE TABLE admin_logs;
```

Résultat attendu : Affiche la structure complète de la table

Test 2: Insertion Manuelle

```
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'TestAdmin', 'TEST', 'TestTarget', 'Test details');
```

Vérification :

```
SELECT * FROM admin_logs WHERE command = 'TEST';
```

Test 3: Requête de Lecture

```
SELECT * FROM admin_logs ORDER BY execution_time DESC LIMIT 10;
```

Résultat attendu : Les 10 derniers logs

? Checklist d'Installation

```
□ Table admin_logs créée
□ Index créés (pid, command, execution_time)
□ Charset utf8mb4 configuré
□ Permissions utilisateur correctes
```

```
□ Sauvegarde configurée
□ Test d'insertion réussi
□ Test de lecture réussi
```

? Nouveautés v2.1

Nouvelles Actions Loggées

Object Manager :

- `DELETE_ITEM` - Suppression d'items individuels depuis le gestionnaire d'objets
- Format des détails : `Type: weapon | Class: hgun_Rook40_F`
- Types possibles : `weapon`, `magazine`, `item`, `backpack`

Exemple de requête pour Object Manager :

```
SELECT * FROM admin_logs
WHERE command = 'DELETE_ITEM'
ORDER BY execution_time DESC
LIMIT 20;
```

Actions Admin avec Logging Renforcé :

- Toutes les commandes admin (11 au total + Object Manager)
- Logs automatiques avec timestamp
- Aucune modification de structure nécessaire

Compatibilité

La structure de la table `admin_logs` v2.0 est **100% compatible** avec v2.1 :

- □ Aucune modification de schéma requise
- □ Les nouvelles actions utilisent les colonnes existantes
- □ Migration transparente depuis v2.0

? Exemple de Rapport

Rapport Hebdomadaire

```
SELECT
    player_name AS Admin,
    command AS Action,
    COUNT(*) AS Total,
    DATE(execution_time) AS Jour
FROM admin_logs
WHERE execution_time >= NOW() - INTERVAL 7 DAY
GROUP BY player_name, command, DATE(execution_time)
ORDER BY execution_time DESC;
```

Top 10 Admins Actifs

```
SELECT
    player_name AS Admin,
    COUNT(*) AS Total_Actions
FROM admin_logs
WHERE execution_time >= NOW() - INTERVAL 30 DAY
GROUP BY player_name
ORDER BY Total_Actions DESC
LIMIT 10;
```

Installation base de données terminée ! □□

Le système de logging est maintenant complètement opérationnel.

Version 2.1 - 10 Janvier 2025

Guide de configuration

? Guide de Configuration Admin - Config_Admin.hpp

Auteur: ProtoMehka **Version**: 2.1 **Dernière mise à jour**: 2025-10-10

? Objectif

Centraliser toute la configuration du menu admin dans un seul fichier pour permettre :

- Définir les niveaux admin requis pour chaque commande
- Personnaliser les couleurs des boutons par niveau
- Organiser les commandes dans des onglets personnalisés
- Gérer les menus déroulants pour les onglets avec beaucoup d'actions
- Modifier facilement sans toucher au code

? Structure du Fichier

Localisation

```
admin_menu.Altis/config/Config_Admin.hpp
```

Architecture

```
class Life_AdminConfig {
    // Niveaux d'administration
    class AdminLevels { ... };

    // Couleurs des boutons par niveau
```

```
    class ButtonColors { ... };

    // Configuration des onglets
    class Tabs { ... };

    // Commandes par catégorie
    class PlayerManagement { ... };
    class AdminActions { ... };
    class ObjectManagement { ... };
    class ServerManagement { ... };
};
```

? Configuration des Couleurs

Couleurs par Niveau Admin

```
class ButtonColors {
    // Format: {R, G, B, Alpha}
    // Valeurs de 0.0 à 1.0

    class Level1 {  // Support
        normal[] = {0.3, 0.7, 0.3, 0.7};      // Vert normal
        hover[] = {0.4, 0.8, 0.4, 0.9};       // Vert hover
        pressed[] = {0.2, 0.6, 0.2, 0.8};     // Vert pressé
        disabled[] = {0.3, 0.3, 0.3, 0.5};    // Gris désactivé
    };

    class Level2 {  // Modérateur
        normal[] = {0.4, 0.6, 0.8, 0.7};      // Bleu clair
        hover[] = {0.5, 0.7, 0.9, 0.9};
        pressed[] = {0.3, 0.5, 0.7, 0.8};
        disabled[] = {0.3, 0.3, 0.3, 0.5};
    };

    class Level3 {  // Admin
        normal[] = {0.38, 0.70, 0.89, 0.7};   // Bleu Altis
        hover[] = {0.48, 0.80, 0.99, 0.9};
```

```
        pressed[] = {0.28, 0.60, 0.79, 0.8};
        disabled[] = {0.3, 0.3, 0.3, 0.5};
    };

    class Level4 {  // Super Admin
        normal[] = {0.8, 0.5, 0.2, 0.7};      // Orange
        hover[] = {0.9, 0.6, 0.3, 0.9};
        pressed[] = {0.7, 0.4, 0.1, 0.8};
        disabled[] = {0.3, 0.3, 0.3, 0.5};
    };

    class Level5 {  // Owner
        normal[] = {0.8, 0.2, 0.2, 0.7};      // Rouge
        hover[] = {0.9, 0.3, 0.3, 0.9};
        pressed[] = {0.7, 0.1, 0.1, 0.8};
        disabled[] = {0.3, 0.3, 0.3, 0.5};
    };
};
```

Palette de Couleurs Recommandées

Couleur	RGB	Usage Recommandé
◫ Vert	`{0.3, 0.7, 0.3}`	Niveau 1 (Support)
◫ Bleu Clair	`{0.4, 0.6, 0.8}`	Niveau 2 (Modérateur)
◫ Bleu	`{0.38, 0.70, 0.89}`	Niveau 3 (Admin)
◫ Orange	`{0.8, 0.5, 0.2}`	Niveau 4 (Super Admin)
◫ Rouge	`{0.8, 0.2, 0.2}`	Niveau 5 (Owner)
∘ Gris	`{0.3, 0.3, 0.3}`	Désactivé

?? Configuration des Onglets

Structure de Base

```
class Tabs {
    // Nombre total d'onglets
```

```
    count = 4;

    // Configuration de chaque onglet
    // Format: {Nom, ID, Icône, Couleur}

    tabs[] = {
        {
            "JOUEURS",                                          // Nom affiché
            "players",                                           // ID interne
            "\A3\ui_f\data\GUI\Rsc\RscDisplayArsenal\face_ca.paa",  // Icône
            {0.38, 0.70, 0.89, 0.7}                             // Couleur
        },
        {
            "ACTIONS",
            "actions",
            "\A3\ui_f\data\IGUI\Cfg\Actions\settimer_ca.paa",
            {0.8, 0.5, 0.2, 0.7}
        },
        {
            "OBJETS",
            "objects",
            "\A3\ui_f\data\IGUI\Cfg\Actions\getindriver_ca.paa",
            {0.3, 0.7, 0.3, 0.7}
        },
        {
            "SERVEUR",
            "server",
            "\A3\ui_f\data\IGUI\Cfg\Actions\settimer_ca.paa",
            {0.8, 0.2, 0.2, 0.7}
        }
    };
};
```

Ajouter un Nouvel Onglet

Exemple: Ajouter un onglet "UTILITAIRES"

```
class Tabs {
    count = 5;  // Incrémenter le nombre
```

```
    tabs[] = {
        // ... onglets existants ...
        {
            "UTILITAIRES",
            "utilities",
            "\A3\ui_f\data\IGUI\Cfg\Actions\repair_ca.paa",
            {0.5, 0.5, 0.9, 0.7}
        }
    };
};
```

? Configuration des Commandes

Format de Base

```
// Format d'une commande:
{
    "Nom Affiché",           // Texte du bouton
    "fonction_sqf",          // Fonction à appeler
    niveau_minimum,          // Niveau admin requis (1-5)
    "onglet_id",            // Dans quel onglet placer
    "icone"                 // (Optionnel) Chemin vers icône
}
```

Exemple Complet

```
class PlayerManagement {
    // Toutes les commandes liées aux joueurs
    commands[] = {
        // Niveau 1 - Support
        {"Get ID", "life_fnc_adminGetID", 1, "players", ""},
        {"Query Player", "life_fnc_adminQuery", 1, "players", ""},

        // Niveau 2 - Modérateur
        {"Compensate", "life_fnc_adminCompensate", 2, "players", ""},
```

```
        // Niveau 3 - Admin
        {"Spectate", "life_fnc_adminSpectate", 3, "players", ""},
        {"TP to Player", "life_fnc_adminTeleport", 3, "players", ""},
        {"Heal Player", "life_fnc_adminHeal", 3, "players", ""},
        {"Revive Player", "life_fnc_adminRevive", 3, "players", ""},

        // Niveau 4 - Super Admin
        {"TP Here", "life_fnc_adminTpHere", 4, "players", ""},
        {"Freeze Player", "life_fnc_adminFreeze", 4, "players", ""},
        {"Kick Player", "life_fnc_adminKick", 4, "players", ""},
        {"Give Money", "life_fnc_adminGiveMoney", 4, "players", ""},

        // Niveau 5 - Owner
        {"Ban Player", "life_fnc_adminBan", 5, "players", ""},
        {"Set Admin Level", "life_fnc_adminSetLevel", 5, "players", ""}
    };
};

class AdminActions {
    // Actions personnelles de l'admin
    commands[] = {
        {"God Mode", "life_fnc_adminGodMode", 4, "actions", ""},
        {"Markers", "life_fnc_adminMarkers", 4, "actions", ""},
        {"Debug Console", "life_fnc_adminDebugCon", 5, "actions", ""},
        {"Invisibility", "life_fnc_adminInvisibility", 4, "actions", ""},
        {"No Recoil", "life_fnc_adminNoRecoil", 4, "actions", ""}
    };
};

class ObjectManagement {
    // Gestion objets et véhicules
    commands[] = {
        {"Repair Object", "life_fnc_adminRepairObject", 3, "objects", ""},
        {"Delete Object", "life_fnc_adminDeleteObject", 4, "objects", ""},
        {"Refuel Vehicle", "life_fnc_adminRefuelVehicle", 3, "objects", ""},
        {"Flip Vehicle", "life_fnc_adminFlipVehicle", 3, "objects", ""},
        {"Spawn Vehicle", "life_fnc_adminSpawnVehicle", 4, "objects", ""}
    };
};
```

```
class ServerManagement {
    // Gestion du serveur
    commands[] = {
        {"Server Message", "life_fnc_adminServerMsg", 4, "server", ""},
        {"Restart Warning", "life_fnc_adminRestartWarning", 5, "server", ""},
        {"Server Statistics", "life_fnc_adminServerStats", 3, "server", ""},
        {"Weather Control", "life_fnc_adminWeather", 4, "server", ""},
        {"Time Control", "life_fnc_adminTime", 4, "server", ""},
        {"Cleanup Dead", "life_fnc_adminCleanup", 4, "server", ""},
        {"Admin Chat", "life_fnc_adminChat", 1, "server", ""}
    };
};
```

? Menu Déroulant (Dropdown)

Pourquoi un Menu Déroulant ?

Quand un onglet contient **plus de 8 commandes**, un menu déroulant est recommandé pour :

- Éviter l'encombrement visuel
- Maintenir une interface propre
- Faciliter la recherche de commandes

Configuration du Seuil

```
class DropdownSettings {
    // Nombre max de boutons avant de passer en dropdown
    maxButtonsBeforeDropdown = 8;

    // Hauteur du dropdown
    dropdownHeight = 0.35;  // En SafeZone

    // Position du bouton "EXÉCUTER"
    executeButtonColor[] = {0.38, 0.70, 0.89, 0.9};
};
```

Architecture Dropdown

Quand `commands[] > 8` :

1. **Combobox (Liste déroulante)**
   - Affiche toutes les commandes disponibles
   - Filtrées par niveau admin
   - Triées alphabétiquement
2. **Bouton "EXÉCUTER"**
   - Apparaît sous la combobox
   - Exécute la commande sélectionnée
   - Couleur selon le niveau de la commande

Quand `commands[] <= 8` :

- Affichage classique en boutons
- Un bouton par commande
- Pas de dropdown

Exemple d'Implémentation

```
// Dans fn_adminMenuTabChange.sqf

private _commands = [];
switch (_tabIndex) do {
    case 0: { // JOUEURS
        _commands = getArray(missionConfigFile >> "Life_AdminConfig" >> "PlayerManagement" >>
"commands");
    };
    // ... autres onglets
};

// Filtrer par niveau admin
private _availableCommands = [];
{
    private _cmdName = _x select 0;
    private _cmdFunction = _x select 1;
    private _cmdLevel = _x select 2;

    if (life_adminlevel >= _cmdLevel) then {
        _availableCommands pushBack _x;
    };
} forEach _commands;
```

```
// Vérifier si dropdown nécessaire
private _maxButtons = getNumber(missionConfigFile >> "Life_AdminConfig" >> "DropdownSettings"
>> "maxButtonsBeforeDropdown");

if (count _availableCommands > _maxButtons) then {
    // CRÉER DROPDOWN
    private _dropdown = _display ctrlCreate ["RscCombo", 29100];
    _dropdown ctrlSetPosition [0.35 * safezoneW + safezoneX, 0.35 * safezoneH + safezoneY,
0.25 * safezoneW, 0.03 * safezoneH];
    _dropdown ctrlCommit 0;

    // Remplir dropdown
    {
        private _cmdName = _x select 0;
        private _index = _dropdown lbAdd _cmdName;
        _dropdown lbSetData [_index, _x select 1]; // Stocker fonction
        _dropdown lbSetValue [_index, _x select 2]; // Stocker niveau
    } forEach _availableCommands;

    // Bouton EXÉCUTER
    private _executeBtn = _display ctrlCreate ["RscButton", 29101];
    _executeBtn ctrlSetText "EXÉCUTER";
    _executeBtn ctrlSetPosition [0.35 * safezoneW + safezoneX, 0.39 * safezoneH + safezoneY,
0.25 * safezoneW, 0.03 * safezoneH];
    _executeBtn ctrlCommit 0;

    _executeBtn ctrlAddEventHandler ["ButtonClick", {
        private _display = findDisplay 29000;
        private _dropdown = _display displayCtrl 29100;
        private _selectedIndex = lbCurSel _dropdown;

        if (_selectedIndex >= 0) then {
            private _function = _dropdown lbData _selectedIndex;
            call compile format["call %1;", _function];
        };
    }];

} else {
    // CRÉER BOUTONS CLASSIQUES
    private _yOffset = 0;
```

```
    {
        private _btn = _display ctrlCreate ["RscButton", 29100 + _forEachIndex];
        // ... création bouton classique
        _yOffset = _yOffset + 0.035;
    } forEach _availableCommands;
};
```

? Personnalisation Avancée

Icônes Personnalisées

```
commands[] = {
    {
        "Heal Player",
        "life_fnc_adminHeal",
        3,
        "players",
        "\A3\ui_f\data\IGUI\Cfg\Actions\heal_ca.paa"  // Icône custom
    }
};
```

Tooltips (Infobulles)

```
class Tooltips {
    enabled = 1;  // Activer les tooltips

    // Format: {"fonction", "Description"}
    descriptions[] = {
        {"life_fnc_adminHeal", "Soigner complètement le joueur sélectionné"},
        {"life_fnc_adminFreeze", "Geler/Dégeler le joueur ciblé"},
        {"life_fnc_adminBan", "Bannir définitivement le joueur du serveur"}
    };
};
```

Sons Personnalisés

```
class Sounds {
    // Sons lors d'actions
    onButtonClick = "click";
    onCommandExecute = "FD_Finish_F";
    onError = "FD_CP_Not_Clear_F";
    onSuccess = "FD_Finish_F";
};
```

? Exemples de Configuration

Configuration Minimaliste (4 onglets, peu de commandes)

```
class Life_AdminConfig {
    class AdminLevels {
        levels[] = {"Joueur", "Support", "Modérateur", "Admin", "Super Admin", "Owner"};
    };

    class DropdownSettings {
        maxButtonsBeforeDropdown = 10;  // Seuil élevé
    };

    class PlayerManagement {
        commands[] = {
            {"Spectate", "life_fnc_adminSpectate", 3, "players"},
            {"TP to Player", "life_fnc_adminTeleport", 3, "players"},
            {"Freeze", "life_fnc_adminFreeze", 4, "players"}
        };
    };

    // ... autres classes minimalistes
};
```

Configuration Avancée (5+ onglets, beaucoup de commandes)

```
class Life_AdminConfig {
    class AdminLevels {
        levels[] = {
            "Joueur",           // 0
            "Trial Support",    // 1
            "Support",          // 2
            "Modérateur",       // 3
            "Admin",            // 4
            "Super Admin",      // 5
            "Head Admin",       // 6
            "Owner"             // 7
        };
    };

    class DropdownSettings {
        maxButtonsBeforeDropdown = 6;  // Seuil bas = dropdown plus fréquent
        dropdownHeight = 0.4;
    };

    class Tabs {
        count = 6;
        tabs[] = {
            {"JOUEURS", "players", "\A3\...", {0.38, 0.70, 0.89, 0.7}},
            {"ACTIONS", "actions", "\A3\...", {0.8, 0.5, 0.2, 0.7}},
            {"VÉHICULES", "vehicles", "\A3\...", {0.3, 0.7, 0.3, 0.7}},
            {"SERVEUR", "server", "\A3\...", {0.8, 0.2, 0.2, 0.7}},
            {"UTILITAIRES", "utilities", "\A3\...", {0.5, 0.5, 0.9, 0.7}},
            {"LOGS", "logs", "\A3\...", {0.4, 0.4, 0.4, 0.7}}
        };
    };

    class PlayerManagement {
        commands[] = {
            // 15+ commandes joueurs
            {"Get ID", "life_fnc_adminGetID", 1, "players"},
            {"Query", "life_fnc_adminQuery", 1, "players"},
            {"Spectate", "life_fnc_adminSpectate", 3, "players"},
            {"TP to", "life_fnc_adminTeleport", 3, "players"},
            {"Heal", "life_fnc_adminHeal", 3, "players"},
            {"Revive", "life_fnc_adminRevive", 3, "players"},
```

```
            {"TP Here", "life_fnc_adminTpHere", 4, "players"},
            {"Freeze", "life_fnc_adminFreeze", 4, "players"},
            {"Kick", "life_fnc_adminKick", 4, "players"},
            {"Give Money", "life_fnc_adminGiveMoney", 4, "players"},
            {"Strip Weapons", "life_fnc_adminStripWeapons", 4, "players"},
            {"Ban Temp", "life_fnc_adminBanTemp", 5, "players"},
            {"Ban Perm", "life_fnc_adminBan", 6, "players"},
            {"Unban", "life_fnc_adminUnban", 6, "players"},
            {"Set Admin", "life_fnc_adminSetLevel", 7, "players"}
        };
    };

    // ... autres classes avec beaucoup de commandes
};
```

?? Modification Dynamique

Ajouter une Commande

1. Ouvrir `config/Config_Admin.hpp`
2. Trouver la classe correspondante (`PlayerManagement`, `AdminActions`, etc.)
3. Ajouter une ligne dans `commands[]`:

```
{"Ma Commande", "life_fnc_maFonction", 3, "players"}
```

4. Recompiler la mission
5. La commande apparaît automatiquement dans le bon onglet !

Changer le Niveau Requis

```
// AVANT
{"Freeze Player", "life_fnc_adminFreeze", 4, "players"}

// APRÈS (accessible dès niveau 3)
{"Freeze Player", "life_fnc_adminFreeze", 3, "players"}
```

Déplacer une Commande d'Onglet

```
// AVANT (dans l'onglet ACTIONS)
{"God Mode", "life_fnc_adminGodMode", 4, "actions"}

// APRÈS (dans l'onglet UTILITAIRES)
{"God Mode", "life_fnc_adminGodMode", 4, "utilities"}
```

? Tests et Validation

Checklist de Validation

```
□ Tous les niveaux admin ont des couleurs distinctes
□ Les onglets sont bien nommés et dans l'ordre souhaité
□ Chaque commande est dans le bon onglet
□ Les niveaux admin sont corrects (pas de niveau 1 avec debug)
□ Le seuil dropdown est adapté au nombre de commandes
□ Les icônes s'affichent correctement
□ Les tooltips sont clairs et utiles
□ Pas de doublons de commandes
□ Toutes les fonctions existent et sont déclarées
```

Tests In-Game

```
// Tester avec différents niveaux
life_adminlevel = 1;  // Support
createDialog "life_admin_menu_new";

life_adminlevel = 3;  // Admin
createDialog "life_admin_menu_new";

life_adminlevel = 5;  // Owner
createDialog "life_admin_menu_new";
```

? Référence Rapide

Commandes Config

Propriété	Type	Valeurs	Description
`count`	NUMBER	1-10	Nombre d'onglets
`maxButtonsBeforeDropdown`	NUMBER	5-15	Seuil dropdown
`levels[]`	ARRAY	Strings	Noms des niveaux admin
`commands[]`	ARRAY	Arrays	Liste des commandes
`tabs[]`	ARRAY	Arrays	Configuration onglets

Format Couleur

```
{R, G, B, Alpha}
// R, G, B: 0.0 à 1.0
// Alpha: 0.0 (transparent) à 1.0 (opaque)
```

Icônes Arma 3

Chemin	Description
`\A3\ui_f\data\GUI\Rsc\RscDisplayArsenal\face_ca.paa`	Visage (joueurs)
`\A3\ui_f\data\IGUI\Cfg\Actions\settimer_ca.paa`	Horloge
`\A3\ui_f\data\IGUI\Cfg\Actions\getindriver_ca.paa`	Véhicule
`\A3\ui_f\data\IGUI\Cfg\Actions\repair_ca.paa`	Réparer
`\A3\ui_f\data\IGUI\Cfg\Actions\heal_ca.paa`	Soigner

? Conclusion

Avec ce système de configuration :

- □ **Facile à modifier** : Tout est dans un seul fichier
- □ **Flexible** : Supporte 1 à 10+ onglets
- □ **Scalable** : Dropdown automatique si trop de commandes
- □ **Personnalisable** : Couleurs, icônes, tooltips
- □ **Maintenable** : Pas besoin de toucher au code SQF

Prochaines étapes :

1. Personnaliser `Config_Admin.hpp` selon vos besoins
2. Adapter `fn_adminMenuTabChange.sqf` pour lire cette config
3. Implémenter le système de dropdown
4. Tester avec différents niveaux admin

Auteur: ProtoMehka **Version**: 2.1 **Date**: 2025-10-10