

Interface

Code

```
enum OrderGroupID
{
    GROUP_FIREDEPT,
    GROUP_POLICE,
    GROUP_AMBULANCE,
    GROUP_TEC
};
```

Code

```

namespace                                     ScriptInterface
{
    void                                         CloseObjectives();

    void     SetMission(const           char*        mission_);
    void     DisableMap(bool            sleep_       = true);
    void     DisableRadar(bool           sleep_      = true);
    void     DisableNavigator(bool        sleep_     = true);
    void     DisableInfoBar(bool          sleep_    = true);
    void     DisableVehicleBrowser(bool    sleep_   = true);

    void                                         ShowMap();
    void                                         HideMap();
    void                                         ShowRadar();
    void                                         OpenRadar();
    void                                         HideRadar();
    void                                         ShowNavigator();
    void                                         OpenNavigator();
    void                                         HideNavigator();
    void                                         ShowInfoBar();
    void                                         OpenInfoBar();
    void                                         HideInfoBar();
    void                                         ShowVehicleBrowser();
    void                                         HideVehicleBrowser();

    void                                         ShowBriefing();
    void                                         HideBriefing();
    void                                         ShowTips();
    void                                         OpenTips();
    void                                         HideTips();
    void                                         ShowActionBar();
    void                                         OpenActionBar();
    void                                         HideActionBar();
    void                                         ShowObjectives();
    void                                         HideObjectives();

    void                                         OpenObjectives();

    void                                         EnableTutorialMode();
    void                                         DisableTutorialMode();

    void                                         ShowVisibleTutorialOverlay();
    void                                         ShowInvisibleTutorialOverlay();
    void                                         HideTutorialOverlay();
    void                                         HideInterface();
    void                                         ResumeFromHideInterface();

void showTutorialMessage(const char* MessageID, const char* VoiceMessageURL, float y=-1.0f);           IsTutorialMessageOpen();

void showTutorialInstruction(const char* instructionID, int soundRef=1, float y=1.0f);                  if (soundRef > -1) ShowTutorialMessage();
void closeTutorialInstruction(int soundRef=-1); // wird durch ShowTutorialMessage aufgerufen
void                                         ShowMainMenu();

void     ShowNotice(const           char        *textName_);
void     HideNotice();
bool    NoticeIsHidden();
bool    DidUserHitOK();

void     BlinkNavigatorOpenCloseButton(bool      Blink_);
void     BlinkNavigatorObjectivesButton(bool      Blink_);
void     BlinkObjectivesOpenCloseButton(bool      Blink_);
void     BlinkNavigatorFireDepartmentButton(bool    Blink_);
void     EnableNavigatorFireDepartmentButton(bool  enable_);
void     BlinkNavigatorAmbulanceButton(bool      Blink_);
void     EnableNavigatorAmbulanceButton(bool      enable_);

https://em-hub.de/lexicon/entry/23-interface/
void     BlinkSendVehicleButton(bool      Blink_?);
void     BlinkSendVehicleTargetButton(bool      Blink_);
void     BlinkFireDepartmentVehicle(const char  *VehicleName_, bool      Blink_);
void     BlinkAmbulanceVehicle(const char  *VehicleName_, bool      Blink_);

```

[Alles anzeigen](#)