Includes for San Andreas Multiplayer Plus
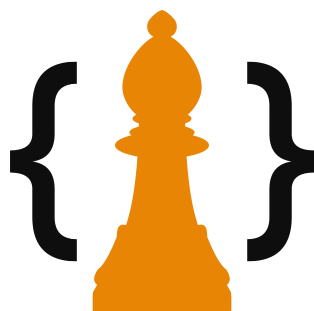# Version 1.5, Documentation

Ivan Kmeťo

September 2021

Contents

1 Changes in ISAMPP 1.5

ISAMPP Library:
- New definitions for body parts *(i_bodyparts.inc)*
- New definitions for player bones *(i_boneids.inc)*
- New definitions for camera mode *(i_cammode.inc)*
- New definitions for vehicle components *(i_carmods.inc)*
- Added remaining 9 GameText styles *(i_textstyle.inc)*
- ICON_OGLOC0 renamed to ICON_MCS *(i_iconids.inc)*
- SKIN_BBFAT renamed to SKIN_BB *(i_skinids.inc)*
- Updated documentation (external links to SA-MP Wiki were removed)
- Expanded section Useful Resources in documentation

ISAMPP Sandbox Game Mode:
- Player will get weapon on spawn
- Added test dummy Actor
- Implemented function OnPlayerGiveDamageActor
- Player will spawn with green hat (for testing Bone IDs)
- Added new command for showing active Camera Mode
- Expanded commands for testing GameText styles
- Added position coordinates variables for AddPlayerClass function
- Code formatting cleanup

Changelogs for previous versions of ISAMPP are located in folder "docs/changelogs"

2   Introduction

San Andreas Multiplayer Includes Library+ or ISAMPP *(Includes for San Andreas Multiplayer Plus)* is library of include files for San Andreas Multiplayer Mod SA-MP. Purpose of ISAMPP is to make development of game modes for SA-MP much easier. SA-MP by default uses in most cases numeric identifiers which are hard to remember and without enough experience you have to use SA-MP wiki a lot. Word identifiers are much more easier to remember, they make much more sense (especially if you are familiar with modding the game Grand Theft Auto: San Andreas) and may also have positive impact on your workflow. Except re-definitions of numeric identifiers, ISAMPP also contains other useful libraries - such as library of locations with coordinates, interior identifiers and location names - or few related custom stock functions. ISAMPP is available with sandbox-styled game mode in which you can see everything implemented.

ISAMPP is not part of San Andreas Multiplayer mod (SA-MP) and it is not affiliated with Rockstar Games, Rockstar North or Take-Two Interactive Software Inc.
Grand Theft Auto and Grand Theft Auto: San Andreas are registered trademarks of Take-Two Interactive Software Inc.

ISAMPP versions 1.3 or newer should be considered public domain.
https://creativecommons.org/publicdomain/zero/1.0/

Older versions were published under MIT License.
http://opensource.org/licenses/mit-license

3   Installation

Copy contents of *include* folder to *"[SA-MP Server folder]/include"* and also to *"include"* folder for Pawno (by default *"[SA-MP Server folder]/pawno/include"*).

In your game mode file you can include ISAMPP header for all its contents:

```
#include <i_sampp>
```

or include header files as you wish separately:

```
#include <i_sampp/i_bodyparts>
#include <i_sampp/i_boneids>
#include <i_sampp/i_cammode>
#include <i_sampp/i_carcols>
#include <i_sampp/i_carmods>
#include <i_sampp/i_colorlist>
#include <i_sampp/i_iconids>
#include <i_sampp/i_locationids>
#include <i_sampp/i_paintjob>
#include <i_sampp/i_pickupids>
#include <i_sampp/i_skinids>
#include <i_sampp/i_textstyle>
#include <i_sampp/i_vehhealth>
#include <i_sampp/i_vehids>
#include <i_sampp/i_weaponids>
#include <i_sampp/i_weatherids>
```

If you wish to run included sandbox game mode, you have to add file testmpp.pwn to your *"gamemodes"* folder and compile it. Alternatively, you can use pre-compiled file testmpp.amx.

4 Includes

Every ISAMPP include file starts with prefix *i_*. Please keep in mind that ISAMPP may not be the only set of include files using this prefix.

bodyparts - List of available player/npc body part identifiers
boneids - List of player bone identifiers
cammode - List of known camera modes
carcols - List of definitions for all available vehicle colors
carmods - List of all available components for vehicle customization
colorlist - List of color definitions
iconids - List of map icon identifiers
locationids - List of location definitions with coordinates, interior identifiers and names
paintjob - List of definitions for all available vehicle paintjobs
pickupids - List of definitions for pickup identifiers
skinids - List of definitions for character skin/model identifiers
textstyle - List of definitions for GameText styles
vehhealth - Vehicle Health configurations
vehids - List of definitions for all available vehicles
weaponids - List of definitions for weapon identifiers, sorted by weapon slot IDs
weatherids - List of definitions for weather identifiers

5 Body Parts

Body part identifiers are included as macro defines with prefix *BODYPART_*. Body parts can be used in functions *OnPlayerGiveDamage*, *OnPlayerTakeDamage* and *OnPlayerGiveDamageActor*. It is unknown if IDs 0, 1 and 2 have any use. You can test this feature in sandbox game mode by shooting Actor NPC.

Identifier	Value	Body Part
BODYPART_TORSO	3	Torso
BODYPART_GROIN	4	Groin
BODYPART_LEFTARM	5	Left Arm
BODYPART_RIGHTARM	6	Right Arm
BODYPART_LEFTLEG	7	Left Leg
BODYPART_RIGHTLEG	8	Right Leg
BODYPART_HEAD	9	Head

Resources: https://open.mp/docs/scripting/resources/bodyparts

6 Bone Identifiers

Bone identifiers with prefix *BONE_* can be used for attaching objects to specific parts of player model/skin with function *SetPlayerAttachedObject*.

Identifier	Value	Bone
BONE_SPINE	1	Spine
BONE_HEAD	2	Head
BONE_LEFTUPPERARM	3	Left Upper Arm
BONE_RIGHTUPPERARM	4	Right Upper Arm
BONE_LEFTHAND	5	Left Hand
BONE_RIGHTHAND	6	Right Hand
BONE_LEFTTHIGH	7	Left Thigh
BONE_RIGHTTHIGH	8	Right Thigh
BONE_LEFTFOOT	9	Left Foot
BONE_RIGHTFOOT	10	Right Foot
BONE_RIGHTCALF	11	Right Calf
BONE_LEFTCALF	12	Left Calf
BONE_LEFTFOREARM	13	Left Forearm
BONE_RIGHTFOREARM	14	Right Forearm
BONE_LEFTSHOULDER	15	Left Clavicle (shoulder)
BONE_RIGHTSHOULDER	16	Right Clavicle (shoulder)
BONE_NECK	17	Neck
BONE_JAW	18	Jaw

Example of implementation:

```
public OnPlayerSpawn(playerid)
{
    //Give player green hat on spawn
    SetPlayerAttachedObject(playerid, BONE_LEFTUPPERARM, 19487, 2, 0.101,
    -0.0, 0.0, 5.50, 84.60, 83.7, 1.0, 1.0, 1.0, 0xFF00FF00);

    return 1;
}
```

Resources: https://open.mp/docs/scripting/resources/boneid

7 Camera Modes

List of known camera modes to be used as returned value of function *GetPlayerCameraMode(playerid)*.
In ISAMPP sandbox game mode you can output more information about active camera mode with command
/cameramode

Identifier	Value	Description
CAMMODE_TRAIN	3	Train/tram camera.
CAMMODE_FOLLOWPED	4	Follow ped (normal behind player camera).
CAMMODE_SNIPER	7	Sniper rifle aiming.
CAMMODE_ROCKETAIM	8	Rocket launcher aiming.
CAMMODE_FIXED	15	Fixed camera (non-moving).
CAMMODE_VEHICLEFRONT	16	Vehicle front camera, bike side camera.
CAMMODE_VEHICLEDEFAULT	18	Normal car/helicopter/airplane camera, several variable distances.
CAMMODE_BOATDEFAULT	22	Normal boat camera.
CAMMODE_WEAPONAIM	46	Weapon aiming camera.
CAMMODE_ROCKETAIM2	51	Heat-seeking Rocket Launcher aiming camera.
CAMMODE_WEAPONAIM2	53	Aiming any other weapon.
CAMMODE_VEHICLEPASSENGER	55	Vehicle passenger drive-by camera.
CAMMODE_HELICHASE	56	Chase camera: helicopter/bird view.
CAMMODE_GROUNDCHASE	57	Chase camera: ground camera, zooms in very quickly.
CAMMODE_FLYBYCHASE	58	Chase camera: horizontal flyby past vehicle.
CAMMODE_GROUNDCHASE2	59	Chase camera (air vehicles only): looking up to the air vehicle.
CAMMODE_FLYBYCHASE2	62	Chase camera (air vehicles only): vertical flyby past air vehicle.
CAMMODE_FLYBYCHASE3	63	Chase camera (air vehicles only): horizontal flyby past air vehicle.
CAMMODE_PILOTCHASE	64	Chase camera (air vehicles only): camera focused on pilot.

Resources: https://open.mp/docs/scripting/resources/cameramodes

8 Color List

Color List contains definitions of some few common colors in two formats. Primary or Main color definitions
have prefix *COLOR_* and are defined in 0xRRGGBBAA format. Secondary or String color definitions with
prefix *SCOL_* are defined in more common hex format "{RRGGBB}" without the alpha color transparency
values. The difference of these two formats can be explained by this example:

```
SendClientMessage(playerid, COLOR_RED, "Hello "SCOL_BLUE"World");
```

where the output will print in game client message box string of text in this format: **Hello World**

Secondary or String colors can be used simply in strings while Primary/Main colors would be used mainly
as parameters in SA-MP functions. To see how every color looks in game, use in included sandbox mode for
ISAMPP command */defcols[0-14]* or */stringcols*. Screenshot of color strings printed out in client message
box is located in *"docs/images/i_colorlist.png"*.

Resources: https://open.mp/docs/scripting/resources/colorslist

9 Map Icon Identifiers

There are in total 64 map icons included as macro defines with prefix *ICON_*.

Resources: https://open.mp/docs/scripting/resources/mapicons

10 Location Identifiers

Locations, in-game interiors and exteriors, are stored in file *i_locationids.inc* as set of macro definitions for identifiers and three arrays. Array *locCoords* is storing float values of XYZ position on the map and float value of PlayerFacingAngle. By default, your PlayerInterior value is set to 0 whenever you are "outside" on the map. When player enters the interior, in order to load the interior you have to change PlayerInterior value accordingly. This is what array *locID* includes. Third array, *locName*, has defined strings of names of particular locations so this information can be displayed to player if needed.

Please note that if you wish to add new location to this list, you have to do it in correct order. Not every location has proper collision boxes because it was used only in cutscene, so in certain interiors you may fall through ground.

Resources: https://open.mp/docs/scripting/resources/interiorids

11 Pickup Identifiers

Pickup identifiers are included as macro defines with prefix *PICKUP_*. Every pickup defined in ISAMPP is placed on map in ISAMPP sandbox game mode.

Resources: https://open.mp/docs/scripting/resources/pickupids

12 Skin/Playermodel Identifiers

Skins are included as macro defines with prefix *SKIN_*, followed by name of model in game. Luckily, Rockstar North did great job in naming their models, so it's pretty simple to know what name represents which skin in game. Usually, with exception of few character models *(mainly main story ones and special characters)* where the name is obvious *(f.e. emmet = Emmet, etc.)*, character models are named in this or similar format: prefix *(f.e. if is located only in certain area)*, race, gender, age, suffix. In this context BMYRI = **B**lack, **M**ale, **Y**oung, **RI**ch. SBFYS = (**S**an **F**ransisco) **B**lack, **F**emale, **Y**oung, **S**treet, etc. This format is very helpful in contrast with number identifiers SA-MP uses by default.

Resources: https://open.mp/docs/scripting/resources/skins

13 GameText Styles

List of definitions for SA-MP function *GameTextForPlayer(playerid, const string[], time, **style**)*. Some styles might not work properly or could crash the game.

Identifier	Value	Notes
GMTEXT_STYLE_PRICEDOWN	0	Appears for 9 seconds regardless of time setting.
GMTEXT_STYLE_RPRICEDOWN	1	Fades out after 8 seconds, regardless of time set.
GMTEXT_STYLE_SA	2	San Andreas main font. Does not disappear until player respawns.
GMTEXT_STYLE_SLIM	3	San Andreas specific font.
GMTEXT_STYLE_SLIM2	4	San Andreas specific font.
GMTEXT_STYLE_SLIMW	5	Displays for 3 seconds, regardless of what time you set.
GMTEXT_STYLE_BPRICEDOWN	6	Blue Pricedown font in middle of screen.
GMTEXT_STYLE_VEHNAME	7	Based on SA vehicle names. Added by fixes.inc
GMTEXT_STYLE_LOCATION	8	Based on SA location names. Added by fixes.inc
GMTEXT_STYLE_RADIO	9	Based on SA radio station names (selected). Added by fixes.inc
GMTEXT_STYLE_RADIOW	10	Based on SA radio station names (switching). Added by fixes.inc
GMTEXT_STYLE_PMONEY	11	Based on SA positive money. Added by fixes.inc
GMTEXT_STYLE_NMONEY	12	Based on SA negative money. Added by fixes.inc
GMTEXT_STYLE_STUNT	13	Based on SA stunt bonuses. Added by fixes.inc
GMTEXT_STYLE_CLOCK	14	Based on SA in-game clock. Added by fixes.inc
GMTEXT_STYLE_NOTIFICATION	15	Based on SA notification popup. Added by fixes.inc

Resources: https://open.mp/docs/scripting/resources/gametextstyles

14 Vehicle Components

List of all available components for vehicle customization, starting with prefix *CARMOD_*
Can be used with function *AddVehicleComponent(vehicleid, **componentid**);*

Resources: https://open.mp/docs/scripting/resources/carcomponentid

15 Vehicle Identifiers

Vehicle identifiers are included as macro defines with prefix *VEH_* followed by the name of vehicle in game with capital letters *(motorcycles, bicycles, boats, planes, helicopters, trailers, trains, etc. are included in this format as well)*. For better orientation in list, defines are sorted by vehicle type.

Resources: https://open.mp/docs/scripting/resources/vehicleid

16 Vehicle Health

Vehicle Health configurations are included as macro defines with prefix *VEH_HEALTH_* followed by the desired pre-defined identifier. These values are only related to engine condition and do not change visual damage of vehicle model.

Identifier	Exact Value	Description
VEH_HEALTH_FULL	1000	Full vehicle health
VEH_HEALTH_FULL_LOW	650	Lowest value for undamaged vehicle
VEH_HEALTH_WHITESMOKE	649	White smoke from engine
VEH_HEALTH_WHITESMOKE_LOW	550	Lowest value for white engine smoke
VEH_HEALTH_GREYSMOKE	549	Grey smoke from engine
VEH_HEALTH_GREYSMOKE_LOW	390	Lowest value for grey engine smoke
VEH_HEALTH_BLACKSMOKE	389	Black smoke from engine
VEH_HEALTH_BLACKSMOKE_LOW	250	Lowest value for black engine smoke
VEH_HEALTH_ONFIRE	249	Sets car on fire

You can test this in ISAMPP sandbox game mode with command **/setvehiclehealth [ID]** while being in any vehicle.

Resources: https://open.mp/docs/scripting/resources/vehiclehealth

17 Vehicle Colors

List of all available colors for vehicles in game. Color names using prefix *CARCOL_SAMP_* are supported only in SA-MP version 0.3x.

Resources: https://open.mp/docs/scripting/resources/vehiclecolorid

18 Vehicle Paintjobs

List of all available paintjobs for vehicles in game. Use *PAINTJOB_REMOVE* to remove applied paintjob.

Example of implementation:

```
AddStaticVehicle(VEH_CAMPER, 425.7991, 2493.3472, 16.5794, 180, CARCOL_WHITE, CARCOL_WHITE);
ChangeVehiclePaintjob(vehicleid, PAINTJOB_CAMPER_TRUTH);
```

Resources: https://open.mp/docs/scripting/resources/paintjobs

19 Weapon Identifiers

Weapon identifiers are included as macro defines with prefix *WEAP_*. In ISAMPP sandbox game mode weapon pickups are fully functional. Every weapon in game can be used only for certain weapon slot and weapon identifiers are ordered in file *i_weaponids.inc* by weapon slot number.

Resources: https://open.mp/docs/scripting/resources/weaponids

20 Weather Identifiers

Weather identifiers are included as macro defines with prefix *WEATHER_*. In ISAMPP sandbox game mode you can test weather settings with command **/w [ID]**

Resources: https://open.mp/docs/scripting/resources/weatherid

21 Custom Functions

ISAMPP uses various stock functions which may be useful in creating your own game modes for SA-MP or simply for debugging purposes. These stock functions are defined in *i_sampp.inc* file.

21.1 isampp_console_printversion()

Outputs ISAMPP version to server console.

Example of implementation:

```
main() {
    isampp_console_printversion();
}
```

21.2 MppTeleport(playerid, locationid)

Teleports player to desired location passed as parameter 'locationid'.

Example of implementation:

```
if (strcmp("/tp barbershop", cmdtext, true, 20) == 0) {
  MppTeleport(playerid, LOC_BARBERSHOP);
  return 1;
}
```

21.3 MppTeleportEx(playerid, locationid, pstringcolor)

Same as MppTeleport, plus outputs location name to in-game chat window.
Parameter 'pstringcolor' must be in hexadecimal format 0xRRGGBBAA.

Example of implementation:

```
if (strcmp("/tp barbershop", cmdtext, true, 20) == 0) {
    MppTeleportEx(playerid, LOC_BARBERSHOP, COLOR_LIMEGREEN);
    return 1;
}
```

21.4 MppTeleportToCoords(playerid, x, y, z, interiorid, facingangle)

Teleports player to specified xyz coordinates, supplied with interior identifier and player facing angle.

Example of implementation:

```
if (strcmp("/tpcoord", cmdtext, true, 20) == 0) {
    MppTeleportToCoords(playerid, 49.4172, 2512.4282, 16.4844, 0, 272);
    return 1;
}
```

21.5   MppShowPlayerPosition(playerid, pstringcolor)

Outputs current player location coordinates, interior identifier, facing angle and player camera position co-ordinates to in-game chat window.
Parameter 'pstringcolor' must be in hexadecimal format 0xRRGGBBAA.
This function might not work properly if player is in a vehicle.

Example of implementation:

```
if (strcmp("/showplayerpos", cmdtext, true, 15) == 0) {
    MppShowPlayerPosition(playerid, COLOR_LIGHTRED);
    return 1;
}
```

21.6   MppShowVehicleInfo(playerid, vehicleid, pstringcolor)

Outputs ID, model, health, position and rotation of vehicle in which is player currently sitting to in-game chat window.
Parameter 'pstringcolor' must be in hexadecimal format 0xRRGGBBAA.
Note: You must pass vehicleid parameter if you want to get model name else function returns '0 Unknown'.

Example of implementation:

```
new VehicleModelID = 0;

public OnPlayerCommandText(playerid, cmdtext[])
{
    if (strcmp("/showvehicleinfo", cmdtext, true, 15) == 0) {
        MppShowVehicleInfo(playerid, VehicleModelID, COLOR_LIGHTBLUE);
        return 1;
    }

    return 0;
}

public OnPlayerEnterVehicle(playerid, vehicleid, ispassenger)
{
    VehicleModelID = GetVehicleModel(vehicleid);

    return 1;
}

public OnPlayerExitVehicle(playerid, vehicleid)
{
    VehicleModelID = 0;

    return 1;
}
```

22 Useful Resources

SA-MP has in certain cases various text defines replacing numeric identifiers by default.

22.1 Animations List

Read More: https://open.mp/docs/scripting/resources/animations

22.2 Bullet Hit Types

To be used with function *OnPlayerWeaponShot*.

Value	Name
0	BULLET_HIT_TYPE_NONE
1	BULLET_HIT_TYPE_PLAYER
2	BULLET_HIT_TYPE_VEHICLE
3	BULLET_HIT_TYPE_OBJECT
4	BULLET_HIT_TYPE_PLAYER_OBJECT

Read More: https://open.mp/docs/scripting/resources/bullethittypes

22.3 Camera Cut Styles

Camera Cut Styles to be used with functions *SetPlayerCameraLookAt*, *InterpolateCameraPos* and *InterpolateCameraLookAt*.

Value	Name
1	CAMERA_MOVE
2	CAMERA_CUT

Read More: https://open.mp/docs/scripting/resources/cameracutstyles

22.4 Component Slots

All available component slots for vehicle customization.
To be used with function *GetVehicleComponentInSlot(vehicleid, **slot**);*

Value	Name
0	CARMODTYPE_SPOILER
1	CARMODTYPE_HOOD
2	CARMODTYPE_ROOF
3	CARMODTYPE_SIDESKIRT
4	CARMODTYPE_LAMPS
5	CARMODTYPE_NITRO
6	CARMODTYPE_EXHAUST
7	CARMODTYPE_WHEELS
8	CARMODTYPE_STEREO
9	CARMODTYPE_HYDRAULICS
10	CARMODTYPE_FRONT_BUMPER
11	CARMODTYPE_REAR_BUMPER
12	CARMODTYPE_VENT_RIGHT
13	CARMODTYPE_VENT_LEFT

Read More: https://open.mp/docs/scripting/resources/Componentslots

22.5 Connection Status

Connection status returned as value of function *NetStats_ConnectionStatus(playerid);*

Value	Name	Description
0	NO_ACTION	N/A
1	DISCONNECT_ASAP	playerid exists, OnPlayerDisconnect called
2	DISCONNECT_ASAP_SILENTLY	N/A
3	DISCONNECT_ON_NO_ACK	N/A
4	REQUESTED_CONNECTION	Connection request cookie sent
5	HANDLING_CONNECTION_REQUEST	N/A
6	UNVERIFIED_SENDER	N/A
7	SET_ENCRYPTION_ON_MULTIPLE_16_BYTE_PACKET	N/A
8	CONNECTED	playerid is connected to the server

Read More: https://open.mp/docs/scripting/resources/connectionstatus

22.6 Constants List

Pre-defined constants and identifiers in various SA-MP header files.

Read More: https://open.mp/docs/scripting/resources/constants

22.7 Dialog Styles

Styles of dialog windows in SA-MP, to be used with function *ShowPlayerDialog.*

Style	Name
Style 0	DIALOG_STYLE_MSGBOX
Style 1	DIALOG_STYLE_INPUT
Style 2	DIALOG_STYLE_LIST
Style 3	DIALOG_STYLE_PASSWORD
Style 4	DIALOG_STYLE_TABLIST
Style 5	DIALOG_STYLE_TABLIST_HEADERS

Read More: https://open.mp/docs/scripting/resources/dialogstyles

22.8 Marker Modes

To be used with function *ShowPlayerMarkers(**mode**);*

Value	Name
0	PLAYER_MARKERS_MODE_OFF
1	PLAYER_MARKERS_MODE_GLOBAL
2	PLAYER_MARKERS_MODE_STREAMED

Read More: https://open.mp/docs/scripting/resources/markermodes