

# SA-MP Includes Library+ 1.3 Documentation

Ivan Kmeřo

October 2020

# Contents

<b>1</b>	<b>Changes in ISAMPP 1.3</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
<b>3</b>	<b>Installation</b>	<b>4</b>
<b>4</b>	<b>Includes</b>	<b>5</b>
<b>5</b>	<b>Color List</b>	<b>6</b>
<b>6</b>	<b>Icon Identifiers</b>	<b>6</b>
<b>7</b>	<b>Location Identifiers</b>	<b>6</b>
<b>8</b>	<b>Pickup Identifiers</b>	<b>6</b>
<b>9</b>	<b>Skin/Playermodel Identifiers</b>	<b>6</b>
<b>10</b>	<b>GameText Styles</b>	<b>6</b>
<b>11</b>	<b>Vehicle Identifiers</b>	<b>7</b>
<b>12</b>	<b>Vehicle Health</b>	<b>7</b>
<b>13</b>	<b>Vehicle Colors</b>	<b>7</b>
<b>14</b>	<b>Vehicle Paintjobs</b>	<b>7</b>
<b>15</b>	<b>Weapon Identifiers</b>	<b>7</b>
<b>16</b>	<b>Weather Identifiers</b>	<b>7</b>
<b>17</b>	<b>Custom Functions</b>	<b>8</b>
17.1	isampp_console_printversion() . . . . .	8
17.2	MppTeleport(playerid, locationid) . . . . .	8
17.3	MppTeleportEx(playerid, locationid, pstringcolor) . . . . .	8
17.4	MppTeleportToCoords(playerid, x, y, z, interiorid, facingangle) . . . . .	8
17.5	MppShowPlayerPosition(playerid, pstringcolor) . . . . .	9
17.6	MppShowVehicleInfo(playerid, vehicleid, pstringcolor) . . . . .	9
<b>18</b>	<b>Useful Resources</b>	<b>10</b>
18.1	Marker Modes . . . . .	10

# 1 Changes in ISAMPP 1.3

## License:

- ISAMPP versions 1.3 or newer are published as work under CC0 1.0 Universal (CC0 1.0) and should be considered public domain.

## ISAMPP Library:

- Changed format of scripting functions (*i\_sampp.inc*)
  - ISAMPP\_TELEPORT: MppTeleport
  - ISAMPP\_TELEPORTEX: MppTeleportEx
  - ISAMPP\_TELEPORT\_TO\_COORDS: MppTeleportToCoords
  - ISAMPP\_SHOWPLAYERPOSITION: MppShowPlayerPosition
  - ISAMPP\_SHOWVEHICLEINFO: MppShowVehicleInfo
- New definitions for vehicle paintjobs (*i\_paintjob.inc*)
- Removed pickup: *PICKUP\_ALTCHAINSAW*
- Updated documentation

## ISAMPP Sandbox Game Mode:

- Changes related to new format of defined scripting functions
- testinc.pwn renamed to testmpp.pwn
- Added compiled file to repository (testmpp.amx)

*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. I believe that 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_colorlist>
#include <i_sampp/i_iconids>
#include <i_sampp/i_locationids>
#include <i_sampp/i_pickupids>
#include <i_sampp/i_skinids>
#include <i_sampp/i_textstyle>
#include <i_sampp/i_vehids>
#include <i_sampp/i_vehhealth>
#include <i_sampp/i_carcols>
#include <i_sampp/i_paintjob>
#include <i_sampp/i_weaponids>
#include <i_sampp/i_weatherids>
```

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

## 4 Includes

Every ISAMPP include file starts with prefix *i\_*. Please keep in mind that ISAMPP may not be the only include with this prefix.

*colorlist* - List of color definitions

*iconids* - List of icon identifiers

*locationids* - List of location definitions with coordinates, interior identifiers and names

*pickupids* - List of definitions for pickup identifiers

*skinids* - List of definitions for character skin/model identifiers

*textstyle* - List of definitions for GameText styles

*vehids* - List of definitions for all available vehicles

*vehhealth* - Vehicle Health configurations

*carcols* - List of definitions for all available vehicle colors

*paintjob* - List of definitions for all available vehicle paintjobs

*weaponids* - List of definitions for weapon identifiers, sorted by weapon slot IDs

*weatherids* - List of definitions for weather identifiers

## 5 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 recommended to be used as parameters. 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"*.

## 6 Icon Identifiers

There are total 64 icons in total included as macro defines with prefix *ICON\_*.

## 7 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 hitboxes because it was used only in cutscene, so in certain interiors you may just fall through ground.

## 8 Pickup Identifiers

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

## 9 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, **R**ich. 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.

## 10 GameText Styles

List of definitions for SA-MP function *GameTextForPlayer(playerid, const string[], time, style)*.

## 11 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.

## 12 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.

## 13 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.

More information is available on SA-MP Wiki: [https://wiki.sa-mp.com/wiki/Vehicle\\_Color\\_IDs](https://wiki.sa-mp.com/wiki/Vehicle_Color_IDs)

## 14 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);
```

## 15 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.

## 16 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]`.

## 17 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.

### 17.1 isampp\_console\_printversion()

*Outputs ISAMPP version to server console.*

**Example of implementation:**

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

### 17.2 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;  
}
```

### 17.3 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;  
}
```

### 17.4 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;  
}
```



## 17.5 MppShowPlayerPosition(playerid, pstringcolor)

*Outputs current player location coordinates, interior identifier, facing angle and player camera position coordinates 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;
}
```

## 17.6 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;
}
```

## 18 Useful Resources

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

### 18.1 Marker Modes

*ShowPlayerMarkers(Marker Mode);*

Identifier	Marker Mode
0	PLAYER_MARKERS_MODE_OFF
1	PLAYER_MARKERS_MODE_GLOBAL
2	PLAYER_MARKERS_MODE_STREAMED

<https://wiki.sa-mp.com/wroot/index.php?title=ShowPlayerMarkers>