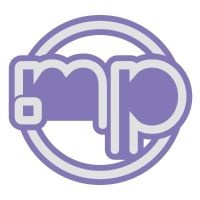
**open.mp (Open Multiplayer)**



**Что такое open.mp?**

**open.mp** (Open Multiplayer) - мультиплеерный мод, разрабатываемый в качестве замены [San Andreas Multiplayer](https://www.sa-mp.com/" \t "_blank) (SA-MP), созданный в ответ на проблемы с обновлениями и разработкой SA-MP. Игроки с клиентом SA-MP смогут подключаться к этому серверу. В дальнейшем станет доступен новый клиент open.mp, что позволит выпускать больше интересных обновлений.

**Возможности**

• **Обратная совместимость** с San Andreas Multiplayer. Это значит, что существующий клиент [SA-MP](https://www.sa-mp.com/) и все скрипты будут работать с [open.mp](https://open.mp/ru).

• **Уникальность.** open.mp не [форк](https://vk.com/away.php?to=https%3A%2F%2Fru.wikipedia.org%2Fwiki%2F%D4%EE%F0%EA&cc_key=" \t "_blank), а полное переписывание с нуля. При таком подходе снимаются юридические вопросы с легальностью использования, и учитываются знания и опыт накопленные за десятилетия.

• **Исправлены ошибки**. Многие [известные ошибки](https://team.sa-mp.com/wiki/Bugs.html) SAMP были исправлены, теперь у вас есть все необходимые инструменты в ваших руках.

• **Производительность.**Значительно повышена производительность и стабильность по сравнению с сервером SA:MP. Сняты [стандартные ограничения SA-MP](https://open.mp/ru/docs/scripting/resources/limits).

•**Расширяемый API**. Используйте расширяемый C++ API с помощью комплекта разработки программного обеспечения open.mp. [PAWN](https://open.mp/ru/docs/scripting/language/reference) по-прежнему остается основным и поддерживаемым языком сценариев.

• **Документация**. [Ресурс заменивший официальную](https://open.mp/ru/docs) *SA-MP wiki*. Наиболее доступный источник документации SA-MP и в конечном счёте open.mp, в который сообщество с легкостью может внести свой вклад.

• **Поддержка** различных платформ. Представьте себе один сервер [open.mp](http://open.mp/), на котором игроки используют оригинальный San Andreas, Android San Andreas и San Andreas Definitive Edition, все три вместе на одном сервере, играя друг с другом.

**История появления**

Несмотря на множественные попытки подтолкнуть вперёд развитие SA-MP официально (в виде советов, просьб и предложений помочь со стороны команды бета-тестеров), а также просьб сообщества сделать хоть что-то новое, не было абсолютно никакого прогресса в работе над SA-MP. Было широко распространено мнение, что причиной этому простое отсутствие интереса со стороны разработчиков мода, что само по себе не проблема, однако не было никакого продвижения.

Команда бета-тестеров, вновь и вновь пыталась заставить *Kalcor'а* заняться обновлениями или передать дальнейшую разработку мультиплеера кому-то другому. Вместо того, чтобы передать разработку мода заинтересованным в этом людям, основатель (***Kalcor***) просто хотел уйти и утянуть всё за собой в пропасть, при этом затягивая процесс как можно дольше для минимизации усилий. Также некоторые утверждают что причина этому - пассивный доход, но нет никаких доказательств этой версии. Несмотря на большой интерес и крупное слаженное сообщество, он был убеждён что моду осталось всего 1-2 года, и что сообщество вложившее много стараний чтобы сделать SA-MP таким какой он есть сейчас, не заслуживает дальнейшего развития. Многие были несогласны с несколькими из его [последних решений](https://vk.com/@rostbomb-intervu-s-yless-ob-openmp), и множественные попытки сменить вектор развития мода ни к чему не привели. Поэтому к сожалению оставалось только одно решение — попытаться создать проект, который будет духовным наследником SA-MP.

Сам по себе мультиплеер великолепен, но его закрытый код означает, что вся власть находится в руках одного человека. Ввиду полного бездействия необходимо было решить главную проблему — его полный контроль над владением исходным кодом. На самом деле никто не хотел переписывать всё с нуля, но это единственный юридически легальный способ возродить всеми любимый мультиплеер. Совершенно новый аналогичный мультиплеер тоже не был бы решением — существует огромное множество серверов и игроков, которые бы не смогли использовать на нём текущие скрипты и игровые моды.

Таким образом основным планом становится создание обратно-совместимого сервера вместе с новым мультиплеером. Мультиплеер позволит игрокам SA-MP подключаться (чтобы владельцы серверов не теряли свою базу игроков), позволяя работать существующим сценариям. Команда из 22 человек во главе с ***Y\_Less*** — одним из создателей существующего SA:MP, работавшего непосредственно с ***Kalсor*** на протяжении 7-8 лет, — с обнадёживающими намерениями взялись за разработку замены существующей платформы с нуля. Разрабатываемый командой профессионалов новый мультиплеер [open.mp](https://open.mp/ru) это своеобразная потенциальная революция, влекущая за собой положительные перемены.



**Установка и настройка сервера open.mp**

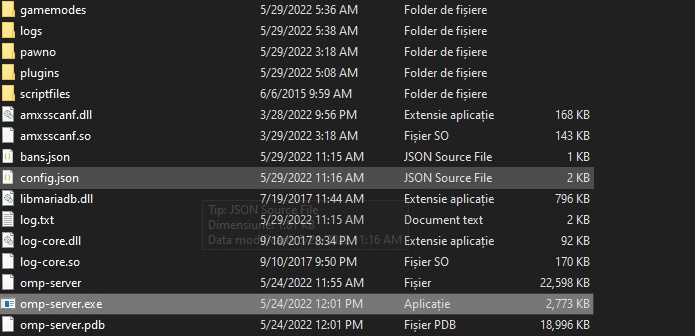
Скачать, опробовать, а также посмотреть полный список изменений и исправлений в [open.mp](http://open.mp/) RC1 вы можете на странице релизов в GitHub репозитории: <https://github.com/openmultiplayer/server-beta/releases>

С релизом приходит новая версия компилятора 3.10.11 и официальные инклуды сервера на замену стандартным самповским (так что временное решение в виде [omp.inc](http://omp.inc/) больше не требуется). Для ваших скриптов появился инструмент "upgrade", который поможет добавить const- и tag- корректность в соответствии со стилем кода в новых библиотеках. Помимо прочего, за это время внутри сервера была обновлена виртуальная машина Pawn, что привносит некоторые изменения и улучшения.

Вы можете легко настроить свой сервер, так как он работает почти так же как сервер SA:MP, но с некоторыми небольшими отличиями. Скачайте последнюю бета-версию сервера <https://github.com/openmultiplayer/server-beta/releases>. Для запуска мода все что вам нужно сделать, это поместить файловую структуру в папку сервера open.mp (или файлы open.mp в корневую папку сервера SA:MP), и все будет готово. читайте server.cfgи загружайте свои скрипты из папок gamemodesи filterscripts, а также загружайте устаревшие плагины из своей pluginsпапки. Новый каталог componentsсодержит библиотеки и [pdb](https://docs.microsoft.com/en-us/visualstudio/debugger/specify-symbol-dot-pdb-and-source-files-in-the-visual-studio-debugger?view=vs-2022" \o "https://docs.microsoft.com/en-us/visualstudio/debugger/specify-symbol-dot-pdb-and-source-files-in-the-visual-studio-debugger?view=vs-2022" \t "_blank) файлы с отладочной информацией.

*Обратите внимание, что плагины для взлома памяти не будут работать! Многие плагины уже работают корректно. (например: sscanf, streamer, MySQL, discord-connector, crashdetect и.т.д.) Список*[*конвертированных плагинов open.mp*](http://open.mp/)

**Запуск omp-server**



Теперь есть три способа управления: классический через существующий [server.cfg](http://server.cfg/" \t "_blank), новый config.jsonили из командной строки. Чтобы использовать существующий файл [server.cfg](http://server.cfg/" \t "_blank), просто скопируйте его в rjhytdjq каталог comp-server и запустите из командной строки:

Linux:

./omp-server

Windows:

omp-server.exe

**Чтобы запустить мод напрямую по имени:**

Linux:

./omp-server --script my-mode.amx

Windows:

omp-server.exe --script my-mode.amx

*Мод должен быть в каталогеgamemodes/, как и раньше.*

**Чтобы сгенерировать [config.json](http://config.json/" \t "_blank):**

Linux:

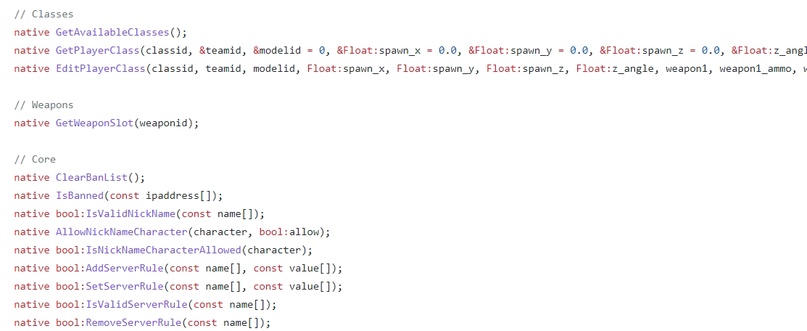
./omp-server --write-config

Windows:

omp-server.exe --write-config

Основные положения описаны в [FAQ](https://open.mp/ru/faq). Если у вас есть какие-либо вопросы относительно настройки вашего сервера с вашими старыми скриптами, не стесняйтесь спрашивать на [Discord сервере](http://discord.gg/samp" \t "_blank) канал ***#omp-server-beta***.

**Подробнее о совместимости и API**



<https://gist.github.com/AmyrAhmady/94a33fc502c2694032523969e7d2ee02>

Во-первых, ОЧЕНЬ важное уточнение - существующие скрипты все равно будут работать как есть. Разработчики много работали над обратной совместимостью и учитывали это при каждом решении.

Так что, опять же, мы должны ясно дать понять, что весь существующий *«код SA:MP»* будет работать. Что это означает? Заработает любой код, который:

* Написан на [pawn](https://ru.wikipedia.org/wiki/Pawn" \t "_blank)
* Использует оригинальный SA:MP API без плагинов
* Перекомпилирован с инклудом [omp.inc](http://omp.inc/)
* Использует [компилятор сообщества](https://github.com/pawn-lang/compiler/releases)

Однако:

* Если вы используете плагин реализованный на языке, отличном от pawn, этот плагин, вероятно, нужно будет сначала портировать. Так что ваш код не будет работать сразу.
* Если вы используете другие плагины, такие как streamer, YSF, audio и.т. д. они могут уже работать, могут нуждаться в переносе или могут быть совершенно излишними, поскольку их функциональность интегрирована в главный сервер.
* Если у вас есть только .AMX вашего мода, а нет исходника. В любом случае, хотя все функции SA:MP существуют, некоторые из них были переделаны или заменены макросами, и вы ДОЛЖНЫ перекомпилировать мод из исходников. Поэтому, если вы не сможете, ваш код вообще не будет работать.

*Слово «plugin» теперь используется для описания устаревших плагинов — тех, которые были написаны для SA-MP, и у open.mp есть API для загрузки этих существующих плагинов. Новое название для подключаемых библиотек в будущем это «modules» и обозначает любой подключаемый модуль который используют более новый, и гораздо более обширный API с open.mp.*

Устаревшие плагины по-прежнему работают нормально и будут продолжать это делать, если только они не используют взлом памяти такой как YSF, который опирается на адреса и структуры, которые совершенно не похожи на этот сервер. Функции базы данных по умолчанию такие же, как и раньше — [SQLite](https://www.sqlite.org/index.html" \t "_blank).

Чтобы пересобрать мод для [open.mp](http://open.mp/), нам нужно изменить только первое включение и добавить одно определение.

#include <a\_samp>

заменить на:

#define OPENMP\_COMPAT

#include <openmp\openmp>

[Пример простого мода](https://pastebin.com/Gs2f6hSu) из официальной документации.

**Предупреждения**

Первая ошибка при компиляции которую вы можете получить:

open.mp scripts require the community compiler from: git.io/pawn-compiler

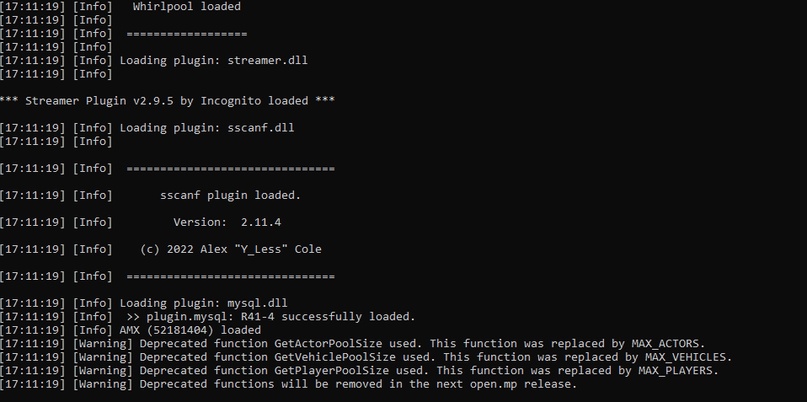
*Если вы получили такую ошибку, перейдите на*[*https://git.io/pawn-compiler*](https://translate.google.com/website?sl=auto&tl=ru&hl=ru&u=https://git.io/pawn-compiler)*и загрузите версию компилятора 3.10.10 или более позднюю. Для релиза мы хотели бы иметь инструмент, эквивалентный pawno, с уже интегрированным компилятором, но пока этого не сделали.*

***Рекомендуется***сначала попытаться скомпилировать ваш режим с помощью этого компилятора, так как он включает предупреждения о корректности константы, поэтому вы вероятно сразу же получите множество новых предупреждений (это НЕ проблема с компилятором, это проблемы в вашем коде которые существовали всегда, но ранее игнорировались). Возможно вам потребуется заменить инклуды на эти: [https://github.com/pawn-lang/pawn-stdlib](https://translate.google.com/website?sl=auto&tl=ru&hl=ru&u=https://github.com/pawn-lang/pawn-stdlib)

*Даже если вы не используете open.mp это будет полезно, так как они исправляют множество проблем с тегами и константами в исходных инклудах!*

Если вы не получили никаких предупреждений при использовании нового компилятора с новой версией *a\_samp* , теперь вы увидите множество новых предупреждений вроде:

warning 234: function is deprecated (symbol "AddPlayerClass") Use "Class\_Add" instead.



У вас есть три варианта, и все они поддерживаются:

* **Не обращайте внимания на предупреждения:** код все равно будет работать.
* **Подавить предупреждения:** Измените OPENMP\_COMPATна OPENMP\_QUIET:

#define OPENMP\_QUIET

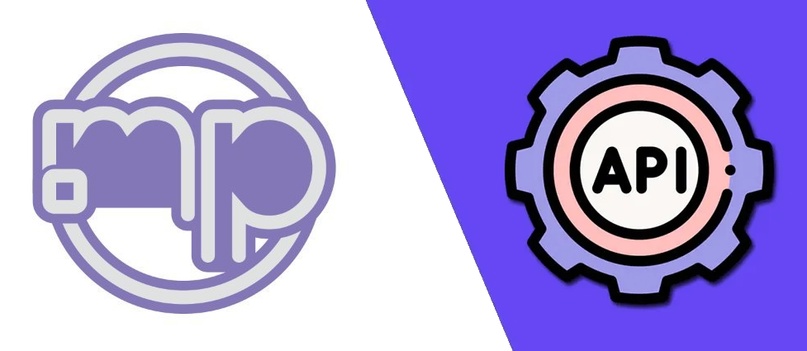
#include <openmp\openmp>

main() {}

* **Исправить предупреждения:** некоторые функции изменили имена для совместимости друг с другом; некоторые функции изменили параметры, потому что старые эволюционировали и не полностью раскрывали возможности open.mp. Пока не существует простого способа преобразовать каждую функцию, но вы можете оставить предупреждения включенными при медленном преобразовании кода — старые функции будут продолжать работать отлично.

**Конверсия мода состоит из трех этапов**

1. **#define OPENMP\_QUIET.**Использование этого определения позволяет компилировать ваш режим без новых предупреждений от устаревших функций. Вы можете конвертировать код в этом режиме, и новый API будет работать, но вы не сможете использовать компилятор чтобы увидеть где остались проблемы.
2. **#define OPENMP\_COMPAT.**Выдаст предупреждения для старых функций, но они по-прежнему будут работать, позволяя вам конвертировать части вашего режима по одной за раз. Используйте когда вы захотите начать пользоваться всеми улучшенными функциями open.mp
3. **No define.**Как только вы думаете, что закончили преобразование своего мода, вы удаляете эти определения. Теперь вы получаете ошибки вместо предупреждений для любого старого кода, который все еще используется.



**Новый API**

Разработчики очень старались сделать API простым в использовании, простым в освоении, но также совместимым с предыдущими версиями. Во многом успех SA:MP обусловлен простотой использования, и необходимо сохранить ее. Но, также есть и опытные пользователи которые хотят получить гораздо больше от своего кода. Есть много улучшений которые возможно сделать, но которые не внедрить по причине сохранения совместимости. Тем не менее, есть улучшения, которые можно сделать повсюду.

*Подробно про новый API прочитайте в оригинальной статье*[*https://open.mp/ru/blog/scripting-api*](https://open.mp/ru/blog/scripting-api)*, все что будет ниже это краткая выжимка из данной статьи.*

**Документация**

Документация по новым стандартным библиотекам:  
<https://github.com/openmultiplayer/omp-stdlib>Документация по новому компилятору:  
<https://github.com/openmultiplayer/compiler/>Документация по редактору кода qawno:  
<https://github.com/openmultiplayer/qawno/>Документация по инструменту upgrade:  
<https://github.com/openmultiplayer/upgrade>

**SDK**

Pawn — это давний и до сих пор официальный способ написания модов для вашего сервера. Он никуда не исчезнет, но для тех из вас, кто хочет большего контроля, мы наконец-то выпускаем полный SDK (Software Development Kit). Это интерфейс C++ для сервера, тот же самый, который используется всеми компонентами, составляющими основной код [open.mp](http://open.mp/). Все, что они могут делать, можете сделать и вы (по сравнению с плагинами, которые были разработаны только для предоставления Pawn функций, а не написания модов). Несколько примеров компонентов для тех из вас, кто хочет сразу вникнуть. Это всё шаблоны, на которые вы можете опираться и быстро перейти к полностью работающему компоненту с наиболее распространёнными функциями:

<https://github.com/openmultiplayer/empty-template><https://github.com/openmultiplayer/basic-template><https://github.com/openmultiplayer/pawn-template><https://github.com/openmultiplayer/full-template>

**Теги**

***«****Тег — это префикс переменной, который сообщает компилятору, что при определенных обстоятельствах следует обрабатывать эту переменную особым образом.****»***

Функции теперь используют [теги](https://open.mp/ru/docs/scripting/language/Tags) практически везде. Разработчики попытались найти баланс между слишком большим количеством предупреждений о тегах, и недостаточным количеством полезной информации. Предупреждения существуют не просто так, и могут помочь найти проблемы, которые вы возможно пропустили. Например, передать транспортное средство в качестве параметра объектной функции или дать кому-то несуществующее оружие:

// No warnings for this code, despite it being clearly wrong.

new object = CreateObject(various, parameters, here);

PutPlayerInVehicle(playerid, object);

// Same here - there's no weapon 20, despite it being amongst valid weapon IDs.

GivePlayerWeapon(playerid, 20, 200);

Было бы намного лучше, если бы все участки явно неправильного кода могли вызывать предупреждения. В этом сила языка с [типобезопасностью](https://vk.com/away.php?to=https%3A%2F%2Fru.wikipedia.org%2Fwiki%2F%D2%E8%EF%EE%E1%E5%E7%EE%EF%E0%F1%ED%EE%F1%F2%FC&cc_key=" \t "_blank), и хотя pawn не является таковым, мы можем приблизиться к этому с помощью [тегов](https://open.mp/ru/docs/scripting/language/Tags).

// warning 213: tag mismatch: expected tag "Vehicle", but found "Object"

new Object:object = Object@Create(various, parameters, here);

Player\_PutInVehicle(playerid, object);

// warning 213: tag mismatch: expected tag "WeaponType", but found but found none ("\_")

GivePlayerWeapon(playerid, 20, 200);

// This doesn't give a warning:

GivePlayerWeapon(playerid, WEAPON\_COLT45, 200);

**Именование функций**

Большинство новых функций используют усовершенствование схемы именования, уже принятой многими библиотеками и плагинами — Module\_VerbNoun. Больше не нужно задаваться вопросом, было ли это «Rot» или «Rotation» для этого типа элемента, нет никаких сокращений.(если только имя функции [не более 32 символов](https://github.com/pawn-lang/compiler/issues/123#issuecomment-265628599), компилятор их не примет). Например хотите изменить модель объекта? Object\_SetModel, хотите показать текст игроку? Text\_Show.

Есть ограниченный набор глаголов - Get, Set, Create, Destroy, Add, Remove, Show, Hide, Run, Move, Stop, и Count. Безусловно, наиболее распространенными являются Getи Set, которые в отличие от SA:MP теперь всегда идут парами. Если вы можете установить какой-либо параметр, вы также можете использовать его позже. Вам нужно указать что получить или установить Health, Position, Modelи.т.д.

Некоторые примеры:

native bool:Menu\_SetDisabled(Menu:menuid, bool:disabled);

native bool:Menu\_GetDisabled(Menu:menuid);

native bool:Text\_SetAlignment(Text:text, alignment);

native Text\_GetAlignment(Text:text);

native bool:Object\_Move(Object:objectid, Float:posX, Float:posY, Float:posZ, Float:speed, Float:rotX = FLOAT\_NAN, Float:rotY = FLOAT\_NAN, Float:rotZ = FLOAT\_NAN);

native DBResult\_CountRows(DBResult:dbresult);

native Player\_Spawn(Player:playerid);

Обратите внимание, что имена модулей и тегов всегда совпадают - Vehicle, DB, Playerи т. д. [Для этого есть причины, помимо согласованности](https://github.com/pawn-lang/compiler/issues/234) - это дает интерфейс, более похожий на [объектно-ориентированный](https://highload.today/obektno-orientirovannoe-programmirovanie/), и его легче запомнить.

**Константы**

Начнем с того что вместо использования [директивы](https://ziggi.org/pawn-direktivy-preprotsessora/) #defineдля всего подряд, мы используем constи enumкак можно больше, за исключением случаев когда мы ожидаем что что-то будет переопределено (MAX\_PLAYERS):

// Defining this first didn't previously work, now it does.

#define MAX\_PLAYERS 100

#include <openmp\openmp>

// The override method still works as well.

#undef MAX\_PLAYERS

#define MAX\_PLAYERS 200

[Использование перечислений](https://open.mp/docs/scripting/language/Initialisers" \t "_blank) означает, что константы имеют теги для проверки во время компиляции:

enum ObjectMaterialTextAlignment

{

MATERIAL\_TEXT\_ALIGN\_LEFT,

MATERIAL\_TEXT\_ALIGN\_CENTRE,

MATERIAL\_TEXT\_ALIGN\_RIGHT,

};

Object\_SetMaterialTextтеперь будет принимать только одно из этих трех значений и ничего больше.

**Функции для каждого игрока**

Таких больше нет каждая ForPlayer, ForAll, CreatePlayerXи.т.д. функция заменена одной простой функцией X\_Display(где Xлюбая сущность), и X\_Hasдля проверки:

Object\_Display(objectid, playerid, true); // Show it.

Object\_Display(objectid, playerid, false); // Hide it.

Text\_Display(textid, true); // Show it to everyone.

if (Zone\_Has(zoneid, playerid))

{

// The player is ALLOWED to see the gang zone.

}

[YSI](https://github.com/pawn-lang/YSI-Includes) использовал X\_SetPlayer но показывать что-то игрокам — это самое основное что нужно делать, поэтому он заслуживает отдельного глагола. Некоторые библиотеки используют X\_Showи X\_Hide но это две функции, которые просто приводят к лишнему коду при проверке того что делать:

if (var) Checkpoint\_Show(cpid, playerid);

else Checkpoint\_Hide(cpid, playerid);

против:

Checkpoint\_Display(cpid, playerid, var);

**Тестирование**

Тестировать код в SA:MP было не очень просто. По большей части это просто подключение к серверу использование print() в ключевых местах. [Библиотеки](https://github.com/openmultiplayer/pawn-test-scripts), такие как [y\_testing](https://github.com/pawn-lang/YSI-Includes" \t "_blank) значительно упрощают написание модульных тестов. Увы это работает только для достаточно «чистого» кода, кода который выполняется на сервере без особого взаимодействия с игроком. Но при тестировании такие методы по-прежнему ограничены, и не помогают сузить круг проблем когда они возникают.

Итак, как open.mp решает эту проблему? Начнем с того, что исходный код является открытым, поэтому у вас есть полный набор существующих инструментов отладки для пошагового выполнения и проверки кода. Например таких как [GDB](https://www.sourceware.org/gdb/) и [Visual Studio](https://docs.microsoft.com/en-us/visualstudio/debugger/?view=vs-2022" \o "https://docs.microsoft.com/en-us/visualstudio/debugger/?view=vs-2022" \t "_blank), У нас также есть планы добавить аналогичные инструменты для самих pawn скриптов.

Но самым большим дополнением является «*[mock](https://vk.com/away.php?to=https%3A%2F%2Fru.wikipedia.org%2Fwiki%2FMock-%EE%E1%FA%E5%EA%F2&cc_key=" \t "_blank)*» — фиктивная библиотека. Библиотека позволяет создавать поддельных игроков, которые действуют точно так же как настоящие игроки, но без какого-либо подключенного клиента. Таким образом любые действия выполняемые над ними сценарием, могут быть прочитаны и проанализированы другим сценарием.

Подробнее ознакомьтесь в основной статье <https://open.mp/ru/blog/debugging>

**Синхронизация и клиентская часть**

Недавно было представлено демо-видео клиента на ранних стадиях, как простая ранняя демонстрация. (поэтому есть некоторые глюки, авторы просто хотим показать, что это возможно и они работает над этим)

YouTube1:48

Демонстрационный ролик на YouTube: <https://youtu.be/-PWXXvHe35M>

Клиент Definitive Edition для open.mp будет совместим с серверами open.mp! Таким образом, вы можете разместить сервер и позволить людям решать, с какой игрой они хотят присоединиться к серверу! Представьте себе один сервер на котором игроки используют оригинальный *San Andreas, Android San Andreas* и *San Andreas Definitive Edition*, все три вместе на одном сервере играя друг с другом. Таким образом вам как владельцу сервера, не нужно беспокоиться о новой платформе на которую можно перейти, начать заново и разделить свое сообщество.

Обратите внимание, ***клиент open.mp все еще находится в разработке,***и в бета версии представлена только серверная часть, к которой подключение работает через [клиент SA-MP 0.3.7.](https://www.sa-mp.com/download.php) Разработчики не будут поддерживать версии ниже 0.3.7, но вы сможете сделать это самостоятельно (в этом вся прелесть открытого кода). Если большинство проектов [запросят](https://www.burgershot.gg/showthread.php?tid=1039) 0.3DL, появится поддержка со стороны команды разработчиков.

**Вместо заключения**

И, конечно же, проект сервера всё ещё находится в активной разработке, поэтому пожалуйста регулярно проверяйте все ссылки, чтобы узнать о всех новшествах. Эта тема будет обновляться по мере развития проекта. Следить за проектом можно на <https://open.mp/ru/> форуме <https://forum.open.mp/>, Discord сервере <http://discord.gg/samp>, группе в *VK* <https://vk.com/open_mp> . Несмотря на то, что open.mp уже на пути к первому релизу [помощь приветствуется](https://www.burgershot.gg/showthread.php?tid=361).

