

# Приложения на устройстве AQSI CUBE

---

## Оглавление

### Приложения на устройстве AQSI CUBE

#### Программа Display Manager (dm)

Каталог приложения

Запуск приложения

#### Программа JPAY

Перезапуск программы и одновременная работа нескольких экземпляров программы

Каталог приложения

Запуск приложения стандартный

Запуск приложения для отладки в режиме консоли

Команды JPAY для работы в режиме консоли

#### Программа Payment Application (PAPP)

Каталог приложения

Запуск приложения

#### Программа VEND

Каталог приложения

Запуск приложения

#### Взаимодействие JPAY PAPP и VEND

Варианты интеграции

#### Программа vend\_ifaces (vi)

Каталог приложения

Запуск приложения

На устройстве AQSI CUBE коммуникация между приложениями осуществляется через интерфейс [D-Bus](#). Исключения составляют платёжные приложения - JPAY, PAPP и VEND. Там используется более "быстрое" взаимодействие через сокеты.

Подробную схему взаимодействия приложений на устройстве AQSI CUBE можно посмотреть по ссылке <https://sketchboard.me/BBY1davoASRI#/>

## Программа Display Manager (dm)

---

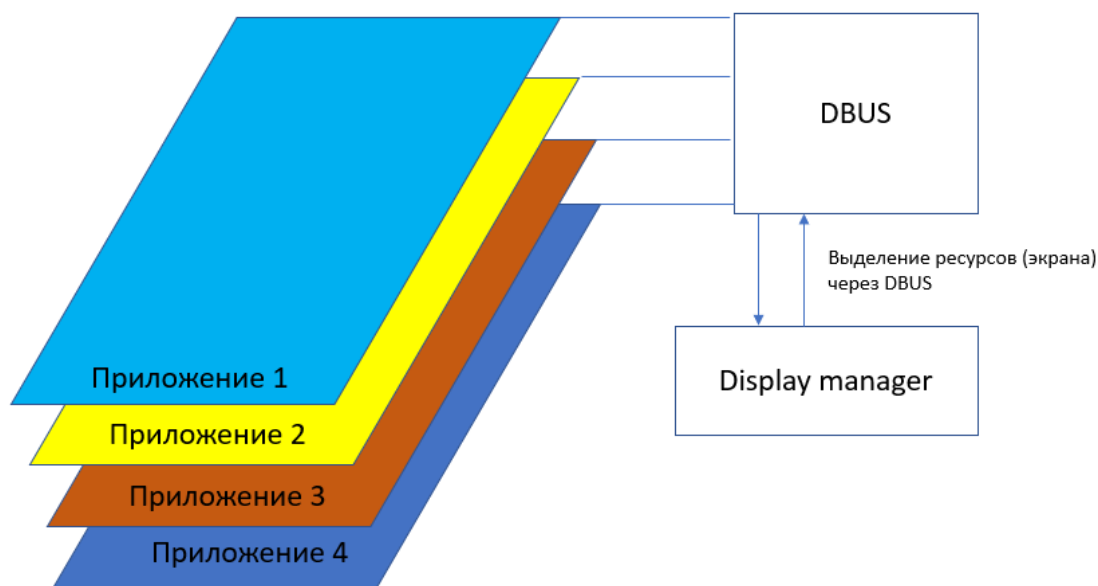
**Display manager** управляет отображением других приложений на экране. Принцип работы приложения иллюстрирует Рисунок 1.

Оконный интерфейс имеет следующие особенности:

- Любое "окно" всегда занимает весь экран устройства AQSI CUBE, частичное наложение" окон **Display manager** не допускает. Т.е. каждое окно занимает все 100% экрана.
- Если приложение хочет "занять место на экране" оно должно уведомить об этом **Display manager** и получить от него подтверждение, что экран выделен. При этом оно оказывается "сверху" всех приложений.

- Если **Display manager** хочет выделить экранный ресурс другому приложению (например, когда очередное приложение хочет "занять экран"), он уведомляет текущее "верхнее" оконное приложение о том, что у него изымается экранный ресурс. После получения такого уведомления, экранное приложение не должно ничего перерисовывать на экране. Но при этом и удалять с экрана ничего не требуется. Т.е. на экране "застывает" предыдущее "верхнее" приложение в том состоянии, которое было на момент отзыва экранного ресурса. Ответственность за дальнейшую перерисовку экрана переходит к новому "верхнему" приложению.

Рисунок 1. Принцип работы приложения Display Manager



## Каталог приложения

Приложение **Display Manager** размещается в каталоге **/opt/dm** Бинарный файл приложения **/opt/dm/bin/dm**

## Запуск приложения

Приложение запускается скриптом **/etc/init.d/start-dm**

Возможные параметры:

- start
- stop
- restart

Пример:

```
/etc/init.d/start-dm start
```

По умолчанию приложение автоматически запускается при загрузке AQSI CUBE, так как линк на этот скрипт включен в runlevel 5 по умолчанию (см. линки в каталоге /etc/init.d/rc5.d):

```
root@143C214C5254324B:/etc# ls -l rc5.d
total 0
lrwxrwxrwx 1 root root 20 Jul 13 15:24 S01networking -> ../init.d/networking
lrwxrwxrwx 1 root root 16 Jul 13 15:23 S02dbus-1 -> ../init.d/dbus-1
lrwxrwxrwx 1 root root 19 Jul 13 15:23 S03rng-tools -> ../init.d/rng-tools
lrwxrwxrwx 1 root root 17 Jul 13 15:24 S05connman -> ../init.d/connman
lrwxrwxrwx 1 root root 14 Jul 13 15:23 S09sshd -> ../init.d/sshd
lrwxrwxrwx 1 root root 29 Jul 13 15:24 S10load-drivers-cube-c -> ../init.d/load-
drivers-cube-c
lrwxrwxrwx 1 root root 19 Jul 13 15:24 S11start-hwk -> ../init.d/start-hwk
lrwxrwxrwx 1 root root 17 Jul 13 15:23 S12rpcbind -> ../init.d/rpcbind
lrwxrwxrwx 1 root root 20 Jul 13 15:24 S12start-ntpd -> ../init.d/start-ntpd
lrwxrwxrwx 1 root root 21 Mar 18 18:33 S15mountnfs.sh -> ../init.d/mountnfs.sh
lrwxrwxrwx 1 root root 19 Jul 13 15:23 S19nfscommon -> ../init.d/nfscommon
lrwxrwxrwx 1 root root 14 Jul 13 15:24 S20apmd -> ../init.d/apmd
lrwxrwxrwx 1 root root 13 Jul 13 15:23 S20atd -> ../init.d/atd
lrwxrwxrwx 1 root root 19 Jul 13 15:24 S20bluetooth -> ../init.d/bluetooth
lrwxrwxrwx 1 root root 21 Jul 13 15:23 S20dhcp-server -> ../init.d/dhcp-server
lrwxrwxrwx 1 root root 17 Jul 13 15:24 S20hostapd -> ../init.d/hostapd
lrwxrwxrwx 1 root root 20 Jul 13 15:24 S20hwclock.sh -> ../init.d/hwclock.sh
lrwxrwxrwx 1 root root 19 Jul 13 15:23 S20nfsserver -> ../init.d/nfsserver
lrwxrwxrwx 1 root root 16 Jul 13 15:23 S20syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 22 Jul 13 15:24 S21avahi-daemon -> ../init.d/avahi-daemon
lrwxrwxrwx 1 root root 15 Jul 13 15:23 S22ofono -> ../init.d/ofono
lrwxrwxrwx 1 root root 18 Jul 13 15:24 S31start-dm -> ../init.d/start-dm
lrwxrwxrwx 1 root root 20 Jul 13 15:24 S32start-vend -> ../init.d/start-vend
lrwxrwxrwx 1 root root 18 Jul 13 15:24 S32start-vi -> ../init.d/start-vi
lrwxrwxrwx 1 root root 20 Jul 13 15:24 S33start-papp -> ../init.d/start-papp
lrwxrwxrwx 1 root root 20 Jul 13 15:24 S34start-jpay -> ../init.d/start-jpay
lrwxrwxrwx 1 root root 18 Jul 13 15:24 S70swupdate -> ../init.d/swupdate
lrwxrwxrwx 1 root root 15 Jul 13 15:23 S90crond -> ../init.d/crond
lrwxrwxrwx 1 root root 23 Jul 13 15:24 S95startup_check ->
../init.d/startup_check
lrwxrwxrwx 1 root root 28 Jul 13 15:24 S96start_updater_root ->
../init.d/start_updater_root
lrwxrwxrwx 1 root root 22 Mar 18 18:33 S99rmnologin.sh -> ../init.d/rmnologin.sh
lrwxrwxrwx 1 root root 23 Mar 18 19:00 S99stop-bootlogd -> ../init.d/stop-
bootlogd
```

При этом следует обратить внимание: Display Manager имеет индекс S31 и запускается раньше, чем другие приложения AQSI (JPAY, VEND и PAPP). Таким образом, все приложения, которые используют экран, запускаются после **Display Manager**.

Код скрипта для запуска **Display Manager**:

```
#!/bin/sh

NAME=dm
DESC="display manager"
```

```

DAEMON="/opt/${NAME}/bin/${NAME}"
PIDFILE="/var/run/start-dm.pid"

[ -x "$DAEMON" ] || exit 0

do_start()
{
    (while true
    do
        $DAEMON
        RET=$?
        if [ $RET == 137 ] || [ $RET == 0 ]
        then
            break
        fi
    done) &
    echo $! > $PIDFILE
}

do_stop()
{
    start-stop-daemon --stop --quiet --oknodo -p $PIDFILE --remove-pidfile
    start-stop-daemon --stop --quiet --oknodo --retry 5 --signal KILL --exec
$DAEMON
}

case "$1" in
    start)
        echo "Starting $DESC"
        do_start
        ;;
    stop)
        echo "Stopping $DESC"
        do_stop
        ;;
    restart)
        $0 stop
        $0 start
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}" >&2
        exit 1
        ;;
esac

exit 0

```

# Программа JPAY

Программа JPAY - это эквайринговое ядро компании AQSI, которое используется как на платформе Linux, так и на платформе Android (для касс AQSI). Взаимодействие с программой осуществляется через сокет по порту 4433. Настройка программы осуществляется на этапе производства устройства через сокет 4434. Программа никак не взаимодействует с экраном. Через сокет передаются сообщения в формате XML.

Для корректной работы программы используются следующие библиотеки и файлы:

| Файл // линк                                    | Назначение библиотеки                                                    |
|-------------------------------------------------|--------------------------------------------------------------------------|
| /usr/lib/libaqc_emv_ct_l1.so.0                  | Библиотека L1 для работы с контактным (ct) интерфейсом карты.            |
| /usr/lib/libaqc_emv_ctls_l1.so.0                | Библиотека L1 для работы с бесконтактным (ctls) интерфейсом карты.       |
| /usr/lib/libL2.so/usr/lib/libaqc_emv_ct_l1.so.0 | Библиотека L2 для работы эквайрингового ядра.                            |
| /opt/jpay/bin/jpay                              | Бинарный исполняемый файл программы JPAY.                                |
| /mnt/data/cardcore/*                            | В каталоге /mnt/data/cardcore расположены конфигурационные файлы JPAY.   |
| /etc/init.d/start-jpay                          | Скрипт для запуска или перезапуска JPAY при старте операционной системы. |

Обычно в каталоге /usr/lib размещаются файлы библиотек с номером с номером версии вида **<lib\_name>.so.x.y.z**, а вместо файла создаётся символическая ссылка на актуальную версию библиотеки:

```
ls -l /usr/lib/libL2*
lrwxrwxrwx 1 root root      25 Jul 23 08:29 /usr/lib/libL2.so ->
/usr/lib/libL2.so.1.0.1_2
-rwxr-xr-x 1 root root 1016924 Jul 13 14:32 /usr/lib/libL2.so.1.0.0
-rwxr-xr-x 1 root root 1103028 Jul 19 09:24 /usr/lib/libL2.so.1.0.1
-rwxr-xr-x 1 root root 1103028 Jul 23 08:29 /usr/lib/libL2.so.1.0.1_2
```

Т.е. в данном случае из приведённого выше листинга видно, что **/usr/lib/libL2.so** - является символической ссылкой, а не обычным файлом. По версиям библиотек L1 и программы JPAY используется аналогичный подход.

## Перезапуск программы и одновременная работа нескольких экземпляров программы

Если пользователь случайно запустил программу JPAY в двух экземплярах, эти два экземпляра программы начинают конфликтовать друг с другом. Ни в коем случае не запускайте программу `/opt/jpay/bin/jpay`, если на устройстве уже работает экземпляр приложения. Если хотите перезапустить программу - сначала остановите текущий работающий экземпляр программы.

Для получения информации о работающих экземплярах программы используйте команду:

```
# ps -ef | grep jpay | grep -v grep
root      446      1  0 Jul27 ?          00:00:00 /bin/sh /etc/rc5.d/S34start-jpay
start
root      447     446  0 Jul27 ?          00:00:00 /opt/jpay/bin/jpay
```

В данном листинге мы видим "нормальную ситуацию" - работает сама программа (процесс `/opt/jpay/bin/jpay`) и скрипт перезапуска (процесс `/etc/rc5.d/S34start-jpay`). Если скрипт перезапуска "обнаруживает", что программа JPAY перестала работать, он автоматически запускает новый экземпляр программы.

Для того чтобы "перезапустить" экземпляр программы, необходимо "убить" текущий работающий процесс `/opt/jpay/bin/jpay`. После этого скрипт запуска/перезапуска автоматически перезапустит jpay.

В частности, для листинга выше, этот процесс имеет номер 447. Соответственно, команда `kill 447` приведёт к тому, что процесс jpay будет автоматически перезапущен и Вы увидите его с новым номером:

```
# kill 447
#
# ps -ef | grep jpay | grep -v grep
root      446      1  0 Jul27 ?          00:00:00 /bin/sh /etc/rc5.d/S34start-jpay
start
root     23606     446  1 15:08 ?          00:00:00 /opt/jpay/bin/jpay
#
```

Как видно из последнего листинга, процесс перезапустился и новый номер процесса JPAY стал 23606.

Для того чтобы полностью остановить программу, необходимо:

1. Сначала остановить скрипт перезапуска программы `/etc/init.d/start-jpay` (или линк на него `/etc/rc5.d/S34start-jpay`)
2. Затем остановить саму программу `/opt/jpay/bin/jpay`

В случае последнего листинга, номера процессов соответственно 446 и 23606. Соответственно, команды для остановки должны быть такие:

```
# kill 446
# kill 23606
```

## Каталог приложения

Приложение **JPAY** размещается в каталоге **/opt/jpay** Бинарный файл приложения **/opt/jpay/bin/jpay**

Конфигурационный каталог приложения находится в папке **/mnt/data/cardcore**

## Запуск приложения стандартный

Приложение запускается скриптом **/etc/init.d/start-jpay**

Возможные параметры:

- start
- stop
- restart

*Пример:*

```
/etc/init.d/start-jpay start
```

По умолчанию приложение автоматически запускается при загрузке AQSI CUBE, так как линк на этот скрипт включен в runlevel 5 по умолчанию (см. линки в каталоге **/etc/init.d/rc5.d**).

## Запуск приложения для отладки в режиме консоли

В целях отладки иногда бывает необходимо запустить приложение JPAY в режиме консоли. В этом случае необходимо выполнить следующие действия:

1. Остановить скрипт **/etc/init.d/start-jpay** и процесс **/opt/jpay/bin/jpay**

```
# ps -ef | grep jpay | grep -v grep
root      446      1  0 Jul27 ?           00:00:00 /bin/sh /etc/rc5.d/s34start-jpay
start
root      23606   446  0 15:08 ?           00:00:00 /opt/jpay/bin/jpay
# kill 446
# kill 23606
```

2. Убедиться, что на устройстве не работает ни один из вышеупомянутых процессов

```
# ps -ef | grep jpay | grep -v grep
```

Команда не должна вернуть ни одного процесса.

3. Запустить программу JPAY в режиме консоли. При этом Вы сможете в консоли JPAY давать какие-то команды для программы JPAY и визуально видеть ответы программы. Для запуска программы в консоли просто наберите **/opt/jpay/bin/jpay** в командной строке.

## Команды JPAY для работы в режиме консоли

В режиме консоли пользователь может давать программе JPAY команды не в режиме XML (как обычно требует JPAY), а в виде текстовых односложных команд. В частности, можно использовать следующие команды:

| Команда | Значение                                                                                                                                                                                                                                                                             |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| login   | Авторизация на устройстве. После этой команды можно выполнять транзакции или обновлять конфигурацию. Обычно при подключении через порт 4433 при авторизации требуется пароль, но при работе в консоли пароль не требуется.                                                           |
| update  | Загрузка конфигурации терминала с сервера AQSI. Может использоваться либо тестовый сервер либо боевой в зависимости от того какой адрес был указан при активации устройства в процессе производства. В настоящий момент (лето 2020 года), как правило, используется тестовый сервер. |
| info    | Отобразить текущую конфигурацию.                                                                                                                                                                                                                                                     |
| paytest | Тестовый эквайринговый платёж.                                                                                                                                                                                                                                                       |
| reset   | Команда для повторной активации устройства, используемая при производстве. Одновременно открывает порт 4434 для получения конфигурационных данных. Просьба к партнёрам не использовать эту команду никогда. Команда должна использоваться только сотрудниками AQSI.                  |

## Программа Payment Application (PAPP)

Это специализированная программа, которая «дополняет» функционал JPAY в части пользовательских интерфейсов. ПП (также как и JPAY) получает команды на выполнение эквайринговых операций в виде сообщений в формате XML. Формат команд ПП и JPAY близок, но не полностью идентичен в силу того, что ПП «берёт на себя» часть технических коммуникаций.

Т.е. основные функции PAPP:

- обеспечение графического пользовательского интерфейса пользователя при осуществлении платежей
- упрощение коммуникации в процессе выполнения транзакции за счёт автоматической обработки технических коммуникационных сообщений JPAY

Программа PAPP в процессе коммуникации использует такой же формат сообщений XML как и программа JPAY с некоторыми упрощениями. Подробности изложены в документе [Протокол взаимодействия с терминалом оплаты AQSI Cube](#)

Как видно из перечисленного выше, PAPP также осуществляет взаимодействие с программой **Display Manager**.

## Каталог приложения

Приложение **PAPP** размещается в каталоге **/opt/papp** Бинарный файл приложения **/opt/papp/bin/papp**



## Запуск приложения

Приложение запускается скриптом **/etc/init.d/start-papp**

Возможные параметры:

- start
- stop
- restart

*Пример:*

```
/etc/init.d/start-papp start
```

По умолчанию приложение автоматически запускается при загрузке AQSI CUBE, так как линк на этот скрипт включен в runlevel 5 по умолчанию (см. линки в каталоге /etc/init.d/rc5.d).

Для корректной работы программы PAPP должна работать программа JPAY.

## Программа VEND

VEND - это специализированная программа, которая выполняет следующие функции:

- конвертирует информацию из формата JPAY в формат транспортного протокола AQSI
- принимает команды от приложения клиента на выполнение эквайринговых операций
- выполняет эквайринговые операции при помощи Платёжного Приложения (PAPP)
- возвращает результат в приложение клиента
- дублирует информацию о платеже в Личный Кабинет AQSI CUBE (на сервер)
- в случае, если предусматривается формирование фискального чека через облачный сервис Оранж-Дата, приложение VEND получает фискальные данные из ЛК (с сервера) и осуществляет отображение QR-кода на экране AQSI CUBE
- в случае, если предусматривается формирование фискального чека при помощи встроенного фискального регистратора rayonline, приложение VEND осуществляет взаимодействие с устройством, формирование фискального чека, передачу фискальных данных в ЛК вместе с чеком и отображение QR-кода на экране

Подробности о протоколе VEND изложены в документе [Протокол взаимодействия с терминалом оплаты AQSI Cube](#)

## Каталог приложения

Приложение **PAPP** размещается в каталоге **/opt/vend** Бинарный файл приложения **/opt/vend/bin/vend**

## Запуск приложения

Приложение запускается скриптом `/etc/init.d/start-vend`

Возможные параметры:

- start
- stop
- restart

Пример:

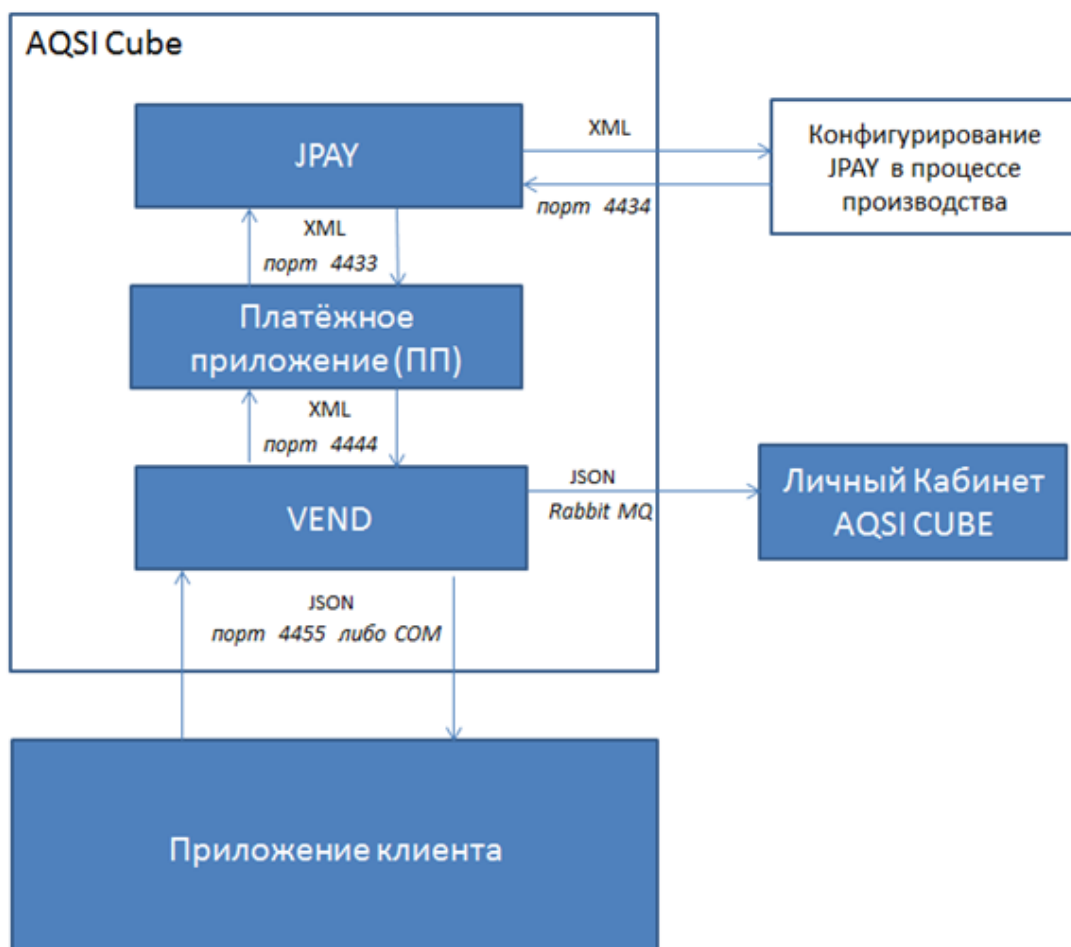
```
/etc/init.d/start-vend start
```

По умолчанию приложение автоматически запускается при загрузке AQSI CUBE, так как линк на этот скрипт включен в runlevel 5 по умолчанию (см. линки в каталоге `/etc/init.d/rc5.d`).

## Взаимодействие JPAY PAPP и VEND

На Рисунке 2 изображена схема взаимодействия трёх платёжных приложений JPAY, PAPP и VEND.

Рисунок 2. Схема взаимодействия трёх платёжных приложений JPAY, PAPP и VEND



## Варианты интеграции

Разработчики приложений на платформе AqSI cube могут осуществлять интеграцию на любом уровне:

- если требуется интеграция с вендинговым автоматом и Личный Кабинетом AqSI Cube - целесообразно интегрироваться с приложением VEND
- если требуется просто выполнение платежей и графический интерфейс - лучше всего интегрироваться с приложением PAPP
- если графический интерфейс не требуется, либо требуется специфический графический интерфейс - можно интегрироваться с программой JPAY
- если требуется нестандартное поведение эквайринга (например, передача транзакций в отложенном режиме) - можно использовать непосредственно библиотеку **/usr/lib/libL2.so**

Соответственно, если Вы не планируете использовать какое-либо приложение (например, Вы не используете VEND и планируете интегрироваться с приложением PAPP), вам необходимо исключить автоматический запуск приложения VEND, убрав соответствующий линк из папки инициализации RunLevel 5: **/etc/rc5.d/**

## Программа vend\_ifaces (vi)

Программа **vend\_ifaces** обеспечивает взаимодействие программы VEND с шиной вендингового автомата. Программа поддерживает различные протоколы (EXE, MDB, Pulse)

### Каталог приложения

Приложение **PAPP** размещается в каталоге **/opt/vend\_ifaces** Бинарный файл приложения **/opt/vend\_ifaces/bin/vend\_ifaces**

### Запуск приложения

Приложение запускается скриптом **/etc/init.d/start-vi**

Возможные параметры:

- start
- stop
- restart

*Пример:*

```
/etc/init.d/start-vi start
```

По умолчанию приложение автоматически запускается при загрузке AqSI CUBE, так как линк на этот скрипт включен в runlevel 5 по умолчанию (см. линки в каталоге **/etc/init.d/rc5.d**).