

Централизованное обновление

- [Настройки Puppet](#)
- [Запуск обновления через Puppet вручную](#)
- [Автоматическое обновление через Puppet](#)
- [Механизм обновления](#)
- [Оповещение об обновлении](#)

Сервер управления конфигурацией Puppet позволяет задавать и контролировать автоматическое обновление или откат отдельных модулей и системы в целом до заданной версии.

Управление системой централизованного обновления осуществляется при помощи пакета `artix-upgrade-system2`. Работа модуля позволяет обеспечить высокий уровень надежности и минимизировать риски благодаря разделению процессов загрузки и установки пакетов. Пакет `artix-upgrade-system2` полностью контролирует загрузку и установку, а при возникновении сбоев программы проверяет корректность исполнения обеих операций и завершает их должным образом.

Настройками пакета `artix-upgrade-system2` предусмотрено ограничение скорости скачивания обновлений. Время для начала загрузки обновлений на кассе выбирается случайным образом в пределах определенного интервала. Такие настройки позволяют исключить возможность высокой нагрузки на сеть.

Настройки Puppet

Настройки периодичности выхода на связь с Puppet задаются в конфигурационном файле `/etc/puppet/puppet.conf` в секции `[agent]`.

Наименование	Тип данных	Возможные значения	Описание	Примечания
<code>runinterval</code>	строковый		Время в секундах выхода на Puppet сервер	Может задаваться в: <ul style="list-style-type: none">• секундах (30 или 30s);• минутах (30m);• часах (6h);• днях (2d).
<code>splay</code>	логический	<ul style="list-style-type: none">• <code>true</code>• <code>false</code>	Спать перед первым выходом на связь сервером	<ul style="list-style-type: none">• <code>true</code> – спим;• <code>false</code> – сразу выходим на связь. По умолчанию <code>false</code> .
<code>splaylimit</code>	строковый		Максимальное время задержки перед первым запуском агента при включенном <code>splay</code>	Может задаваться в: <ul style="list-style-type: none">• секундах (30 или 30s);• минутах (30m);• часах (6h);• днях (2d). По умолчанию используется <code>\$runinterval</code> агента.

```
...
[agent]
...
runinterval = 10
splay = false
splaylimit = 7200
...
```

Команда для просмотра конкретной настройки сервиса Puppet

```
puppet agent --configprint runinterval
```



Полный перечень настроек можно найти на сайте [Puppet](#).

Запуск обновления через Puppet вручную

Для обновления через Puppet вручную необходимо выполнить команду:

```
puppet agent -t
```

Автоматическое обновление через Puppet

Для работы сервиса Puppet в фоновом режиме необходимо:

1. Создать файл:

```
touch /etc/default/puppet
```

2. Настроить периодический выход на связь с Puppet в конфигурационном файле /etc/puppet/puppet.conf.
3. Выполнить команды:

```
systemctl enable puppet  
systemctl start --no-block puppet
```



После перезагрузки кассового ПО сервис Puppet продолжит свою работу.

Механизм обновления

Механизм обновления программного обеспечения построен таким образом, чтобы минимизировать время простоя кассового узла. Экономия времени на выполнение обновления достигается за счет того, что обновление ПО производится только после загрузки всех требуемых модулей на кассу.

Обновление программы выполняется в четыре этапа:

1. Инициация процесса загрузки обновленных модулей на кассу осуществляется через сервер управления конфигурацией.
2. Специальная служба на кассе производит закачку обновлений и контролирует корректность завершения этой операции. Для предотвращения чрезмерных нагрузок на сетевую инфраструктуру в пике массового обновления скорость закачки принудительно ограничивается.
3. После завершения закачки производится контроль целостности данных и, при необходимости, производится получение недостающих данных. Успешная проверка позволяет перейти к процессу установки.
4. Установка выполняется при перезапуске программного обеспечения. Как правило, этот процесс занимает менее двух минут (время зависит от аппаратных ресурсов и количества обновляемых модулей). Информация о наличии доступных обновлений отображается в верхней части экрана кассира. Для установки обновлений программа автоматически перезапускается:
 - после закрытия смены, если при запуске обновления системы на кассе имела открытая смена или чек;
 - сразу после получения данных о доступных обновлениях, если на кассе не работает пользователь, отсутствуют открытые чеки или смены.

Обновление/откат до заданной версии пакета выполняются по определенному алгоритму:

1. В файле /etc/apt/sources.list.d/artix.list указывается номер сборки, до которой требуется обновиться/откатиться:

```
deb http://update.artix.su/bionic artix-4.6.244-3 artix main contrib
```

2. Для получения информации о новых/обновлённых пакетах выполняется команда:

```
aptitude update
```

3. Необходимая версия пакета artix-upgrade-system2 устанавливается командой aptitude install artix-upgrade-system2=версия-номер_сборки, где версия и номер сборки должны совпадать с версией и номером сборки, указанными в /etc/apt/sources.list.d/artix.list.

```
aptitude install artix-upgrade-system2=4.6.244-3
```

При установке пакета `artix-upgrade-system2` на кассе создается директория `/linuxcash/cash/data/upgrade-system`, в которой будут размещены следующие файлы:

- `artix-packages.version` – включает список всех пакетов в сборке, до которой необходимо обновиться/откатиться;
- `artix-packages.instlist` – включает список пакетов с номерами версий, которые были загружены на кассу;
- `new-install` – файл-флаг, который необходимо создать для того, чтобы началась загрузка пакетов;
- `downloaded` – файл-флаг, будет создан после загрузки новых версий пакетов;
- `status.code` – будет создан после загрузки пакетов, отражает информацию о статусе загрузки/установки пакетов в виде соответствующего кода;
- `status.text` – будет создан после загрузки пакетов, отражает информацию о статусе загрузки/установки пакетов в виде текста.

4. В директории `/linuxcash/cash/data/upgrade-system` создается файл-флаг с именем `new-install`.

```
touch /linuxcash/cash/data/upgrade-system/new-install
```

5. В конфигурационный файл `/linuxcash/cash/conf/artix-packages.list` добавляется список пакетов, которые будут установлены при обновлении. Запись выполняется в формате: одна строка – один пакет. Пакеты указываются без версий.

```
artix-pos  
artix45-nes2
```

6. Скачивание и установка заданных пакетов выполняется при помощи скрипта `/linuxcash/cash/bin/artix-upgrade.sh`. Параметры скрипта задаются в файле `/linuxcash/cash/conf/artix-upgrade.conf`.

- `runsleeper` – интервал времени в минутах, в котором случайным образом задается начало обновления;
- `download_limit` = 25 – ограничение пропускной способности при скачивании обновлений, в КБ/с.

7. Касса перезапускается, после чего выполняется установка пакетов.

Оповещение об обновлении

Крупные торговые сети устанавливают регламентные процедуры для выполнения обновления кассовых программ в магазинах. Кассовое ПО Artix настроено таким образом, чтобы уведомлять пользователя о наличии доступных обновлений.

Для оповещения об обновлении и задания действия используется файл `/linuxcash/cash/data/tmp/notification.json`. Он представляет собой файл формата json, создаваемый модулем `artix-upgrade-system2`, после скачивания всех необходимых пакетов.

Пример файла `notification.json`

```
{  
  "message": "Доступны обновления",  
  "action": "restart"  
}
```

В зависимости от того, в каком состоянии находится кассовое ПО в момент появления возможности обновления, предусмотрено разное поведение программы при появлении сообщения:

- если при появлении сообщения о наличии доступных обновлений отсутствуют открытые смены или документы, то при нахождении в режиме авторизации или переходе к нему программа начнет установку обновлений и будет перезагружена через 10 секунд. На время выполнения работ на экран будет выведено сообщение о перезапуске программы;
- если при появлении сообщения о наличии доступных обновлений на кассе имеются открытые смены или документы, то установка обновлений будет произведена только при закрытии смены. После закрытия смены программа выполнит процедуру обновления и автоматически перезагрузит программу.



Файл `/linuxcash/cash/data/tmp/notification.json` может использоваться и в других ситуациях, когда необходимо оповещение пользователя и выполнение какого-либо действия. Подробнее об этом можно прочитать в разделе "[Обслуживание системы](#)".