

Задание правил разбора штрих-кодов в input.ini

- [Конфигурирование](#)
- [Принцип работы](#)
- [Примеры разбора](#)
- [Логирование](#)



Если не были применены правила разбора, заданные в конфигурационном файле `/linuxcash/cash/conf/ncash.ini.d/input.ini`, то будут применены правила разбора из файла `/linuxcash/cash/conf/bcode.ini`.

Задание правил разбора штрих-кодов в `input.ini` позволяет добиться большой гибкости при обработке последовательностей данных, отличающихся по структуре.

Конфигурирование

Для возможности задания правил разбора штрих-кодов в `input.ini` необходимо:

1. Активировать плагин `input` в конфигурационном файле `/linuxcash/cash/conf/ncash.ini.d/input.ini` в секции `[plugins]`.

Наименование	Тип данных	Возможные значения	Описание	Примечания
input	строковый	<ul style="list-style-type: none">• on• off	Активация плагина для разбора вводимых данных	По умолчанию on

Пример настройки

```
[plugins]
; Плагин разбора вводимых данных
input = on
```

2. Задать правила разбора и преобразований в конфигурационном файле `/linuxcash/cash/conf/ncash.ini.d/input.ini`.

Правила разбора представляют собой набор секций вида `[Rule.<порядок>]` или `[Rule.<порядок>.<имя захваченного значения>]`, где `<порядок>` – целое положительное число.

Пример

```
;[Rule.1]
; Правило разбора с порядком 1
...

;[Rule.1.code]
;Правило преобразования захваченного значения code
...

;[Rule.1.weight]
;Правило преобразования захваченного значения weight
...

;[Rule.2]
; Правило разбора с порядком 2
...

;[Rule.2.barcode]
;Правило преобразования захваченного значения barcode
...

;[Rule.2.weight]
;Правило преобразования захваченного значения weight
...
```

В каждой секции задается одно правило в формате:

```
search = <регулярное выражение со значениями для захвата>
```

В каждой секции вида [Rule.<порядок>.<имя захваченного значения>] задаются два правила в формате:

```
search = <регулярное выражение со значениями для захвата>
replace = <выражение для подстановки обработанных захваченных значений в исходное регулярное выражение>
```

Допускается использование следующих имен для захваченных значений:

- barcode – штрих-код товара,
- code – код товара,
- weight – вес товара,
- quantity – количество товара.

Пример настройки

```
;[Rule.1]
; Правило разбора с порядком 1
;search = "((?<piece>26)|(?<fractional>27))(?<code>[0-9]{10})(?(piece)(?<quantity>[0-9]{5})|(?<weight>[0-9]{5}))[0-9]{5};"

;[Rule.1.code]
;Правило преобразования захваченного значения code
;search = "0*([0-9]{0,10})"
;replace = "\1"

;[Rule.1.weight]
;Правило преобразования захваченного значения weight
;search = "0*([0-9]{0,2})([0-9]{3})"
;replace = "\1.\2"
```



Имена piece и fractional используются для захвата целых и дробных значений соответственно и могут быть заданы произвольно.



Значения для префиксов кодов или штрих-кодов (в примере выше – **26** и **27**) могут быть заданы произвольно.



Значения по умолчанию не заданы. Для корректной работы необходимо раскомментировать секции и параметры.

Принцип работы

Поиск подходящего правила разбора происходит по возрастанию порядка. Если введенная последовательность данных удовлетворяет правилу разбора, то будет осуществлен переход к разбору по секциям для захваченных значений. При корректных данных товары с введенными штрих-кодами (кодами) будут добавлены в чек.

Если введенные данные содержат как штрих-код товара так и код, то товар будет добавлен по штрих-коду.

Если введенные данные содержат как вес товара так и количество, то товар будет добавлен со значением веса.



Если последовательность содержит ошибочный код товара (или была ошибка при добавлении товара, например, при отсутствии товара в базе данных), то **все** позиции этой последовательности не будут добавлены в чек.

Примеры разбора

Пример 1. Разбор штрих-кодов

Правило разбора в input.ini:

[Rule.1]

```
search = "((?<piece>26)|(?<fractional>27))(?<barcode>[0-9]{20})(?(piece)(?<quantity>[0-9]{5})|(?<weight>[0-9]{5})),[0-9]{5};"
```

[Rule.1.weight]

```
search = "0*([0-9]{0,2})([0-9]{3})"
```

```
replace = "\\1.\\2"
```

Разбор:

При вводе **26460708092027100015,00020;27460708092024002315,00025**; строка будет разобрана следующим образом:

Реальный штрих-код первого товара = **4607080920271**,

Кол-во = 15 шт.

Реальный штрих-код второго товара = **4607080920240**,

Вес = 2.315 кг.

В чек будут добавлены товары со штрих-кодами **4607080920271** и **4607080920240**.



Последовательность введенных данных может содержать любое ненулевое количество штрих-кодов с неunikальными префиксами, расположенных в произвольном порядке. Например:

- **26460708092027100015,00020;**
- **27460708092024002315,00025;26460708092027100015,00020;27460708092024402315,00025;**



В качестве количества товара может быть использовано значение из quantity или weight, стоящее до или после запятой в регулярном выражении. Для этого необходимо изменить правило разбора.

Пример 2. Разбор кода

Правило разбора в input.ini:

[Rule.1]

```
search = "((?<piece>301)|(?<fractional>401))(?<code>[0-9]{10})[0-9]{5},(?(piece)(?<quantity>[0-9]{5})|(?<weight>[0-9]{5}));"
```

[Rule.1.code]

```
search = "0*([0-9]{0,10})"
```

```
replace = "\\1"
```

Разбор:

При вводе **301000001006800010,00020**; строка будет разобрана следующим образом:

Реальный код (с отсеченными лидирующими нулями) = **10068**,

Кол-во товара = 20 шт.

В чек будет добавлен товар с кодом **10068**.



В кассовом ПО Artix реализована возможность использования правил интерпретации штрих-кодов при считывании карты через ридер и при добавлении купона по кнопке. Подробнее об этом можно прочитать в статье ["Использование правил интерпретации штрих-кодов"](#).

Логирование

Логирование работы плагина ведется в файле `/linuxcash/logs/current/terminal.log`.