



Специфікація використання  
**драйвера Cashälot для BAS**

v. 4.2

# Зміст

1. Зміст
2. Поняття і визначення
3. Загальна інформація
4. Апаратні та програмні вимоги
5. Опис інтерфейсу драйвера Cashälot для BAS
6. Порядок виклику функцій при виконанні касових операцій
7. Отримання номера версії драйвера
8. Отримання налаштувань драйвера
9. Отримання останньої помилки
10. Встановлення налаштувань драйвера
11. Підключення фіскального реєстратора ПРРО Cashälot
12. Відключення фіскального реєстратора ПРРО Cashälot
13. Перевірка ПРРО Cashälot
14. Відкриття зміни
15. Реєстрація чеків
16. Відміна відкритого чека
17. Вивести візуальне відображення чека на екран
18. Отримання фіскального номера чека за його локальним номером
19. Отримання локального номера чека за його фіскальним номером
20. Отримання xml структури фіскального чека
21. Друк періодичних звітів
22. Отримання інформації про поточний стан ПРРО
23. Друк довільної текстової інформації
24. Службове внесення/видача
25. Формування X-звіту
26. Формування Z-звіту
27. Закриття зміни

## Поняття і визначення

**КЕП** – кваліфікований електронний підпис.

**Локальний номер чека** – номер, призначений чеку системою виписки електронних касових чеків, що використовується продавцем.

**Офлайн сесія** – сукупність документів, створених в режимі офлайн, між припиненням і відновленням зв'язку ПРРО з ФСКО.

**ПРРО** – програмний реєстратор розрахункових операцій. Система виписки електронних касових чеків, що використовується продавцем.

**Фіскальний номер чека (ФН)** – номер, призначений чеку сервером ФСКО.

**ФСКО** – фіскальний сервер контролюючого органу.

## Загальна інформація

**Драйвер Cashälot** використовується для інтеграції ПРРО з розрахунково-обліковими системами 1С/BAS. В цьому документі приведено специфікацію методів бібліотек Cashälot для підключення до 1С/BAS.

За допомогою інтеграції безпосередньо з розрахунково-облікової системи можна скористатися наступним функціоналом програмного забезпечення Cashälot:

- Відкриття та закриття касової зміни;
- Створення та реєстрація чеків продажу/повернення;
- Створення та реєстрація чеків з типами оплати передоплата, післяплата, інтернет-продажа;
- Створення видаткового чека на основі чека продажу та його реєстрація;
- Формування та друк Z-звіту за даними фіскального сервера контролюючого органу (скорочено ФСКО)/X-звіту за даними ПРРО;
- Створення та реєстрація службових чеків;
- Службове внесення;
- Службова видача;
- Робота в режимі офлайн;
- Можливість відображення на екрані зареєстрованого чека на пристрої продавця з QR-кодом після успішної реєстрації;
- Друк зареєстрованого чека;
- Формування періодичних звітів.

## Апаратні та програмні вимоги

Для оптимальної роботи ПРРО Cashälot рекомендовані наступні системні та програмні вимоги:

### **Операційна система**

Програмний комплекс коректно функціонує на комп'ютерах з сучасними операційними системами Microsoft та компонентами, що входять до її складу. Рекомендовано використовувати останню версію ОС, наприклад, Windows 10 або Windows Server 2019 відповідно до потреб, з встановленими актуальними оновленнями.

### **Апаратне забезпечення:**

- процесор з мінімальною тактовою частотою від 2 ГГц;
- оперативна пам'ять від 2 GB;
- вільне місце на жорсткому диску від 1,5 GB;
- кольоровий графічний дисплей;
- маніпулятор типу миша та клавіатура;
- встановлений український або російський мовний стандарт регіональних налаштувань операційної системи;
- підтримка кирилиці в операційній системі;
- доступ до поштового сервера;
- наявний зв'язок з фіскальним сервером контролюючого органу.

### **Мережа**

Для роботи користувача необхідний доступ до мережі інтернет. Також, необхідно мати доступ до кабінету Cashalot, фіскального сервера контролюючого органу та до ресурсу центру сертифікації ключів. Наприклад, якщо ви використовуєте ключі від "АЦСК Україна", необхідно надати доступ до ресурсу: <https://uakey.com.ua/> - 212.90.186.150 - 80/443. Якщо ж ви використовуєте КЕП (ЕЦП) інших кваліфікованих надавачів (АЦСК), то необхідно надати доступ до серверів TSP, OCSP, CMP відповідних центрів.

#### *Загальний список ресурсів:*

- <https://load.cashalot.org.ua/update/> - 94.131.247.22:443 - Сервіс завантаження інсталяцій програми (дистрибутиви, оновлення);
- <http://fs.tax.gov.ua:8609/fs/cmd> - Сервіс API ДПС;
- <https://my.cashalot.org.ua/> - 94.131.247.21:443 - Особистий кабінет Cashalot.

# Опис інтерфейсу драйвера Cashälot для BAS

(компоненти CashaLotApiDrv32.dll/CashaLotApiDrv64.dll)

Драйвер Cashälot - це окремий механізм для інтеграції ПРРО Cashälot, що призначений для використання виключно в 1С/BAS. Для інтеграції використовуються бібліотеки **CashaLotApiDrv32.dll** та **CashaLotApiDrv64.dll** (далі - CashaLotApiDrv), які розміщені в кореновому каталозі ПРРО Cashälot.

В конфігураторі 1С/BAS для драйвера фіскального реєстратора вказуються методи, призначені для розробки драйверів обладнання, що підключається. Ці методи відправляють запити і отримують відповіді від драйвера **CashaLotApiDrv**.

**CashaLotApiDrv** є в поставці 1С/BAS, починаючи з версій:

- Управління виробничим підприємством (1.3.75.2)
- Управління торговим підприємством для України (1.2.64.1)
- Управління торгівлею для України (2.3.37.1)
- Роздріб для України (1.0.14.1)
- BAS Малий бізнес (Small company) PROF (1.6.15.1)
- BAS Управління торгівлею (Trade management) (3.2.23.2)
- BAS Роздрібна торгівля (Retail) (2.2.12.1)
- BAS ERP (Enterprise resource planning) (2.1.23.2)
- BAS Комплексне управління підприємством (Integrated enterprise management) (2.1.23.2)

Перед початком роботи з **CashalotApiDrv** потрібно:

- зареєструватися в особистому кабінеті <https://my.cashalot.org.ua/>;
- виконати всі пункти з розділу “Початок роботи” в особистому кабінеті;
- встановити ПРРО Cashälot, попередньо завантаживши дистрибутив з офіційного [сайту](#). Виконати запуск програми Cashälot (В момент першого запуску завантажуються дані з бек-офісу). Для паралельної роботи в декількох касах одночасно необхідно встановлювати ПРРО Cashälot для кожної каси окремо;
- зареєструвати бібліотеки в ОС за допомогою команди **regsvr32**

**regsvr32 “<Повний шлях до ПРРО Cashalot>\CashalotApiDrv32.dll”**

(для 32-бітної платформи)

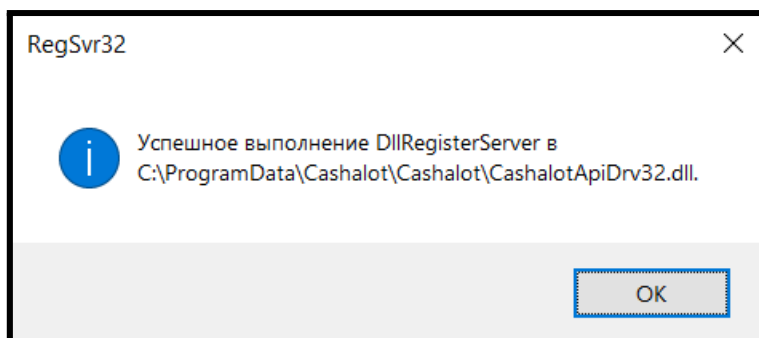
**regsvr32 “<Повний шлях до ПРРО Cashalot>\CashalotApiDrv64.dll”**

(для 64-бітної платформи)

Наприклад:

regsvr32 “C:\ProgramData\Cashalot\Cashalot\CashalotApiDrv32.dll”

Після успішної реєстрації бібліотеки з’явиться повідомлення зразку:



В CashalotApiDrv реалізована обробка наступних методів:

- ПолучитьНомерВерсии
- ПолучитьОшибку
- ПолучитьПараметры
- УстановитьПараметр
- Подключить
- Отключить
- ТестУстройства
- ОткрытьСмену
- ОткрытьЧек
- НапечататьФискальнуюСтроку
- НапечататьФискСтроку
- УстановитьРасширенныйПорядокОплатыЧека
- ЗакрытьЧек
- ОтменитьЧек
- ПолучитьXMLЧека
- ОтобразитьЧек
- ПолучитьФискальныйНомерЧека
- ПолучитьЛокальныйНомерЧека
- НапечататьПериодическийОтчетПоДатам
- НапечататьЧекВнесенияВыемки
- НапечататьОтчетБезГашения
- НапечататьОтчетСГашением
- ЗакрытьСмену

Ініціалізація компоненти відбувається шляхом створення об'єкту відповідно до синтаксису:

```
&НаКлиенте
Перем ДрайверCashalot Экспорт;
&НаКлиенте
Процедура ИнициализацияДрайвераCashalot ()

    БиблиотекаДрайвера = "Путь к библиотеке драйвера
CashalotApiDrv32.dll/CashalotApiDrv64.dll";

    Попытка
        ПодключитьВнешнююКомпоненту (БиблиотекаДрайвера, "CashalotApiDrv");
        ДрайверCashalot = Новый ("AddIn.CashalotApiDrv.CashalotApiDrv");
        Сообщить ("Драйвер Cashalot инициализирован.");
    Исключение
        Сообщить ("Не удалось инициализировать драйвер Cashalot." +
            Символы.ПС +
            ОписаниеОшибки ());
    КонецПопытки;

КонецПроцедуры
```

При роботі одного користувача з декількома касами необхідно встановити налаштування для фіскального реєстратора таким чином, щоб драйвер мав можливість розрізнити різні екземпляри ПРРО Cashalot, тобто для кожної каси свій фіскальний номер (**DeviceIDFnRRO**) та встановлений окремий екземпляр програми Cashalot, до якого вказаний шлях в параметрах драйвера (**PathToCashalotDir**).

Ім'я	Тип	Опис
DeviceIDFnRRO	строка	Фіскальний номер ПРРО.
PathToCashalotDir	строка	Шлях до каталогу з встановленим ПРРО Cashalot.

## Порядок виклику функцій при виконанні касових операцій

При роботі з касою в системі BAS рекомендується послідовність виклику функцій ПРРО Cashalot, зазначена в таблиці:

Касова операція	Послідовність функцій драйвера
<b>Відкриття зміни</b>	ОткрытьСмену
<b>Внесення грошей</b>	НапечататьЧекВнесенияВыемки
<b>Фіскалізація чеків</b>	ОткрытьЧек НапечататьФискальнуюСтроку або НапечататьФискСтроку УстановитьРасширенныйПорядокОплатыЧека ЗакрытьЧек або ОтменитьЧек
<b>Повернення товару</b>	ОткрытьЧек НапечататьФискальнуюСтроку або НапечататьФискСтроку УстановитьРасширенныйПорядокОплатыЧека ЗакрытьЧек або ОтменитьЧек
<b>X-звіт</b>	НапечататьОтчетБезГашения
<b>Видача грошей</b>	НапечататьЧекВнесенияВыемки
<b>Z-звіт</b>	НапечататьОтчетСГашением
<b>Закриття зміни</b>	ЗакрытьСмену

### 1. Отримання номера версії драйвера

Для отримання номера версії драйвера використовується метод:

**СТРОКА** `ПолучитьНомерВерсии()`

**Приклад:**

```

Функция ПолучитьВерсиюДрайвераCashalot()

    ВерсияДрайвераCashalot = ДрайверCashalot.ПолучитьНомерВерсии();
    Сообщить ("Версия драйвера Cashalot: " + ВерсияДрайвераCashalot);
    Возврат ВерсияДрайвераCashalot;

КонецФункции

```

### 2. Отримання налаштувань драйвера

Для отримання списку параметрів налаштувань драйвера використовують функцію:

**БУЛЕВО** `ПолучитьПараметры()`.

Параметри функції для отримання параметрів драйвера:

Ім'я	Тип	Опис
XMLСтрокаПараметров	строка [out]	Опис параметрів. Таблиця списку параметрів є XML-рядком.

**Увага!** Функція не повертає поточні значення параметрів.

**Приклад:**

```
Процедура ПолучитьПараметрыCashalot ()

XMLСтрокаПараметров = "";
ДрайверCashalot.ПолучитьПараметры (XMLСтрокаПараметров);

ЧтениеXML = Новый ЧтениеXML ();
ЧтениеXML.УстановитьСтроку (XMLСтрокаПараметров);
Пока ЧтениеXML.Прочитать () Цикл
    Если ЧтениеXML.ТипУзла = ТипУзлаXML.НачалоЭлемента И ЧтениеXML.Имя =
        "Parameter"
    Тогда
        ИмяПараметра = ЧтениеXML.ПолучитьАтрибут ("Name");
        ОписаниеПараметра = ЧтениеXML.ПолучитьАтрибут ("Caption");
        Сообщить ("Параметр драйвера "" + ИмяПараметра + "": " +
            ОписаниеПараметра + ".");
    КонецЕсли;
КонецЦикла;

ЧтениеXML.Закреть ();

КонецПроцедуры
```

### 3. Отримання останньої помилки

Для отримання опису помилки використовується функція **ПолучитьОшибку()**. У функції повинен бути вказаний параметр з іменем `ТекстОшибкиCashalot` та типом `строка [out]`.

**Приклад:**

```
Функция ПолучитьОшибкуCashalot ()

ТекстОшибкиCashalot = "";
ДрайверCashalot.ПолучитьОшибку (ТекстОшибкиCashalot);

Возврат ТекстОшибкиCashalot;

КонецФункции
```

### 4. Встановлення налаштувань драйвера

Функція призначена для встановлення значень параметрів налаштувань драйвера Cashalot для BAS. **БУЛЕВО УстановитьПараметр**(ИмяПараметра, ЗначениеПараметра)

Кожен параметр повинен містити в собі:

Ім'я	Тип	Опис
ИмяПараметра	строка	Ім'я параметра.
ЗначениеПараметра	строка	Значення параметра.

**УстановитьПараметр** складається з обов'язкових параметрів (виділені червоним) та необов'язкових параметрів. Параметри повинні мати послідовність, як в таблиці нижче.



При роботі з **CashalotApiDrv**, необхідно зазначити такі параметри:

Ім'я параметра	Значення параметра
PathToCashalotDir	Шлях до каталогу з встановленим ПРРО Cashalot.
DeviceIDFnRRO	Фіскальний номер ПРРО.
NOINTERFACEMODE	<p>Ознака використання інтерфейсу РМК для авторизації та службових операцій видачі та внесення готівки в касу.</p> <p>"False" – при роботі з драйвером викликається діалогове вікно.</p> <p>"True" – авторизація відбувається шляхом передачі параметрів авторизації PathToCertificate та PwdToCertificate. Вікна службових операцій видачі та внесення готівки не викликаються (проте, результат операції службового внесення/видачі готівки відображається).</p>
PathToCertificate	<p>Шлях до каталогу з файловим ключем користувача.</p> <p>Важливо: в папці повинні міститися ключ і сертифікат тільки одного користувача. Якщо касою користуються декілька касирів, необхідно помістити їх сертифікати і ключі в окремі каталоги.</p>
PwdToCertificate	Пароль від ключа.
NOAUTOOPENSHIFT	<p>Перевірка стану зміни (відкрита або закрита). Якщо зміна закрита, при виклику будь-якої функції зміна автоматично відкривається.</p> <p>"False" – автоматично відкривати зміну, якщо вона закрита.</p> <p>"True" – не відкривати автоматично зміну. В цьому випадку робота функцій буде перервана і результат виконання буде повернуто у вигляді помилки.</p>
DEFAULTPRINTERNAME	<p>Визначає принтер для друку при натисканні кнопки "Друк" в діалогових вікнах:</p> <ol style="list-style-type: none"> <li>1) "" – пустий рядок, виводиться діалог вибору принтера;</li> <li>2) "DEFAULTWINDOWS" – використовувати принтер Windows за замовчуванням;</li> <li>3) "ім'я принтера чеків" – в параметр передається ім'я принтера чеків, який буде використовуватись для друку чеків в ПРРО Cashalot.</li> </ol>
AUTOPRINTMODE	<p>Параметр визначає автоматичну відправку на друк чека або службового звіту без відображення інформаційного вікна:</p> <p>"True" – відправляти на друк;</p> <p>"False" – не відправляти на друк.</p> <p>Якщо задано принтер за замовчуванням (DEFAULTPRINTERNAME), чек друкується одразу, в іншому випадку відображається вікно вибору принтера для друку.</p>

### Приклад:

```
Процедура УстановитьПараметрCashalot (ИмяПараметра, ЗначениеПараметра)

    ДрайверCashalot = ИнициализацияДрайвераCashalot ();
    ИмяПараметра = "DeviceIDFnRRO";
    ЗначениеПараметра = "4000000000";
    ПараметрУстановлен = ДрайверCashalot.УстановитьПараметр (ИмяПараметра,
ЗначениеПараметра);

    Если НЕ ПараметрУстановлен Тогда
        Сообщить ("Не удалось установить параметр Cashalot "" +
            ИмяПараметра +
            "" из-за ошибки: " +
            ПолучитьОшибкуCashalot ());
    КонецЕсли;

КонецПроцедуры
```

## 5. Підключення фіскального реєстратора ПРРО Cashälot

Для підключення ПРРО Cashälot з поточними значенням параметрів, встановлених функцією **УстановитьПараметр**, використовується функція:

**БУЛЕВО** **Подключить**(ФНПодключеннойКассы)

Параметри функції підключення пристрою:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.

При успішному підключенні, функція повертає ідентифікатор підключеного ПРРО.

### Приклад:

```
Функция ПодключитьCashalot ()

    ФНПодключеннойКассы = "";
    CashalotПодключен = ДрайверCashalot.Подключить (ФНПодключеннойКассы);

    Если CashalotПодключен Тогда
        Сообщить ("Драйвер подключен к ПРРО Cashalot и кассе с ФН: " +
            ФНПодключеннойКассы);
    Иначе
        Сообщить ("Не удалось подключить драйвер к ПРРО Cashalot из-за
            ошибки: " + ПолучитьОшибкуCashalot ());
    КонецЕсли;

    Возврат CashalotПодключен;

КонецФункции
```

## 6. Відключення фіскального реєстратора ПРРО Cashälot

Для відключення ПРРО Cashälot використовується функція:

**БУЛЕВО Отключить**(ФНПодключеннойКассы)

Параметри функції відключення пристрою:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.

**Приклад:**

```
Функция ОтключитьCashalot ()

    CashalotОтключен = ДрайверCashalot.Отключить (ФНПодключеннойКассы);

    Сообщить (? (CashalotОтключен,
        "Драйвер отключен от ПРРО Cashalot.",
        "Не удалось отключить драйвер от ПРРО Cashalot из-за ошибки:"
        + ПолучитьОшибкаCashalot ())) );

    Возврат CashalotОтключен;

КонецФункции
```

## 7. Перевірка ПРРО Cashälot

Сценарій, в якому функція використовується користувачем для перевірки одного чи декількох ПРРО. Перевірка виконується шляхом пробного підключення і опитування пристрою з поточними значеннями параметрів, встановлених для кожного екземпляру окремо. Використовується функція:

**БУЛЕВО ТестУстройства** (Описание, ПризнакДемоРежима )

Параметри:

Ім'я	Тип	Опис
Описание	строка [out]	Опис результату виконання тесту.
ПризнакДемоРежима	строка [out]	Опис обмежень демонстраційного режиму, за його наявності, і пустий результат, при його відсутності.

**Увага!** На даному етапі розробки результат виконання функції не містить значень вихідних параметрів.

**Приклад:**

```
Процедура ТестУстройстваCashalot ()

    ТестУспешен = ДрайверCashalot.ТестУстройства ("", "");

    Сообщить (? (ТестУспешен,
        "Тестирование подключения к ПРРО Cashalot выполнено
        успешно.",
        "Тестирование подключения к ПРРО Cashalot неудачно."));

КонецПроцедуры
```

## 8. Відкриття зміни

Сценарій, в якому функція використовується користувачем при відкритті зміни для одного ПРРО чи для декількох екземплярів ПРРО, в залежності від налаштувань. Для кожного екземпляру окремо необхідно вказати обов'язкові значення параметра ФНПодключеннойКассы, щоб мати змогу застосовувати функцію

### БУЛЕВО ОткрытьСмену(ФНПодключеннойКассы)

При виклику функції (наприклад: фіскалізації чека), якщо зміна не відкрита, відбувається автоматичне відкриття зміни з викликом функції **ОткрытьСмену**. Для запобігання автоматичного відкриття зміни за допомогою функції **УстановитьПараметр** встановіть параметр:

**NOAUTOOPENSHIFT** = "True"

При виконанні функції **ОткрытьСмену** драйвер ПРРО Cashalot викликає вікно авторизації, в якому користувач зазначає папку з файловим ключем та сертифікатом касира або старшого касира, пароль, після чого відбувається відкриття касової зміни.

*Параметри функції відкриття зміни:*

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.

### Приклад:

```
Процедура ОткрытьСменуCashalot ()  
  
    ФНПодключеннойКассы = "4000000000";  
    СменаОткрыта = ДрайверCashalot.ОткрытьСмену (ФНПодключеннойКассы) ;  
  
    Сообщить (? (СменаОткрыта,  
                "Смена открыта.",  
                "Не удалось открыть смену из-за ошибки: " +  
                ПолучитьОшибкуCashalot ())) ;  
  
КонецПроцедуры
```

## 9. Реєстрація чеків

При реєстрації чеків дані передаються через ланцюжок функцій, призначених для друку чеків на фіскальних реєстраторах в такому порядку:

1. **БУЛЕВО ОткрытьЧек**(ФНПодключеннойКассы, ПризнакФискальногоЧека, ПризнакЧекаВозврата, АннулироватьПредыдущийОткрытыйЧек, НомерОткрытогоЧека, НомерСмены);
2. **БУЛЕВО НапечататьФискальнуюСтроку**(ФНПодключеннойКассы, НаименованиеНоменклатуры, КодУКТВЭД, Артикул, Штрихкод, ЕдиницаИзмерения, НомерСекции, Количество, Цена, Сумма, ПроцентСкидки, СуммаСкидки, СтавкаНДС, ЦенаВключаетНДС, СуммаНДС, ПризнакАкцизногоНалога, Номер строки, ДополнительныеРеквизиты);  
**або**  
**БУЛЕВО НапечататьФискСтроку**(ФНПодключеннойКассы, НаименованиеНоменклатуры, Количество, Цена, Сумма, НомерСекции, СтавкаНДС, ДополнительныеРеквизиты);
3. **БУЛЕВО НапечататьНефискСтроку**(ФНПодключеннойКассы, КомментарийКЧеку);
4. **БУЛЕВО УстановитьРасширенныйПорядокОплатыЧека** (ФНПодключеннойКассы, ТипОплаты, СуммаПредоплаты, НомерЗаказа, ФискальныйНомерЧекаПредоплатыИлиВозврата, ФискальныйНомерКассыВозврата, ДопРеквизиты);
5. **БУЛЕВО ЗакретьЧек** (ФНПодключеннойКассы, СуммаОплатыНаличными, СуммаОплатыКартой, СуммаОплатыКредитом, СуммаОплатыСертификатом, СуммаКОплате, ФискальныйНомерЧекаДляВозврата, НомерОткрытогоЧека, ФискальныйНомерЧека);

Функція **ОткрытьЧек** формує заголовок чека.

Функції **НапечататьФискальнуюСтроку** та **НапечататьФискСтроку** призначені для заповнення чека рядками товарами, що купуються. Ці функції рівноцінні, проте, перша з них має повний перелік параметрів товару, а друга має скорочений перелік. Обидві функції мають параметр **ДополнительныеРеквизиты**, в якому передається сформований XML-рядок з переліком реквізитів товару. Реквізити, які зазначені в XML-рядку, мають пріоритет над явно заданими параметрами функції, тому рекомендується використовувати **ДополнительныеРеквизиты**, а також функцію **НапечататьФискСтроку** з використанням цього параметра. Крім того, у випадку використання функції **НапечататьФискСтроку**, в **ДополнительныеРеквизиты** необхідно обов'язково вказати значення для артикула та одиниці вимірювання товару.

Функція **ЗакретьЧек** виконує завершення формування чека, передає дані про оплату в ПРРО Cashalot, де чек підписується і відправляється на сервер ФСКО для реєстрації, та повертає як кінцеві параметри фіскальний номер зареєстрованого чека та номер для наступного чека. Порушення послідовності функцій призведе до помилки реєстрації чеків.

Для відображення коментаря в чеку необхідно застосувати метод **НапечататьНефискСтроку**. При побудові нефіскального рядка необхідно врахувати ширину чекової стрічки та допустиму кількість символів в одному рядку. Якщо необхідно надрукувати декілька нефіскальних рядків, необхідно заповнити параметр **КоментарийКЧеку** у вигляді:

```
ПереносСтроки = "\n";  
КоментарийКЧеку = "Перший рядок коментаря" + ПереносСтроки + "Другий рядок  
коментаря" + ПереносСтроки + "Останній рядок коментаря";
```

Для встановлення розширеного порядку сплати поточного чека, використовується функція **УстановитьРасширенныйПорядокОплатыЧека**. Для різних варіантів потрібно встановити **ТипОплаты = 0, 1, 2, 3**.

У випадку формування чека передоплати (**ТипОплаты = 1**), заповнити поле суми передоплати **СуммаПредоплаты**, сума післяплати буде перерахована автоматично, залежно від суми всіх рядків чека за вирахуванням суми передоплати.

У чеку післяплати (**ТипОплаты = 2**) потрібно вказати фіскальний номер чека, за яким була здійснена передоплата, **ФискальныйНомерЧекаПредоплатыИлиВозврата** та фіскальний номер каси, на якій була передоплата, якщо він відрізняється. Якщо не відрізняється, встановити значення null або порожній рядок.

Також функцію **УстановитьРасширенныйПорядокОплатыЧека** можна використовувати для повернення чеків, які були сформовані в інших ПРРО та заповнення додаткової інформації в чек про термінал, при оплаті карткою, за допомогою параметра **ДопРеквизиты**.

**Заокруглення суми** по чеку реалізовано наступним чином:

В функцію **ЗакретьЧек** передається значення готівки, що надана клієнтом на касі, значення оплати картою, сертифікатом та кредитом, а також, заокруглена сума, розрахована за певним алгоритмом користувача **СуммаКОплате** (наприклад, відповідно до [Постанови НБУ п.4](#), 1-4 коп заокруглюється до 0 коп, 5-9 коп заокруглюється до 10 коп). В чеку буде відображено заокруглення, надану готівку та інші форми оплати, заокруглену суму до сплати, решту. У випадку, якщо параметр **СуммаКОплате** не визначений або дорівнює нулю, то він замінюється сумою по всім формам оплати. В такому випадку заокруглення буде дорівнювати нулю.

**Решта** по чеку розраховується так:

**Сума решти = СуммаОплатыНаличными + СуммаОплатыКартой + СуммаОплатыКредитом + СуммаОплатыСертификатом - СуммаКОплате**

Решта розраховується програмою CashÄlot автоматично, згідно вказаних користувачем даних.

### **Повернення**

Для створення та фіскалізації чека повернення товарів та/або послуг необхідно сформувати ланцюжок з функцій для реєстрації чеків.

В функції **ОткрытьЧек** необхідно встановити **ПризнакЧекаВозврата = Истина**, а в функціях **ЗакретьЧек** та, за необхідності, **УстановитьРасширенныйПорядокОплатыЧека** вказати ФН чека, на основі якого формується повернення, та ФН каси, де був сформований чек, у відповідних параметрах.

Опис параметрів функцій для формування чека вказані в таблиці:

ОткрытьЧек		
Ім'я параметра	Тип	Опис параметра
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.
ПризнакФискальногоЧека	булево	Ознака фіскального чека
ПризнакЧекаВозврата	булево	Ознака чека повернення
АннулироватьПредыдущийОткрытыйЧек	булево	Ознака автоматичного анулювання раніше відкритого чека
НомерОткрытогоЧека	число [out]	Номер нового чека
НомерСмены	число [out]	Номер відкритої зміни
НапечататьФискальнуюСтроку		
Ім'я параметра	Тип	Опис параметра
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.
НаименованиеНоменклатуры	строка	Найменування товару. Допускається довжина до 127 символів
КодУКТВЭД	строка	Код УКТЗЕД
Артикул	строка	Артикул. Допускається довжина до 63 символів.
Штрихкод	строка	Штрих-код
ЕдиницаИзмерения	строка	Одиниця виміру
НомерСекции	строка	Номер секції (не використовується)
Количество	число	Кількість товару
Цена	число	Ціна одиниці товару
Сумма	число	Сума по позиції чека
ПроцентСкидки	число	Процент знижки (ігнорується, якщо заповнена сума знижки)
СуммаСкидки	число	Сума знижки на позицію
СтавкаНДС	число	ПДВ у процентах (від 0 до 99,99)
ЦенаВключаетНДС	булево	Ознака, що сума містить ПДВ
СуммаНДС	число	Сума ПДВ за одиницю товару
ПризнакАкцизногоНалога	булево	Ознака підакцизного товару.
НомерСтроки	булево	Номер рядка (не використовується)
ДополнительныеРеквизиты	строка	Додаткові параметри (XML таблиця)
НапечататьФискСтроку		
Ім'я параметра	Тип	Опис параметра
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.
НаименованиеНоменклатуры	строка	Найменування товару. Допускається довжина до 127 символів
Количество	число	Кількість товару

Цена	число	Ціна одиниці товару
Сумма	число	Сума по позиції чека
НомерСекции	строка	Номер секції (не використовується)
СтавкаНДС	число	ПДВ у процентах (від 0 до 99,99)
ДополнительныеРеквизиты	строка	Додаткові параметри (XML таблиця)
<b>НапечататьНефискСтроку</b>		
<b>Имя параметра</b>	<b>Тип</b>	<b>Описание параметра</b>
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.
КомментарийКЧеку	строка	Нефіскальний рядок, коментар.
<b>УстановитьРасширенныйПорядокОплатыЧека</b>		
<b>Имя параметра</b>	<b>Тип</b>	<b>Описание параметра</b>
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.
ТипОплаты	число	Спосіб оплати: 0 - звичайна, 1 - передплата, 2 - післяплата, 3 - інтернет продаж
СуммаПредоплаты	число	Сума попередньої оплати.
НомерЗаказа	строка	Номер замовлення/договору.
ФискальныйНомерЧекаПредоплатыИлиВозврата	строка	Фіскальний номер чека, по якому здійснюється доплата/повернення.
ФискальныйНомерКассыВозврата	строка	Фіскальний номер РРО, по якому здійснюється доплата/повернення.
ДопРеквизиты	строка	Додаткові параметри у вигляді XML таблиці.
<b>ЗакретьЧек</b>		
<b>Имя параметра</b>	<b>Тип</b>	<b>Описание параметра</b>
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.
СуммаОплатыНаличными	число	Сума оплати готівкою
СуммаОплатыКартой	число	Сума оплати картою
СуммаОплатыКредитом	число	Сума оплати кредитом
СуммаОплатыСертификатом	число	Сума оплати сертифікатом
СуммаКОплате	число	Сума до оплати, що враховує всі форми оплати, знижки та заокруглення
ФискальныйНомерЧекаДляВозврата	строка	Фіскальний номер чека, за яким виконується повернення
НомерОткрытогоЧека	число [out]	Номер нового чека
ФискальныйНомерЧека	строка[out]	Фіскальний номер нового чека

**ДополнительныеРеквизиты** та **ДопРеквизиты**. Текст в форматі XML з кодуванням UTF-8, що передається за допомогою параметрів з типом “строка”.



**Увага!** Параметри рядка чека, які вказані в **ДополнительныеРеквизиты**, мають пріоритет над параметрами, які передані в функціях **НапечататьФискальнуюСтроку** або **НапечататьФискСтроку**.

Атрибути XML-рядка параметра **ДополнительныеРеквизиты**:

Атрибут XML-рядка	Опис атрибута
Nomenclature *	Найменування товару. Допускається довжина до 127 символів
UKTZED	Код УКТЗЕД
VendorCode *	Артикул, допускається довжина до 63 символів
Barcode	Штрихкод
Dimension *	Одиниця вимірювання
Discount	Процент знижки (ігнорується, якщо заповнена сума знижки)
DiscountAmount	Сума знижки
VAT *	Ставка ПДВ у процентах
VATLetter	Літера ставки ПДВ
PriceIncludesVAT або PriceIncludeVAT	Для випадків, коли сума по позиції не містить суму ПДВ (true/false)
AmountVAT	Сума ПДВ
IsExcisable	Ознака підакцизного товару, True/False
ExciseLetter	Літера ставки акцизного податку
ExciseStampBarcode	Штрих-код марки акцизного податку. У випадку коли товарів більше ніж 1, по одній позиції номенклатури, то всі штрих-коди заповнюються через кому. Поле ігнорується, якщо ознака підакцизного товару <b>IsExcise</b> = false.

\* обов'язкові параметри в рядку **ДополнительныеРеквизиты**, якщо не вказані відповідні явні параметри.

#### Приклад формування **ДополнительныеРеквизиты**:

Процедура **СформироватьXMLДополнительныхРеквизитов (ПозицияЧека)**

```

ЗаписьXML = Новый ЗаписьXML;
ЗаписьXML.УстановитьСтроку ("utf-8");
ЗаписьXML.ЗаписатьОбъявлениеXML ();

ЗаписьXML.ЗаписатьНачалоЭлемента ("Table");
    ЗаписьXML.ЗаписатьНачалоЭлемента ("Record");
        ЗаписьXML.ЗаписатьАтрибут ("NOMENCLATURE",
            ПозицияЧека.НаименованиеНоменклатуры);
        ЗаписьXML.ЗаписатьАтрибут ("UKTZED", ПозицияЧека.КодУКТВЭД);
        ЗаписьXML.ЗаписатьАтрибут ("VENDORCODE", ПозицияЧека.Артикул);
        ЗаписьXML.ЗаписатьАтрибут ("BARCODE", ПозицияЧека.Штрихкод);
        ЗаписьXML.ЗаписатьАтрибут ("DIMENSION",
            ПозицияЧека.ЕдиницаИзмерения);
        ЗаписьXML.ЗаписатьАтрибут ("DISCOUNT",
            Строка (ПозицияЧека.ПроцентСкидки));
        ЗаписьXML.ЗаписатьАтрибут ("DISCOUNTAMOUNT",
            Строка (ПозицияЧека.СуммаСкидки));
        ЗаписьXML.ЗаписатьАтрибут ("VAT", Строка (ПозицияЧека.СтавкаНДС));
        ЗаписьXML.ЗаписатьАтрибут ("VATLetter",

```

```

        Строка (ПозицияЧека.БукваСтавкиНДС) );
        ЗаписьXML.ЗаписатьАтрибут ("PRICEINCLUDEVAT",
        ?(ПозицияЧека.ЦенаВключаетНДС, "True", "False"));
        ЗаписьXML.ЗаписатьАтрибут ("AMOUNTVAT",
        Строка (ПозицияЧека.СуммаНДС) );
        ЗаписьXML.ЗаписатьАтрибут ("ISEXCISABLE",
        ?(ПозицияЧека.ПризнакАкцизногоНалого, "True", "False"));
        ЗаписьXML.ЗаписатьАтрибут ("ExciseLetter",
        Строка (ПозицияЧека.БукваСтавкиАкциз) );
        ЗаписьXML.ЗаписатьАтрибут ("EXCISESTAMPBARCODE",
        ПозицияЧека.ШтрихкодАкцизнойМарки) );
        ЗаписьXML.ЗаписатьКонецЭлемента ();
        ЗаписьXML.ЗаписатьКонецЭлемента ();

        ПозицияЧека.ДополнительныеРеквизиты = ЗаписьXML.Закреть ();

```

КонецПроцедуры

Атрибути XML-рядка параметра **ДопРеквизиты** при використанні оплати картою через термінал для методу **УстановитьРасширенныйПорядокОплатыЧека**:

Атрибут XML-рядка	Опис атрибута
TerminalID	Код терміналу.
ApprovalCode	Код підтвердження.
RRN	РРН транзакції.
IssuerName	Платіжна система.
PAN	Номер картки.
TransactionDate	Дата транзакції (в форматі dd.mm.yyyy hh:mm:ss).
SignVerif	Електронний підпис.
AcquireName	Термінал.
InvoiceNumber	Номер чека.
ParentRRN	РРН транзакції чека, по якому здійснено повернення.

#### Приклад формування ДопРеквизиты:

```

Функция СформироватьТаблицуДопРеквизитовТерминала (Терминал) Экспорт
ДанныеЗагрузки = Неопределено;
Результат = Новый Массив ();

ЗаписьXML = Новый ЗаписьXML;
ЗаписьXML.УстановитьСтроку ("UTF-8");
ЗаписьXML.ЗаписатьОбъявлениеXML ();

ЗаписьXML.ЗаписатьНачалоЭлемента ("Table");

        ЗаписьXML.ЗаписатьНачалоЭлемента ("Record");

        ЗаписьXML.ЗаписатьАтрибут ("TerminalID", Строка (? (TerminalID =
неопределено, "", TerminalID)));
        ЗаписьXML.ЗаписатьАтрибут ("ApprovalCode", Строка (? (ApprovalCode =
неопределено, "", ApprovalCode)));
        ЗаписьXML.ЗаписатьАтрибут ("RRN", Строка (? (RRN =
неопределено, "", RRN)));
        ЗаписьXML.ЗаписатьАтрибут ("IssuerName", Строка (? (IssuerName =
неопределено, "", IssuerName)));
        ЗаписьXML.ЗаписатьАтрибут ("PAN" , ? (PAN = неопределено, "", PAN));

        ЗаписьXML.ЗаписатьАтрибут ("TransactionDate" , ? (TransactionDate =

```

```

неопределено, "", TransactionDate));
    ЗаписьXML.ЗаписатьАтрибут("AcquireName" , ?(AcquireName =
неопределено, "", AcquireName));
    ЗаписьXML.ЗаписатьАтрибут("InvoiceNumber" , ?(InvoiceNumber =
неопределено, "", InvoiceNumber));
    ЗаписьXML.ЗаписатьАтрибут("ParentRRN" , ?(ParentRRN =
неопределено, "", ParentRRN));
    ЗаписьXML.ЗаписатьАтрибут("SignVerif" , ?(SignVerif =
неопределено, "", SignVerif));
    ЗаписьXML.ЗаписатьКонецЭлемента();

    ЗаписьXML.ЗаписатьКонецЭлемента();
    Возврат ЗаписьXML.Закреть();

КонецФункции

```

**Увага!** Для передачі параметрів функцій з дробовою частиною (зокрема, нульовою, наприклад, "2,00") допускається використовувати десятковий розділювач "." (крапка) або "," (кома). Якщо параметр не задано, або задано некоректно, його значення приймається рівним одиниці (1).

#### Приклад ОткрытьЧек:

```

Процедура ОткрытьЧекCashalot(Результат, ПризнакЧекаВозврата = Ложь)

    ПризнакФискальногоЧека = Истина;
    АннулироватьПредыдущийОткрытыйЧек = Истина;

    НомерОткрытогоЧека = 0;
    НомерСмены = 0;

    ЧекОткрыт = ДрайверCashalot.ОткрытьЧек(ФНПодключеннойКассы,
                                           ПризнакФискальногоЧека,
                                           ПризнакЧекаВозврата,

АннулироватьПредыдущийОткрытыйЧек,

                                           НомерОткрытогоЧека,
                                           НомерСмены);

    Сообщить(? (ЧекОткрыт,
                "Открыт чек №" + НомерОткрытогоЧека,
                "Не удалось открыть чек из-за ошибки: " +
ПолучитьОшибкуCashalot()));
    Результат = ЧекОткрыт;

КонецПроцедуры

```

#### Приклад НапечататьФискальнуюСтроку:

```

Процедура НапечататьФискальнуюСтрокуCashalot(Результат, ПозицияЧека)

    Для Каждого ПозицияЧека Из ПозицииЧека Цикл
        ФискальнаяСтрокаНапечатана =
ДрайверCashalot.НапечататьФискальнуюСтроку(ФНПодключеннойКассы,
ПозицияЧека.НаименованиеНоменклатуры, ПозицияЧека.КодУКТВЭД,
ПозицияЧека.Артикул, ПозицияЧека.Штрихкод, ПозицияЧека.ЕдиницаИзмерения, "",
ПозицияЧека.Количество, ПозицияЧека.Цена, ПозицияЧека.Сумма,
ПозицияЧека.ПроцентСкидки, ПозицияЧека.СуммаСкидки, ПозицияЧека.СтавкаНДС,
ПозицияЧека.ЦенаВключаетНДС, ПозицияЧека.СуммаНДС,
ПозицияЧека.ПризнакАкцизногоНалога, "",
ПозицияЧека.ДополнительныеРеквизиты);

```

```

Если НЕ ФискальнаяСтрокаНапечатана Тогда
    Сообщить ("Не удалось заполнить номенклатуру "" +
        ПозицияЧека.НаименованиеНоменклатуры +
        "" в чек Cashalot из-за ошибки: " +
        ПолучитьОшибкуCashalot ());
    Прервать;
КонецЕсли;
КонецЦикла;
Результат = ФискальнаяСтрокаНапечатана;

```

КонецПроцедуры

### Приклад НапечататьФискСтроку:

Процедура НапечататьФискСтрокуCashalot (Результат, ПозицияЧека)

```

Для Каждого ПозицияЧека Из ПозицииЧека Цикл
    ДополнительныеРеквизиты =
    СформироватьXMLДополнительныхРеквизитов (ПозицияЧека);
    ФискальнаяСтрокаНапечатана =
    ДрайверCashalot.НапечататьФискСтроку (ФНПодключеннойКассы, ПозицияЧека.НаименованиеНоменклатуры,
    ПозицияЧека.Количество, ПозицияЧека.Цена,
    ПозицияЧека.Сумма, НомерСекции, ПозицияЧека.СтавкаНДС,
    ДополнительныеРеквизиты);

```

```

Если НЕ ФискальнаяСтрокаНапечатана Тогда
    Сообщить ("Не удалось заполнить номенклатуру "" +
        ПозицияЧека.НаименованиеНоменклатуры +
        "" в чек Cashalot из-за ошибки: " +
        ПолучитьОшибкуCashalot ());
    Прервать;
КонецЕсли;
КонецЦикла;
Результат = ФискальнаяСтрокаНапечатана;

```

КонецПроцедуры

### Приклад НапечататьНефискСтроку:

Процедура НапечататьНефискСтрокуCashalot (Результат, КомментарийКЧеку)

```

КомментарийДобавлен =
    ДрайверCashalot.НапечататьНефискСтроку (ФНПодключеннойКассы,
    КомментарийКЧеку);

```

```

Если НЕ КомментарийДобавлен Тогда
    Сообщить ("Не удалось добавить комментарий к чеку из-за ошибки: " +
        ПолучитьОшибкуCashalot ());
КонецЕсли;
Результат = КомментарийДобавлен;

```

КонецПроцедуры

### Приклад УстановитьРасширенныйПорядокОплатыЧека:

Процедура УстановитьРасширенныйПорядокОплатыЧекаCashalot (Результат, СтруктураРасширенныхПараметровОплаты)

```

РасширенныйПорядокОплатыУстановлен =
    ДрайверCashalot.УстановитьРасширенныйПорядокОплатыЧека (ФНПодключеннойКассы,

```

```

СтруктураРасширенныхПараметровОплаты.ТипОплаты,
СтруктураРасширенныхПараметровОплаты.СуммаПредоплаты,
СтруктураРасширенныхПараметровОплаты.НомерЗаказа,
СтруктураРасширенныхПараметровОплаты.ФискальныйНомерЧекаПредоплатыИлиВозврат
,
СтруктураРасширенныхПараметровОплаты.ФискальныйНомерКассыВозврата,
ДопРеквизиты) ;

```

```

Если НЕ РасширенныйПорядокОплатыУстановлен Тогда
    Сообщить ("Не удалось установить расширенный порядок оплаты из-за
        ошибки: " + ПолучитьОшибкуCashalot());
КонецЕсли;
Результат = РасширенныйПорядокОплатыУстановлен;

```

КонецПроцедуры

### Приклад ЗакрытьЧек:

Процедура ЗакрытьЧекCashalot(Результат, СтруктураОплат)

```

ФискальныйНомерЧека = "";

```

```

ЧекЗакрыт = ДрайверCashalot.ЗакрытьЧек(ФНПодключеннойКассы,
СтруктураОплат.СуммаОплатыНаличными, СтруктураОплат.СуммаОплатыКартой,
СтруктураОплат.СуммаОплатыКредитом, СтруктураОплат.СуммаОплатыСертификатом,
СтруктураОплат.СуммаОплате, ФискальныйНомерЧекаВозврата,
НомерОткрытогоЧека,
ФискальныйНомерЧека);

```

```

Сообщить (? (ЧекЗакрыт,
    "Чек №" + ФискальныйНомерЧека + " успешно фискализирован.",
    "Не удалось фискализировать чек из-за ошибки: " +
ПолучитьОшибкуCashalot()));
Результат = ЧекЗакрыт;

```

КонецПроцедуры

## 10. Відміна відкритого чека

Для відміни раніше відкритого чека використовується функція:

**БУЛЕВО ОтменитьЧек**(ФНПодключеннойКассы)

Параметри функції відміни чека:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.

Приклад:

Процедура ОтменитьЧекCashalot(Результат)

```

ЧекОтменен = ДрайверCashalot.ОтменитьЧек(ФНПодключеннойКассы);

```

```

Сообщить (? (ЧекОтменен,
    "Открытый чек в Cashalot был отменен.",
    "Не удалось отменить чек из-за ошибки: " +
ПолучитьОшибкуCashalot()));
Результат = ЧекОтменен;

```

КонецПроцедуры

## 11. Вивести візуальне відображення чека на екран

Для виводу візуального відображення чека на екран, використовується функція:

**БУЛЕВО** **ОтобразитьЧек** (ФНПодключеннойКассы, ФНЧекаДляОтображения)

Функція **ОтобразитьЧек** використовується для виклику вікна з візуальним відображенням зареєстрованого чека. Вікно з відображенням чека також дозволяє використати додаткові функції друку або відправки чека на електронну пошту, кнопки яких присутні у вікні чека.

Параметри:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.
ФНЧекаДляОтображения	строка	Фіскальний номер чека.

Приклад:

```
Процедура ОтобразитьЧек ()  
  
    Результат = ДрайверCashalot.ОтобразитьЧек (ФНПодключеннойКассы,  
ФНЧекаДляОтображения) ;  
  
    Сообщить (? (Результат <> Неопределено,  
                "Чек отобразили.",  
                "Не удалось отобразить из-за ошибки: " +  
                ПолучитьОшибкуCashalot ())) ;  
  
КонецПроцедуры
```

## 12. Отримання фіскального номера чека за його локальним номером

Для отримання фіскального номера чека за його локальним номером, використовується функція:

**БУЛЕВО** **ПолучитьФискальныйНомерЧека** (ФНПодключеннойКассы,  
ЛокальныйНомерЧека)

Функція **ПолучитьФискальныйНомерЧека** дозволяє визначити локальний номер чека за відомим фіскальним номером чека. Пошук чека відбувається в локальній базі чеків, якщо в локальній базі чек не знайдено, виконується пошук чека на сервері ФСКО.

У відповіді надходить локальний номер чека. Якщо номер чека не знайдено, то повертається пустий рядок.

Параметри:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.
ЛокальныйНомерЧека	строка	Локальний номер чека.

Приклад:

```
Процедура ПолучитьФискальныйНомерЧека ()  
  
    ПолученныйФН =  
ДрайверCashalot.ПолучитьФискальныйНомерЧека (ФискальныйНомерКассы,  
ЛокальныйНомерЧека) ;  
    Если (ПолученныйФН = "") Тогда  
        Сообщить ("Чека с номером " + ЛокальныйНомерЧека + " нету в базе") ;
```

Иначе

Сообщить ("ФН чека: " + ПолученныйФН) ;

КонецЕсли;

КонецПроцедуры

### 13. Отримання локального номера чека за його фіскальним номером

Для отримання фіскального номера чека за його локальним номером, використовується функція:

**БУЛЕВО** **ПолучитьЛокальныйНомерЧека** (ФНПодключеннойКассы, ФНчека)

Функція **ПолучитьЛокальныйНомерЧека** дозволяє визначити локальний номер чека за відомим фіскальним номером чека. Пошук чека відбувається в локальній базі чеків, якщо в локальній базі чек не знайдено, виконується пошук чека на сервері ФСКО.

У відповіді надходить локальний номер чека. Якщо номер чека не знайдено, то повертається пустий рядок. У випадку коли задано параметр **NOAUTOOPENSHIFT** = True, виклик функції при закритті змін переривається і відображається повідомлення "Операція неможлива, необхідно відкрити зміну"

Параметри:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.
ФНчека	строка	Локальний номер чека.

Приклад:

Процедура **ПолучитьЛокальныйНомерЧека** ()

ПолученныйЛокальныйНомер =

ДрайверCashalot.ПолучитьЛокальныйНомерЧека (ФискальныйНомерКассы, ФНчека) ;

Если (ПолученныйЛокальныйНомер = "") Тогда

Сообщить ("Чека с номером " + ФНчека + " нету в базе");

Иначе

Сообщить ("Локальный номер чека: " + ПолученныйЛокальныйНомер) ;

КонецЕсли;

КонецПроцедуры

### 14. Отримання xml структури фіскального чека

Для отримання фіскального номера чека за його локальним номером, використовується функція:

**БУЛЕВО** **ПолучитьXMLЧека**(ФНПодключеннойКассы, ФНчекаXML)

Функція дозволяє отримати XML-структуру зареєстрованого чека за його відомим фіскальним номером.

Параметри:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.
ФНчекаXML	строка	Локальний номер чека.

### Приклад:

```
Процедура ПолучитьXMLчека ()  
  
    XMLчека = ДрайверCashalot.ПолучитьXMLЧека (ФискальнНомерКассы,  
ФНчекаXML);  
    Если (XMLчека = "") Тогда  
        Сообщить ("Чека с номером " + ФНчекаXML + " нету в базе");  
    Иначе  
        Сообщить ("XML чека: " + XMLчека);  
    КонецЕсли;  
  
КонецПроцедуры
```

## 15. Друк періодичних звітів

Для друку періодичних звітів, використовується функція:

**БУЛЕВО НапечататьПериодическийОтчетПоДатам** (ФНПодключеннойКассы,  
ДатаНачалаОтчета, ДатаОкончанияОтчета, ПризнакСокращенногоОтчета)

Період це проміжок часу, що встановлюється між ДатаНачалаОтчета та ДатаОкончанияОтчета.

Параметри:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.
ДатаНачалаОтчета	дата	Початок періоду в форматі dd.MM.yyyy
ДатаОкончанияОтчета	дата	Кінець періоду в форматі dd.MM.yyyy
ПризнакСокращенногоОтчета	булево	Ознака скороченого звіту

### Приклад:

```
Процедура НапечататьПериодическийОтчетCashalot (Результат, ДатаНачалаОтчета =  
Неопределено, ДатаОкончанияОтчета = Неопределено, ПризнакСокращенногоОтчета  
= Ложь)  
  
    СтруктураОтвета =  
ДрайверCashalot.НапечататьПериодическийОтчетПоДатам (ФНПодключеннойКассы,  
? (ДатаНачалаОтчета = Неопределено, ТекущаяДата (), ДатаНачалаОтчета),  
? (ДатаОкончанияОтчета = Неопределено, ТекущаяДата (), ДатаОкончанияОтчета),  
ПризнакСокращенногоОтчета);  
  
    Сообщить (? (СтруктураОтвета <> Неопределено,  
        "Периодический отчет напечатан.",  
        "Не удалось напечатать периодический отчет из-за ошибки: " +  
        ПолучитьОшибкаCashalot ());  
  
    Результат = СтруктураОтвета <> Неопределено;  
  
КонецПроцедуры
```

## 16. Отримання інформації про поточний стан ПРРО

Для отримання інформації про поточний стан використовується функція:

**БУЛЕВО ПолучитьТекущееСостояние** (ФНПодключеннойКассы, ВходныеПараметры,  
XMLСтрокаТекущихПараметров)



Параметри функції поточного стану ПРРО:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.
ВходныеПараметры	строка	Таблиця вхідних параметрів, параметр ігнорується.
XMLСтрокаТекущихПараметров	строка	XML-структура параметрів з поточними значеннями

Таблиця вихідних параметрів має вигляд XML-структури з переліком параметрів.

Параметри XML-структури вихідних параметрів:

Ім'я параметра	Значення
CheckNumber	Локальний номер останнього чека
CashBalance	Залишок готівки в касі
IsOfflineMode	Стан офлайн-режиму каси: 0 – каса в онлайн-режимі; 1 – каса в офлайн-режимі
NextLocalNumber	Наступний локальний номер чека
LastCheckFiscalNumber	Фіскальний номер останнього чека
ShiftNumber	Номер поточної відкритої зміни/ останньої закритої зміни
ShiftID	Ідентифікатор зміни
ShiftFiscalNumber	Фіскальний номер зміни
ShiftState	Стан зміни: 1 – закрыта. 2 – відкрита.
DateTime	Дата і час формування фіскального документа

Приклад:

```
Процедура ПолучитьТекущееСостояниеCashalot (Результат)
```

```
XMLСтрокаТекущихПараметров = "";  
ПараметрыПолучены =  
ДрайверCashalot.ПолучитьТекущееСостояние (ФНПодключеннойКассы, "",  
XMLСтрокаТекущихПараметров);
```

```
Сообщить ("Состояние кассы в XML: " + XMLСтрокаТекущихПараметров);
```

```
КонецПроцедуры
```

## 17. Друк довільної текстової інформації

Для друку на чекову стрічку довільного тексту використовується функція

**БУЛЕВО НапечататьТекстовыйДокумент**(ФНПодключеннойКассы, ТекстДляПечати)

Після виклику відображається вікно з зображенням тексту і можливістю роздрукувати.

У випадку, коли задано автоматичну відправку чеків на друк і встановлено принтер для друку за замовчуванням, друк заданого тексту відбувається без попереднього перегляду.

Функцію можливо викликати без відкриття касової зміни.

Параметри:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка [out]	Фіскальний номер ПРРО. Ідентифікатор пристрою.
ТекстДляПечати	строка	Довільний текст, який передається на друк.

**Приклад:**

Процедура НапечататьТекстовыйДокументCashalot (ТекстДляПечати)

```
ПереносСтроки = "\n";
ТекстДляПечати = "Перший рядок" + ПереносСтроки + "Другий рядок";

ТекстовыйДокументНапечатан =
ДрайверCashalot.НапечататьТекстовыйДокумент (ФНПодключеннойКассы,
ТекстДляПечати);

Сообщить (? (ТекстовыйДокументНапечатан,
"Текстовый документ напечатан.",
"Не удалось напечатать текстовый документ из-за ошибки: " +
ПолучитьОшибкуCashalot ())) ;
```

КонецПроцедуры

## 18. Службове внесення/видача

Для виконання операції службового внесення або видачі використовується функція

**БУЛЕВО** НапечататьЧекВнесенияВыемки (ФНПодключеннойКассы, СуммаВнесения  
або СуммаВыдачи)

Параметри функції службового внесення/видачі:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.
СуммаВнесения або СуммаВыдачи	число	Сума внесення. Сума видачі вказується з мінусом.

**Приклад внесення:**

Процедура СлужбноеВнесениеCashalot (СуммаВнесения)

```
Если СуммаВнесения < 0.01 Тогда
    Сообщить ("Сумма служебного внесения не может быть меньше 0.01 грн.")
    Возврат;
КонецЕсли;

Результат =
ДрайверCashalot.НапечататьЧекВнесенияВыемки (ФНПодключеннойКассы,
СуммаВнесения);
Сообщить (? (Результат,
СуммаВнесения + " грн. успешно внесены в кассу.",
"Не удалось внести " + СуммаВнесения + " грн. в кассу из-за
ошибки: " + ПолучитьОшибкуCashalot ())) ;
```

КонецПроцедуры

### Приклад видачі:

Процедура СлужбнаяВыдачаCashalot (СуммаВыдачи)

```
Если СуммаВнесения < 0 Тогда
    Сообщить ("Сумма служебной выемки должна быть больше или равна 0
грн.")
    Возврат;
КонецЕсли;

Результат =
ДрайверCashalot.НапечататьЧекВнесенияВыемки (ФНПодключеннойКассы,
-СуммаВыдачи);
Сообщить (? (Результат,
    СуммаВыдачи + " грн. успешно изъяты с кассы.",
    "Не удалось изъять " + СуммаВыдачи + " грн. с кассы из-за
ошибки: " + ПолучитьОшибкуCashalot ())) );

КонецПроцедуры
```

## 19. Формування Х-звіту

Для виконання операції формування звіту без гасіння (Х-звіту) використовується функція

**БУЛЕВО НапечататьОтчетБезГашения**(ФНПодключеннойКассы)

Параметри функції для формування Х-звіту:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.

### Приклад:

Процедура СформироватьХОтчетCashalot ()

```
ХОтчетСформирован =
ДрайверCashalot.НапечататьОтчетБезГашения (ФНПодключеннойКассы);
Сообщить (? (ХОтчетСформирован,
    "Х-отчет успешно сформирован.",
    "Не удалось сформировать Х-отчет из-за ошибки: " +
ПолучитьОшибкуCashalot ())) );
```

КонецПроцедуры

## 20. Формування Z-звіту

Для формування Z-звіту використовується функція:

**БУЛЕВО НапечататьОтчетСГашением**(ФНПодключеннойКассы)

**Увага!** Якщо встановлений параметр `NOINTERFACEMODE = True`, то службову видачу, потрібно передбачити перед виконанням функції **НапечататьОтчетСГашением**.

Параметри функції для формування Z-звіту:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.

**Приклад:**

```
Процедура СформироватьZОтчетCashalot ()
    ZОтчетСформирован =
    ДрайверCashalot.НапечататьОтчетСГашением (ФНПодключеннойКассы);
    Сообщить (? (ZОтчетСформирован,
        "Z-отчет успешно сформирован.",
        "Не удалось сформировать Z-отчет из-за ошибки: " +
        ПолучитьОшибкуCashalot ())) ;

КонецПроцедуры
```

## 21. Закриття зміни

Для виконання операції закриття зміни використовується функція:

**БУЛЕВО** **ЗакретьСмену**(ФНПодключеннойКассы)

**Увага!** Якщо встановлений параметр **NOINTERFACEMODE** = **True**, то службову видачу, потрібно передбачити перед виконанням функції **ЗакретьСмену**.

Параметри функції закриття зміни:

Ім'я	Тип	Опис
ФНПодключеннойКассы	строка	Фіскальний номер ПРРО. Ідентифікатор пристрою.

**Приклад:**

```
Процедура ЗакретьСменуCashalot ()

    СменаЗакрыта = ДрайверCashalot.ЗакретьСмену (ФНПодключеннойКассы);

    Сообщить (? (СменаЗакрыта,
        "Смена закрыта.",
        "Не удалось закрыть смену из-за ошибки: " +
        ПолучитьОшибкуCashalot ())) ;

КонецПроцедуры
```