**1.1.4**

Реализованы следующие запросы DirectIO   
  
**Запрос РНМ.**

DirectIO(command,args,dataObject);   
где:   
command = 35   
int[] args={1} не используется .   
String[] dataObject = new String[1]; dataObject[0] присвоит результат   
 **Номер последнего ФД.**  
DirectIO(command,args,dataObject);   
где:   
command = 36   
int[] args={1} не используется .   
String[] dataObject = new String[1]; dataObject[0] присвоит результат   
  
  
**Дата и время последнего ФД в виде (DDMMYYYYhhmm).**  
DirectIO(command,args,dataObject);   
где:   
command = 37   
int[] args={1} не используется .   
String[] dataObject = new String[1]; dataObject[0] присвоит результат   
  
  
**Получение диапазона квитанций (от заданного номера до последнего включительно).**  
DirectIO(command,args,dataObject);   
где:   
command = 38   
int[] args={1}   
args[0] содрежит номер от которого начнется выборка документов   
Object[] dataObject = new Object[1]; - Массив Object в котором под индексом 0 будет коллекция типа Vector<PiritFNTicket>   
Где в классe PiritFNTicket метод getdocNumber() вернет HEX массив с квитанцией ОФД. getDocNumber() вернет int номер документа.   
isSigned() вернет boolean статус - подписан ли документ в ОФД.   
  
**Запрос номера смены.**  
DirectIO(command,args,dataObject);   
где:   
command = 39   
int[] args={1} не используется .   
String[] dataObject = new String[1]; dataObject[0] присвоит результат   
 **Кол-во чеков за смену.**  
DirectIO(command,args,dataObject);   
где:   
command = 40   
int[] args={1}   
args[0] - присвоит результат   
String[] dataObject = new String[1]; - не используется   
  
**Номер документа последней перерегистрации.**  
DirectIO(command,args,dataObject);   
где:   
command = 41   
int[] args={1}   
args[0] - присвоит результат   
String[] dataObject = new String[1]; - не используется

**Аннулирование документа при окончании бумаги.**

**1.1.5**

**Запрос данных из регистров ККТ.**

используемый метод - DirectIO(command,args,dataObject);   
где:   
command = 27   
int[] args={1}   
args[0] - содержит номер запрашиваемого регистра .   
String[] dataObject = new String[1]; dataObject[0] присвоит результат   
  
Соответствия запросов и номеров регистров   
  
121 - продажа за смену  
123 - возвраты за смену  
185 - скидки при продаже  
187 - скидки при возврате  
193 - продажа наличными  
195  - возврат наличными  
205  - продажа картой  
207 -  возврат картой   
  
общие необнуляемые суммы   
112 - сумма приходов   
113 - возвратов прихода   
114 - покупка (расход)   
115 - возврат покупки (возврат прихода)   
  
необнуляемые суммы по типам оплаты.   
212 - Продажа (приход)   
213 - Возврат (возврат прихода)   
214 - Покупка (расход)   
215 - Возврат покупки (возврат расхода)

**метод DirectIO для получения квитанций ОФД начиная с заданного номер чека и количество чеков.**

используемый метод -  DirectIO(command,args,dataObject);   
где:   
command = 42   
int[] args={2}   
args[0] содержит номер от которого начнется выборка документов.

args[1] содержит количество запрашиваемых чеков.  
Object[] dataObject = new Object[1]; - Массив Object в котором под индексом 0 будет коллекция типа Vector<PiritFNTicket>  где каждый экземпляр PiritFNTicket соответствует квитанции полученной из ФН.

Для извлечения информации из квитанции в классe PiritFNTicket предусмотрены следующие методы :

***getdocNumber()*** - вернет HEX массив с квитанцией ОФД.

***getDocNumber()*** - вернет int номер чека.   
***isSigned()***- вернет boolean статус - подписан ли документ в ОФД.

***getDateOfDoc()*** - вернет String с датой и временем квитанции в формате - ddMMyyyyHHmm.

***getTicketNumber()*** - вернет String с номером квитанции.

***getValueFPD()*** -вернет String значение ФПД.

**1.1.19**

**Для включения/выключения отрезки чека необходим вызов метода directIO перед вызовом метода beginFiscalReceipt**

directIO(command, args, dataObject);

где:

command = 15

args[0] = 1 - не выполнять отрезку чека, 0 - выполнять

dataObject - не используется.

**Для включения/выключения печати чека необходим вызов метода directIO перед вызовом метода beginFiscalReceipt**

directIO(command, args, dataObject);

где:

command = 14

args[0] = 1 - не печатать чек на ленте, 0 - печатать

dataObject - не используется.

**1.1.20**

**Драйвер переведен на библиотеку JSSC, теперь при переподключение ФР не теряется драйвер не теряет связь.**

**Для включения/выключения отрезки чека необходим вызов метода directIO перед вызовом метода beginFiscalReceipt**

directIO(command, args, dataObject);

где:

command = 15

args[0] = 1 - не выполнять отрезку чека, 0 - выполнять

dataObject - не используется.

**x и z отчет отрезаются всегда.**

**Реализован интерфейс EventCallBacks**

При открытии крышки ФР генерируется событие FPTR\_SUE\_COVER\_OPEN

При закрытии крышки ФР FPTR\_SUE\_COVER\_OK

При окончании бумаги ФР генерируется событие FPTR\_SUE\_REC\_EMPTY

При поступлении бумаги ФР генерируется событие FPTR\_SUE\_REC\_PAPEROK

При потери связи с ФР генерируется событие JPOS\_SUE\_POWER\_OFFLINE

При установки связи с ФР генерируется событие JPOS\_PS\_ONLINE

**Пример мониторинга статуса ФР через EventCallBacks:**

PiritSvc112 piritSvc112 = new PiritSvc112();

EventCallbacks callBacks = piritSvc112.createPiritCallBacks();

StatusUpdateListener listener = new StatusUpdateListener(){

@Override

public void statusUpdateOccurred(StatusUpdateEvent statusUpdateEvent) {

if(statusUpdateEvent.getStatus()== JposConst.JPOS\_PS\_ONLINE) {

System.out.println("FR online");

}

}

piritSvc112.addStatusUpdateListener(listener);

piritSvc112.open("PIRIT-2F",callBacks);

**1.1.21**

**При вызове beginFiscalReceipt printXReport printZReport и при смене статуса EventCallBacks JPOS\_SUE\_POWER\_OFFLINE на JPOS\_PS\_ONLINE вызывается команда - “Начало работы”**

**1.1.24**

**Реализована печать копии чека при окончании ленты, и получении статуса “Документ закрыт в ФН”.**

**Реализовано влючение/выключение печати блока расчета драйвером вместо ФР.**

**Для включения/выключения печати блока расчета драйвером используется directIO-16**

directIO(command, args, dataObject);

где:

command = 16

args[0] = 1 - печатать блок, 0 - не печатать

dataObject - не используется.

**1.1.25**

**Добавлены следующие DIrectIO**

**55**

arg0**: 55**

Печать копии z-отчета.

Печать копии z-отчета. Параметр int[] arg1 и параметр Object arg2[] не используются.

**56**

arg0**: 56**

Запрос информации с ФР.

В метод передается параметр String arg2[], параметр int[] arg1 не используется.

Результат запроса будет записан как массив в String arg2[] в следующем виде:

· РН ККТ

· ЗН ККТ

· Модель ФР

· Наименование ОФД

· Адрес сервера ОФД

· № порта

· Таймаут ожидание ответа от ОФД

**57**

arg0**: 57**

Запрос данных последнего записанного ФД в ФН.

В метод передается параметр String arg2[], параметр int[] arg1 не используется.

Результат запроса будет записан как массив в String arg2[] в следующем виде:

· Тип документа

· Флаг получения квитанции ОФД

· Дата и время

· ФД №

· ФПД №

· Тип операции

· Сумма

**58**

arg0**: 58**

Запрос информации с отчета об открытии смены.

В метод передается параметр String arg2[], параметр int[] arg1 не используется.

Результат запроса будет записан как массив в String arg2[] в следующем виде:

· Номер смены

· Тип документа (отчет об открытии)

· Дата и время

· РН ККТ

· ЗН ККТ

· ФН №

· ФД №

· ФПД №

· Наименование ОФД

**59**

arg0**: 59**

Запрос информации с отчета об закрытии смены.

В метод передается параметр String arg2[], параметр int[] arg1 не используется.

Результат запроса будет записан как массив в String arg2[] в следующем виде:

· Номер смены

· Тип документа (отчет об открытии)

· Дата и время

· РН ККТ

· ЗН ККТ

· ФН №

· ФД №

· ФПД №

· Наименование ОФД

**1.1.28**

**Добавлены следующие DIrectIO**

**17**

arg0**: 17**

Установить атрибут текста при печати свободной строкой.

В метод передается параметр int[] arg1. Параметр Object arg2 не используется.

Атрибут устанавливается по значению arg1[0].

**18**

arg0**: 18**

Команда отрезки бумаги.

Параметры не передаются.

**19**

arg0**: 19**

Команда печати разделительной линии.

В метод передается параметр int[] arg1. Параметр Object arg2 не используется.

Позиция начала линии задается arg1[0].

Позиция конца линии задается arg1[1].

Толщина линии задается arg1[1].

**20**

arg0**: 20**

Команда печати QR кода с текстом.

В метод передается параметр int[] arg1. и Object arg2.

Атрибут текста задается arg1[0].

Ширина QR-кода задается arg1[1].

Начало текста задается arg1[2].

Начало QR-кода задается arg1[3].

Строка1 задается arg2[0].

Строка2 задается arg2[1].

Содержимое QR-кода arg2[2].

Драйвер теперь отрабатывает некорректный ивент “НЕТ БУМАГИ” при отключении питания ФР.

Исправлен расчет поля сдача в блоке расчета.

При вызове getData c параметром FiscalPrinterConst.FPTR\_GD\_Z\_REPORT - метод теперь возвращает номер последней закрытой смены, а не текущей. В соответствии UPOS.

В DirectIO 52 добавлена установка кода магазина

Для установки кода магазина используется directIO - 52 arg1[0] = 15000, arg2 = Строке с кодом магазина

Добавлено разделение блока налоговых ставок от блока расчета

Сервисный документ теперь не печатает блок ФНС  
  
**1.1.29**

**Изменена работа PrintRecItemAdjustment, функция сохраняет параметры скидки и применяет их к следующему вызову PrintRecItem**

**1.1.30**

**Добавлен возврат данных с счетчиков по типам продаж в DIrectIO 27**

Номера счетчиков по типам продаж передаваемые в args[0]:

args[0] = 193; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT0

args[0] = 197; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT1

args[0] = 198; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT2

args[0] = 202; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT3

args[0] = 206; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT4

args[0] = 232; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT5

args[0] = 233; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT6

args[0] = 234; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT7

args[0] = 235; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT8

args[0] = 236; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT9

args[0] = 237; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT10

args[0] = 238; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT11

args[0] = 239; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT12

args[0] = 240; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT13

args[0] = 201; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT14

args[0] = 205; // REGISTER\_DAILY\_TOTAL\_SALE\_PAYMENT15

args[0] = 195; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT0

args[0] = 199; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT1

args[0] = 203; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT2

args[0] = 220; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT3

args[0] = 221; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT4

args[0] = 222; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT5

args[0] = 223; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT6

args[0] = 224; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT7

args[0] = 225; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT8

args[0] = 226; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT9

args[0] = 227; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT10

args[0] = 228; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT11

args[0] = 229; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT12

args[0] = 230; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT13

args[0] = 231; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT14

args[0] = 207; // REGISTER\_DAILY\_TOTAL\_REFUND\_PAYMENT15

**1.1.31**

**Драйвер теперь implements JPOS 1.13.1 - FiscalPrinterService113.**

**21**

arg0**: 21**

Команда установки атрибута признака расчета.

Атрибут задается arg1[0].

**22**

arg0**: 22**

Команда установки атрибута предмета расчета.

Атрибут задается arg1[0].

**1.1.32**

**Добавлен флаг включения ожидания статуса связи с принтером, при получении статуса об окончании бумаги, т.к при выключении питания на ФР, в первую очередь питание просаживается на датчиках наличия бумаги, и иногда ФР перед выключением успевает отправить ответ на опрос статуса - нет бумаги.**

**61**

arg0**: 61**

Команда установки флага ожидания.

Атрибут задается arg1[0].

Пример

int arg1[0] = 0; - не ждать статус связи с ФР, сначала будет передан статус об окончании бумаги, потом об отсутствии связи.

piritSvc112.directIO(61,arg1,null);

**1.1.33**

**Добавлен флаг управления печатью блока ФНС на сервисном документе**

**62**

arg0**: 62**

Атрибут задается arg1[0].

int arg1[0] = 1; - печатать блок

**1.1.36**

**Добавлена команда передачи КТН в товарную позицию 73**

arg0**: 73**

Атрибут задается arg2[0] в виде массива строковых параметров.

* (Строка[1..32]) Код товарной номенклатуры
* (Строка[1..64]) Дополнительный реквизит предмета расчёта
* (Строка[1..16]) Единица измерения предмета расчёта
* (Целое число) Признак агента по предмету расчёта
* (Строка)[0..12] ИНН поставщика
* (Строка)[0..19] Телефон поставщика
* (Строка)[0..256] Наименование поставщика
* (Строка)[0..256] Адрес оператора перевода (для банк.пл.агента/банк.пл.субагента, иначе пустой)
* (Строка)[0..12] ИНН оператора перевода (для банк.пл.агента/банк.пл.субагента, иначе пустой)
* (Строка)[0..64] Наименование оператора перевода (для банк.пл.агента/банк.пл.субагента, иначе пустой)
* (Строка)[0..19] Телефон оператора перевода (для банк.пл.агента/банк.пл.субагента, иначе пустой)
* (Строка)[0..24] Операция платежного агента (для банк.пл.агента/банк.пл.субагента, иначе пустой)
* (Строка)[0..19] Телефон платежного агента (для пл.агента/пл.субагента, иначе пустой)
* (Строка)[0..19] Телефон оператора по приему платежей (для пл.агента/пл.субагента, иначе пустой

int arg1[0] не используется

piritSvc112.directIO(73,null,arg2);

**1.1.37**

**Указан полный путь к классу PiritSvc112.**

**1.1.38**

**Исправление для команды 0x24 - если включен пакетный режим, то ответ игнорируется.**

**1.1.39**

**Исправление ошибки таймаута после пакетной печати больших документов.**

**1.13.42**

**Исправление ошибки версии драйвера**

**Передача тега 2108 DIrectIO 79**

**Метод запроса статуса проверки КМ DIrectIO 80**

**Метод полной очистки буфера ФН DIrectIO 81**

**Метод передачи 1244, 1245, 1246, 1254 DIrectIO 82**

**Режим проверки КМ в DIrectIO 74**

**Метод отказа от реализации КМ DIrectIO 75**

**Метод установки таймаута проверки КМ DIrectIO 83**

**Добавлен реквизит 1263 в DIrectIO 73**

**1.13.43**

**Метод для принятия КМ без добавления в чек DIrectIO 84  
Метод для добавления КМ в чек без принятия DIrectIO 85**

**1.14.0**

**Добавлен параметр часовой зоны (тег 1011) в DirectIO 24**

**Добавлены методы DirectIO 86, 87**

**1.16.0**

**Исправление DirectIO 24**

**1.16.1**

**Исправление DeviceServiceVersion**