



**ООО “АСТРОН ЛТД”**

**Пакет прикладных программ  
PowerPOS Версия 5.0**

**Создание драйверов  
подсистемы сообщений**

Минск – 2015

ООО "Астрон ЛТД"  
220113, г. Минск, ул. Мележа 5, корп. 2 оф.  
1201  
тел. +375 (17) 392-56-00, 01, 02, 03, 04, 05  
факс +375 (17) 392-56-  
06 <http://www.astron.by>

© Copyright Astron Ltd., 2014. All right reserved.

Данная публикация или ее часть не могут быть воспроизведены в любой форме без предварительного письменного разрешения фирмы Астрон ЛТД.

## **Содержание:**

Описание драйвера подсистемы сообщений .....	4
Принципы работы подсистем сообщений .....	4
Интерфейс ICashDeskMessageClient .....	4
Свойства .....	4
Методы .....	5
Интерфейс ICashDeskMessage .....	5
Свойства .....	5
Методы .....	5
События .....	5
Класс EventRecivedEventArgs .....	6
Свойства .....	6
Пример реализации .....	6

## Описание драйвера подсистемы сообщений.

Драйвер банковского терминала должен представлять собой сборку для Microsoft .Net 2.0.

Класс драйвера должен реализовать интерфейс

SoftMarket.MP2003.CashDesk.Globals.MessageService.ICashDeskMessageClient, находящийся в сборке CashDeskGlobals.dll.

Сборку, реализующую драйвер, необходимо положить в каталог приложения ( по умолчанию C:\Program Files\Astron\PowerPOS\v5).

Нежелательно что бы драйвер генерировал какие либо исключения.

В проект драйвера нужно включить сборки MP2005Globals.dll и Globals.dll.

## Принципы работы подсистем сообщений.

При запуске кассы центр сообщений ищет все драйвера клиентов и, если флаг IsActive клиента установлен в true, запускает его, вызывая функцию `Start (ICashDeskMessage[] messageSubsystems)`. В функции `Start(ICashDeskMessage[] messageSubsystems)` клиенту передается список подсистем вывода сообщений, куда, собственно говоря он и может передавать сообщения. Так же у каждой подсистемы имеется событие, которое генерируется каждый раз, когда на кассе производится какое либо действие. Для генерации событий клиент может поступить двумя способами. Подписаться на событие подсистемы и соответственно реагировать на действия кассы. Либо же он может запустить свой собственный поток и реагировать на какие либо внешние события (например: получать сообщения от удаленной системы администрирования). В любом случае для передачи сообщения в подсистему используется функция `SendMessage (string message)`.

## Интерфейс ICashDeskMessageClient

Сборка: CashDeskGlobals.dll.

Полное имя:

SoftMarket.MP2003.CashDesk.Globals.MessageService.ICashDeskMessageClient.

Описание: Интерфейс клиента подсистемы сообщений кассы.

```
public interface ICashDeskMessageClient
{
    void Start(ICashDeskMessage[] messageSubsystems);

    void Stop();

    bool IsActive{get;}

    string ClientName{get;}
}
```

### Свойства

Свойство	Описание
<code>string</code> IsActive	Признак того, что клиент активен, и будет запущен центром сообщений кассы. Зависит от внутренних настроек клиента.
<code>string</code> ClientName	Уникальный идентификатор клиента. Обычно GUID.

## Методы

Метод	Описание
<code>void Start (ICashDeskMessage [] messageSubsystems)</code>	Запускает клиента. В параметрах передается список подсистем вывода сообщений кассы.
<code>void Stop ()</code>	Останавливает клиента.

## Интерфейс ICashDeskMessage

Сборка: CashDeskGlobals.dll.

Полное имя: `SoftMarket.MP2003.CashDesk.Globals.MessageService.ICashDeskMessage`.

Описание: Итерфейс подсистемы вывода сообщений кассы.

```
public interface ICashDeskMessage
{
    void SendMessage (string message);

    string SubsystemName {get;}

    event EventHandler<EventArgs> EventReceived;
}
```

## Свойства

Свойство	Описание
<code>String SubsystemName</code>	Уникальный идентификатор подсистемы.  <b>BottomLine</b> - подсистема вывода сообщений в нижнюю строку кассы. <b>PaymentWindow</b> - подсистема вывода сообщений в окно оплаты. <b>ReceiptPrint</b> - подсистема вывода сообщений для печати на чеке.

## Методы

Метод	Описание
<code>void SendMessage (string message)</code>	Отсылает сообщение в подсистему.

## События

Событие	Описание
<code>EventHandler&lt;EventArgs&gt; EventReceived</code>	Генерируется подсистемой при выполнении кассой какого либо действия.

## Класс EventRecivedEventArgs

Сборка: CashDeskGlobals.dll.

Полное имя:

SoftMarket.MP2003.CashDesk.Globals.MessageService.EventRecivedEventArgs.

Описание: Класс для событий подсистемы вывода сообщений.

### Свойства

Свойство	Описание
<code>Const.Events</code> EventCode {get;}	Код события. Список событий приведет в документе «Синхронизация с бэк-офисом» глава «События системного лога»
<code>Hashtable</code> Parameters {get;}	Дополнительные данные события.

### Пример реализации

Пример реализации клиента, который реагирует на добавление в чек товара и выводит сообщения об этом во все три подсистемы.

```
using System;
using System.Collections.Generic;
using System.Text;
using SoftMarket.MP2003.CashDesk.Globals.MessageService;
using SoftMarket.MP2003.CashDesk.CashDeskSys.Domain;
using System.Collections;
using SoftMarket.MP2003.CashDesk.Globals;
using SoftMarket.MP2003.Globals.CashDesk;
using System.IO;
using SoftMarket.Globals;

namespace SoftMarket.MP2003.CashDesk.MCCCashDeskStd
{
    public class ArticleFoundMessageCenterClient : ICashDeskMessageClient
    {
        #region Members

        private ICashDeskMessage bottomLineMessageSubsystem;
        private ICashDeskMessage paymentWindowMessageSubsystem;
        private ICashDeskMessage receiptPrintMessageSubsystem;

        #endregion

        #region ICashDeskMessageClient

        /// <summary>
        /// Инициализация и запуск клиента.
        /// </summary>
        /// <param name="messageSubsystems">Список подсистем вывода
сообщений.</param>
        public void Start(ICashDeskMessage[] messageSubsystems)
        {
            foreach(ICashDeskMessage messageSubsystem in messageSubsystems)
            {
                if(messageSubsystem.SubsystemName == "BottomLine")
                {
                    bottomLineMessageSubsystem = messageSubsystem;
                }
            }
        }
    }
}
```

```

        bottomLineMessageSubsystem.EventRecived += new
EventHandler<EventRecivedEventArgs>(BottomLineMessageSubsystem_EventRecived);
    }
    else if(messageSubsystem.SubsystemName == "PaymentWindow")
    {
        paymentWindowMessageSubsystem = messageSubsystem;
        paymentWindowMessageSubsystem.EventRecived += new
EventHandler<EventRecivedEventArgs>(PaymentWindowMessageSubsystem_EventRecive
d);
    }
    else if(messageSubsystem.SubsystemName == "ReceiptPrint")
    {
        receiptPrintMessageSubsystem = messageSubsystem;
        receiptPrintMessageSubsystem.EventRecived += new
EventHandler<EventRecivedEventArgs>(ReceiptPrintMessageSubsystem_EventRecived
);
    }
}

/// <summary>
/// Остановка клиента.
/// </summary>
public void Stop()
{
    if(bottomLineMessageSubsystem != null)
        bottomLineMessageSubsystem.EventRecived -= new
EventHandler<EventRecivedEventArgs>(BottomLineMessageSubsystem_EventRecived);

    if(paymentWindowMessageSubsystem != null)
        paymentWindowMessageSubsystem.EventRecived -= new
EventHandler<EventRecivedEventArgs>(PaymentWindowMessageSubsystem_EventRecive
d);

    if(receiptPrintMessageSubsystem != null)
        receiptPrintMessageSubsystem.EventRecived -= new
EventHandler<EventRecivedEventArgs>(ReceiptPrintMessageSubsystem_EventRecived
);
}

/// <summary>
/// Признак того, что клиент активен и его необходимо запускать.
/// </summary>
public bool IsActive
{
    get
    {
        return true; //Признак активности должен браться из настроек
клиента.
    }
}

/// <summary>
/// Уникальный идентификатор клиента.
/// </summary>
public string ClientName
{
    get
    {
        return "902602F3-8806-48bb-A04E-B48AF62F1CE1";
    }
}
}
#endregion

```

```

#region Private

/// <summary>
/// Получить сообщение для события добавления товара в чек.
/// </summary>
/// <param name="eventCode">Код события.</param>
/// <param name="parameters">Дополнительные параметры.</param>
/// <returns>Сообщение, если не то событие то null.</returns>
private string
GetMessage(SoftMarket.MP2003.CashDesk.Globals.Const.Events eventCode,
Hashtable parameters)
{
    if(eventCode ==
SoftMarket.MP2003.CashDesk.Globals.Const.Events.AddArticle)
    {

        if(parameters.Contains(SoftMarket.MP2003.CashDesk.Globals.Const.EventInfo.Art
icle))
        {
            Article article =
parameters[SoftMarket.MP2003.CashDesk.Globals.Const.EventInfo.Article] as
Article;

            if(article != null)
                return string.Format("Добавлен товар {0}",
article.ArticleName);
        }

        return null;
    }

    /// <summary>
    /// Событие от подсистемы вывода сообщений в нижнюю строку кассы.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void BottomLineMessageSubsystem_EventRecived(object sender,
EventRecivedEventArgs e)
    {
        try
        {
            string message = GetMessage(e.EventCode, e.Parameters);

            if(message != null)
                bottomLineMessageSubsystem.SendMessage(message);
        }
        catch(Exception err)
        {
            Log.Write(err, this);
        }
    }

    /// <summary>
    /// Событие от подсистемы вывода сообщений в окно оплаты.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void PaymentWindowMessageSubsystem_EventRecived(object
sender, EventRecivedEventArgs e)
    {
        try
        {
            string message = GetMessage(e.EventCode, e.Parameters);

```



```

        if (message != null)
            paymentWindowMessageSubsystem.SendMessage(message);
    }
    catch (Exception err)
    {
        Log.Write(err, this);
    }
}

/// <summary>
/// Событие от подсистемы вывода сообщений на чек.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void ReceiptPrintMessageSubsystem_EventRecived(object sender,
EventRecivedEventArgs e)
{
    try
    {
        string message = GetMessage(e.EventCode, e.Parameters);

        if (message != null)
            receiptPrintMessageSubsystem.SendMessage(message);
    }
    catch (Exception err)
    {
        Log.Write(err, this);
    }
}

#endregion
}
}

```