

پروژه PAX.PCPOS

نام مدرک:		عنوان سند Developer Guid
کد مدرک: CDI-PAX.PCPOS- DeveloperGuid		تعداد صفحات: ۱۵

وضعیت تایید			بازنگری : ۰۴
تصویب کننده	تایید کننده	تهیه کننده	نام و نام خانوادگی
محمدجواد صمدی		ناصر آفرین	
مدیرعامل		مدیر پروژه	سمت
			تاریخ و امضاء

تاریخچه اصلاحات مستند			
بازنگری	تاریخ اصلاح	اقدام کننده	شرح تغییرات
۰۴	۹۲/۰۹/۲۵	آفرین	ایجاد
۰۵	۹۳/۰۲/۱۴	آفرین	

حق تألیف:

این سند شامل اطلاعات اختصاصی و محرمانه شرکت توسعه و نوآوری شهر می باشد. بدون اعلام اجازه از سوی شرکت توسعه و نوآوری شهر، حق هیچ گونه نشر، افشاسازی، استفاده از این سند یا محتویات آن وجود ندارد.

City Development and Innovation Co. all rights reserved. This document and the intellectual property contained herein is the property of City Development and Innovation Co. and may not be reproduced or transmitted in any form or by any means, or forwarded to any third party, without prior written consent from City Development and Innovation Co.

فهرست مطالب

فهرست مطالب

۴	هدف	۱-
۴	محدوده	۲-
۴	مراجع	۳-
۴	تعاریف	۴-
۴	مقدمه	۵-
۴	روش کار و فراخوانی دستورات DLL	6-
۴	پیش نیاز ها :	۱-۶-
۵	اضافه کردن به پروژه	۲-۶-
۶	فراخوانی تابع ها	۳-۶-
۶	Init	۱-۳-۶-
۷	خرید (Purchase)	۲-۳-۶-
۷	پرداخت قبض	۳-۳-۶-
۸	دریافت موجودی (BalanceInquiry)	۴-۳-۶-
۹	دریافت آخرین تراکنش (GetLastTransaction)	۵-۳-۶-
۹	دریافت آخرین تراکنش PC POS (GetLastPcPosTransaction)	۶-۳-۶-
۱۰	دریافت اطلاعات ترمینال (GetPosInfo)	۷-۳-۶-
۱۰	چاپ متن (PrintText)	۸-۳-۶-
۱۰	چاپ بارکد (PrintText)	۹-۳-۶-
۱۱	آپدیت Response	۱۰-۳-۶-
۱۲	نمونه کد ها	۱۱-۳-۶-

۱- هدف

ارائه راهنمای مناسب برای برنامه نویسان

۲- محدوده

برنامه نویسان برنامه های استفاده کننده از PAX PC-POS بانک شهر

۳- مراجع

ندارد

۴- تعاریف

۵- مقدمه

با کمک این ابزار که به صورت DLL در اختیار برنامه نویسان قرار میگیرد قابلیت ارسال دستورات خرید و دریافت موجودی و ... از طریق برنامه ها به دستگاه های PAX بانک شهر فراهم میگردد. ارتباط ترمینال و کامپیوتر یا با استفاده از پورت سریال انجام میشود و یا از طریق LAN. این افزونه امکان ارسال دستورات را با هر دو روش امکان پذیر کرده است.

۶- روش کار و فراخوانی دستورات DLL

۶-۱- پیش نیاز ها :

برای استفاده از امکانات PC_POS نیاز به ابزار زیر دارید.

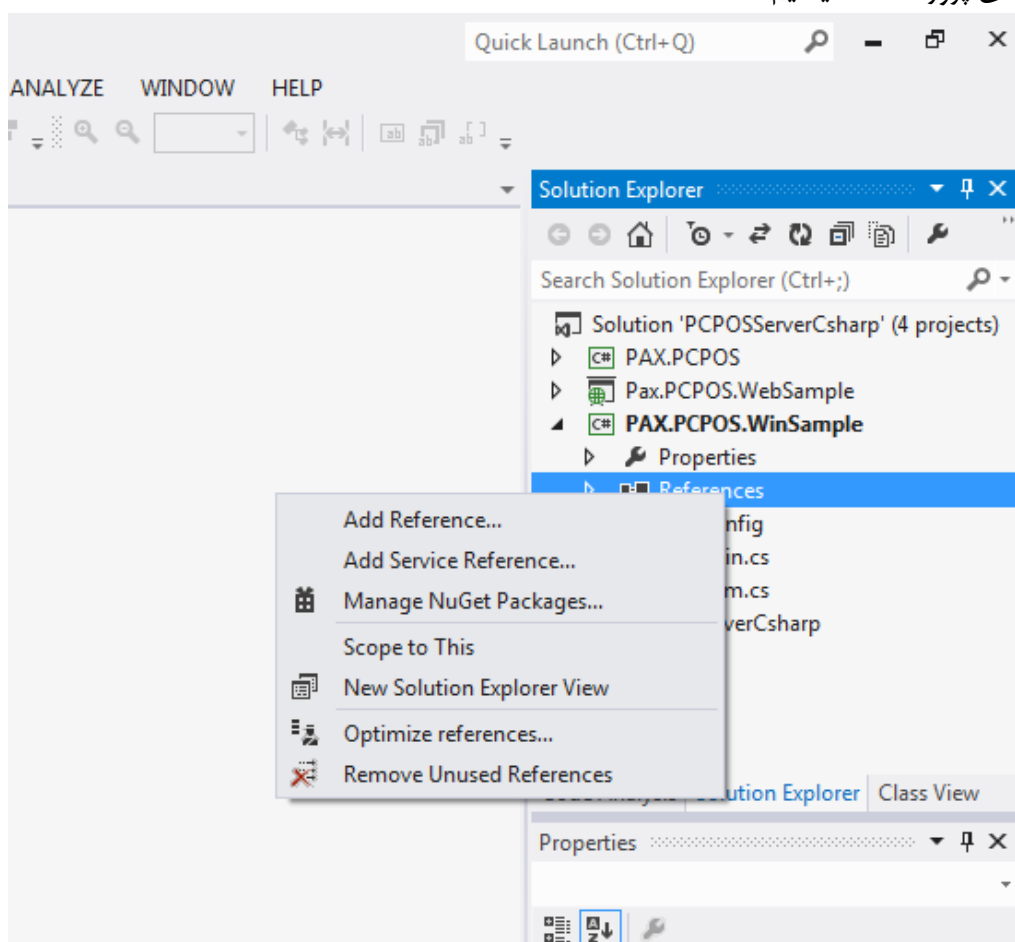
۱- یک عدد پوز S80 و یا S90 بانک شهر بروزرسانی شده

۲- پوشه حاوی فایل های PAX.PCPOS.dll و PCPOSServer.dll

۳- برای پوز های S80 وجود زیر ساخت های لازم (شبکه و تنظیمات لازم) برای انتقال اطلاعات به شبکه مبنا کارت

۶-۲- اضافه کردن به پروژه

۱- قبل از فراخوانی دستورات DLL ابتدا میبایست از گزینه Add Reference در VS 2010,2012 فایل PAX.PCPOS.dll را به رفرنس های پروژه اضافه میکنیم.



۳-۶- فراخوانی تابع ها

نکته : جهت فراخوانی تابع ها نیازی به ایجاد نسخه جدید از آبجکت PCPOS نیست و فراخوانی PCPOS.PosCommander.Instance و سپس فراخوانی تابع ها کفایت میکند.

Init - ۱-۳-۶

اولین تابعی که قبل از هر تابع دیگری باید فراخوانی شود تابع Init میباشد که میتواند در طول برنامه فقط یک بار فراخوانی شود و نیازی نیست به طور دائم قبل از دیگر تابع ها فراخوانی شود. این تابع با سه امضای متفاوت جهت ایجاد ارتباط از طریق SerialPort و یا LAN با ترمینال به کار میرود .

```
bool Init(String Com, int timeOut)
```

این تابع برای ارتباط با ترمینال از طریق پورت سریال میباشد .

۱-۱-۳-۶ ورودی

ردیف	نام	نوع	توضیحات
۱-	COM	String	رشته ای حاوی نام COMPort متصل به دستگاه پوز
۲-	Timeout	int	مدت زمان انتظار برای رسیدن جواب از پوز
۳-	BaudRate	String	این فیلد دارای مقدار عددی تعیین شده توسط شرکت توسعه میباشد. مقدار پیش فرض آن نیز ۳۸۴۰۰ میباشد که در صورت عدم ارسال این پارامتر، این مقدار جایگزین خواهد شد.

```
public Boolean Init(IPAddress host, int timeOut)
```

این تابع جهت ارتباط با ترمینال از طریق LAN میباشد.

ردیف	نام	نوع	توضیحات
۱-	host	String	رشته ای حاوی IP ترمینال میباشد.
۲-	Timeout	int	مدت زمان انتظار برای رسیدن جواب از پوز

```
public Boolean Init()
```

این تابع جهت ارتباط با ترمینال از طریق پارامتر های تعیین شده در فایل Setting کنار DLL (PAX.PCPOS.dll.config) میباشد.

۲-۱-۳-۶ خروجی

خروجی این تابع از نوع Boolean است که مقدار True نشان دهنده درست بودن پارامتر های ارتباط میباشد و خروجی False برای این تابع نشان دهنده عدم وجود پورت کام یا باز بودن آن توسط دیگر برنامه ها، میباشد.

مثال ۳-۱-۳-۶

فراخوانی تابع زیر، پارامترهای ارتباط را بر روی پورت COM1 و زمان Timeout برابر ۶۰ ثانیه قرار میدهد.

```
PCPOS.PosCommander.Instance.Init("COM1", 60);
```

خرید (Purchase) ۳-۲-۶

این تابع، ارسال دستور خرید به پوز را بر عهده دارد.

```
Response Purchase(String additionalData,String amount,Boolean print, String carId, String carOwner, String subject)
```

ورودی ها ۳-۲-۱-۶

ردیف	نام	نوع	توضیحات
۱-	additionalData	String	این اطلاعات بدون تغییر در جواب بازگردانده میشود و نقش مرجع برای درخواست تراکنش به PC POS را دارد.
۲-	amount	String	مبلغ
۳-	print	Boolean	امکان چاپ نسخه مشتری
۴-	carId	String	شماره خودرو
۵-	carOwner	String	مشخصات مالک خودرو
۶-	subject	String	عنوان قبض صادر شده

• توجه: جهت ایجاد خط جدید در متن Text از کاراکتر "^" استفاده نمایید مثال "شهرداری تهران"

واحد درامد"

• توجه: تعداد کاراکترهای بارکد باید حداکثر ۱۶ و زوج باشد.

خروجی ۳-۲-۲-۶

پاسخ نیز به صورت آبجکت Response بر میگردد. در مورد این آبجکت بعدا توضیح داده میشود.

مثال ۳-۲-۳-۶

```
PCPOS.PosCommander.Instance.Purchase("RefNumber","10000",true,"12334","نام مالک","موضوع","نام مالک");
```

پرداخت قبض ۳-۳-۶

این تابع، ارسال دستور پرداخت قبض به پوز را بر عهده دارد.

`Response CallBillPayment(String additionalData, String billId, String paymentId, Boolean print)`

ورودی ها ۱-۳-۳-۶

ردیف	نام	نوع	توضیحات
۱-	additionalData	String	این اطلاعات بدون تغییر در جواب بازگردانده میشود و نقش مرجع برای درخواست تراکنش به PC POS را دارد.
۲-	billId	String	شماره قبض
۳-	paymentId	String	شماره پرداخت
۴-	print	Boolean	امکان چاپ نسخه مشتری

- توجه: جهت ایجاد خط جدید در متن Text از کاراکتر "^" استفاده نمایید مثال "شهرداری تهران"
- واحد درآمد
- توجه: تعداد کاراکتر های بارکد باید حداکثر ۱۶ و زوج باشد.

خروجی ۲-۳-۳-۶

پاسخ نیز به صورت آبجکت Response بر میگردد. در مورد این آبجکت بعدا توضیح داده میشود.

مثال ۳-۳-۳-۶

`PCPOS.PosCommander.Instance.BillPayment("RefNumber","111111111111","22222222",true);`

۴-۳-۶ دریافت موجودی (BalanceInquiry)

این تابع وظیفه ارسال دستور دریافت موجودی به پوز را برعهده دارد .

`Response BalanceInquiry(Boolean Print = true)`

ورودی ۱-۴-۳-۶

ردیف	نام	نوع	توضیحات
۱-	additionalData	String	این اطلاعات بدون تغییر در جواب بازگردانده میشود و نقش مرجع برای درخواست تراکنش به PC POS را دارد.
۲-	print	Boolean	امکان چاپ نسخه مشتری

خروجی -۲-۴-۳-۶

پاسخ نیز به صورت آبجکت Response بر میگردد. در مورد این آبجکت بعدا توضیح داده میشود.

مثال -۳-۴-۳-۶

```
PCPOS.PosCommander.Instance.BalanceInquiry();
```

۵-۳-۶ دریافت آخرین تراکنش (GetLastTransaction)

این تابع ارسال دستور دریافت آخرین تراکنش انجام شده توسط پوز را برعهده دارد.

```
Response CallGetLastTransaction()
```

ورودی -۱-۵-۳-۶

این تابع ورودی ندارد.

خروجی -۲-۵-۳-۶

پاسخ نیز به صورت آبجکت Response بر میگردد. در مورد این آبجکت بعدا توضیح داده میشود.

مثال -۳-۵-۳-۶

```
PCPOS.PosCommander.Instance.GetLastTransaction();
```

۶-۳-۶ دریافت آخرین تراکنش PC POS (GetLastPcPosTransaction)

این تابع ارسال دستور دریافت آخرین تراکنش PC POS به پوز را برعهده دارد.

```
Response CallGetLastPcPosTransaction()
```

ورودی -۱-۶-۳-۶

این تابع ورودی ندارد.

خروجی -۲-۶-۳-۶

پاسخ نیز به صورت آبجکت Response بر میگردد. در مورد این آبجکت بعدا توضیح داده میشود.

مثال -۳-۶-۳-۶

```
PCPOS.PosCommander.Instance.GetLastTransaction();
```

۶-۳-۷- دریافت اطلاعات ترمینال (GetPosInfo)

این تابع ارسال دستور دریافت اطلاعات ترمینال به پوز را برعهده دارد.

`Response GetPosInfo()`

۶-۳-۷-۱ ورودی

این تابع ورودی ندارد.

۶-۳-۷-۲ خروجی

پاسخ نیز به صورت آبجکت Response بر میگردد. در مورد این آبجکت بعداً توضیح داده میشود.

۶-۳-۷-۳ مثال

`PCPOS.PosCommander.Instance.GetPosInfo();`

۶-۳-۸- چاپ متن (PrintText)

این تابع جهت ارسال متن برای چاپ بر روی رسید.

`Boolean PrintText(String text, Language lang)`

۶-۳-۸-۱ ورودی

ردیف	نام	نوع	توضیحات
۱-	text	Boolean	متن مورد نظر جهت چاپ بر روی رسید.
۲-	lang	Language	زبان (جهت چاپ- راست چین یا چپ چین)

۶-۳-۸-۲ خروجی

پاسخ نیز به صورت آبجکت Response بر میگردد. در مورد این آبجکت بعداً توضیح داده میشود.

توجه: جهت ایجاد خط جدید از کاراکتر "^\n" استفاده نمایید مثال "شهرداری تهران" واحد درآمد

۶-۳-۸-۳ مثال

`PCPOS.PosCommander.Instance.PrintText("شهرداری تهران", Language.fa);`

۶-۳-۹- چاپ بارکد (PrintText)

این تابع جهت ارسال بارکد برای چاپ بر روی رسید.

`Boolean PrintBarcode(String text, Language lang)`

۱-۹-۳-۶ ورودی

ردیف	نام	نوع	توضیحات
۱-	text	Boolean	متن مورد نظر جهت چاپ بر روی رسید.
۲-	lang	Language	زبان (جهت چاپ- راست چین یا چپ چین)

۲-۹-۳-۶ خروجی

پاسخ نیز به صورت آبجکت Response بر میگردد. در مورد این آبجکت بعدا توضیح داده میشود.

۳-۹-۳-۶ مثال

```
PCPOS.PosCommander.Instance.PrintBarcode("1234567890", Language.en);
```

۱۰-۳-۶ آبجکت Response

خروجی توابع Purchase و BalanceInquiry و GetLastTransaction و GetLastPcPosTransaction و GetPosInfo از نوع Response میباشد که به شکل زیر میباشد.

```
public class Response
{
    public Boolean Success;

    public byte ErrorCode = 0 ;
    public byte WarningCode = 0;

    public Transaction TransactionInfo = null;

    public PosInfo PosInformation = null;
}
```

```
public class Transaction
{
    public String PAN = ""; شماره کارت بانکی
    public String SVC = ""; شماره شناسایی در سمت بانک
    public String Stan = ""; شماره تراکنش در ترمینال
    public String RRN = ""; شماره ارجاع تراکنش
    public String Amount = ""; مبلغ تراکنش
    public String DateTime = ""; زمان انجام تراکنش
    public String ResponseCode = ""; وضعیت تراکنش

    public string EntryType { get; set; } روش ورود داده
}
```

```
public class PosInfo
{
    public String TerminalId = ""; // شماره ترمینال
    public String SerialNumber = ""; // شماره سریال سخت افزاری ترمینال
    public String MerchantId = ""; // شماره دارنده ترمینال
    public String MerchantName { get; set; } // نام و مشخصات دارنده ترمینال
}
```

- داده های این آبجکت وقتی معتبر است که مقدار Success برابر True باشد. و در صورتی که مقدار این فیلد False باشد، فیلد های ErrorCode و WarningCode معتبر خواهد بود و فیلد های TransactionInfo و PosInformation معتبر نخواهد بود.
- در رابطه با تابع GetPosInfo ، با وجود True بودن فیلد Success، مقدار TransactionInfo معتبر نیست و فقط PosInformation معتبر خواهد بود.
- در توابع Purchase و BalanceInquiry و GetLastTransaction، وقتی تراکنش **کامل و موفق** انجام شده است که در خروجی این توابع شرط روبرو برقرار باشد "00" == TransactionInfo.ResponseCode .

نمونه کدها ۱۱-۳-۶

تابع خرید ۱-۱۱-۳-۶

```
private void btnPurchase_Click(object sender, EventArgs e)
{
    PCPOS.PosCommander.Instance.Init(txtCOM.Text, int.Parse(txtTimeout.Text));

    Response res2 =
    PCPOS.PosCommander.Instance.Purchase(txtAdditionalDataBuy.Text, txtAmount.Text, chkPrint
    Bu.Checked, txtText.Text, txtBarcode.Text);

    if (res2.Success)
    {
        if (res2.TransactionInfo.ResponseCode == "00")
        {
            MessageBox.Show("شده انجام موفقیت با تراکنش");
        }
        ShowInfo(res2);
    }
    else
    {
        MessageBox.Show("Error Code =" + res2.ErrorCode.ToString());
    }
}
```

}

۶-۳-۱۱-۲- تابع پرداخت قبض

نکته :

توابع پرداخت قبض و خرید دارای دو ورودی برای متن و بارکد میباشند که در صورت ارسال متن خالی هیچگونه چاپی بر روی رسید انجام نخواهد شد. و در عوض میتوان مستقیماً توابع چاپ بارکد و چاپ متن را فراخوانی کرد. در زیر نمونه ای از کدهای مذکور آمده است.

```
private void btnBillPayment_Click(object sender, EventArgs e)
{
    ClearInfo();
    PCPOS.PosCommander.Instance.Init(txtCOM.Text, int.Parse(txtTimeout.Text));
    Response res2 =
        PCPOS.PosCommander.Instance.BillPayment(txtAdditionalDataBill.Text, txtBillId.Text, tx
        tPaymentId.Text, chkPrintBu.Checked);
    if (chkPrintBarcode.Checked)
    {
        PCPOS.PosCommander.Instance.PrintBarcode(txtBarcode.Text, Language.en);
    }
    if (chkPrintText.Checked)
    {
        PCPOS.PosCommander.Instance.PrintText(txtText.Text + "salam", Language.fa);
    }
    if (res2.Success)
    {
        if (res2.TransactionInfo.ResponseCode == "00")
        {
            MessageBox.Show("شد انجام موفقیت با تراکنش");
        }
        ShowInfo(res2);
    }
    else
    {
        MessageBox.Show("Error Code =" + res2.ErrorCode.ToString());
    }
}
```

۶-۳-۱۱-۳- تابع آخرین تراکنش

```
private void btnLastTransaction_Click(object sender, EventArgs e)
{
    PCPOS.PosCommander.Instance.Init(txtCOM.Text, int.Parse(txtTimeout.Text));

    Response res2 = PCPOS.PosCommander.Instance.GetLastTransaction();

    if (res2.Success)
    {
        if (res2.TransactionInfo.ResponseCode == "00")
        {
            MessageBox.Show("شد انجام موفقیت با تراکنش");
        }
    }
}
```

```

        ShowInfo(res2);
    }
    else
    {
        MessageBox.Show("Error Code =" + res2.ErrorCode.ToString());
    }
}

```

تابع دریافت اطلاعات پوز ۳-۱۱-۴-

```

private void btnGetPosInfo_Click(object sender, EventArgs e)
{
    PCPOS.PosCommander.Instance.Init(txtCOM.Text, int.Parse(txtTimeout.Text));

    Response res2 = PCPOS.PosCommander.Instance.GetPosInfo();

    if (res2.Success)
    {
        if (res2.TransactionInfo.ResponseCode == "00")
        {
            MessageBox.Show("شد انجام موفقیت با تراکنش");
        }
        ShowInfo(res2);
    }
    else
    {
        MessageBox.Show("Error Code =" + res2.ErrorCode.ToString());
    }
}

```

تابع دریافت موجودی ۳-۱۱-۵-

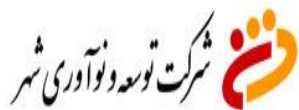
```

private void btnBalanceInquiry_Click(object sender, EventArgs e)
{
    PCPOS.PosCommander.Instance.Init(txtCOM.Text, int.Parse(txtTimeout.Text));

    Response res2 = PCPOS.PosCommander.Instance.BalanceInquiry();

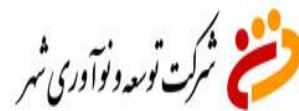
    if (res2.Success)
    {
        if (res2.TransactionInfo.ResponseCode == "00")
        {
            MessageBox.Show("شد انجام موفقیت با تراکنش");
        }
        ShowInfo(res2);
    }
    else
    {
        MessageBox.Show("Error Code =" + res2.ErrorCode.ToString());
    }
}

```



پروژه PAX.PCPOS

عنوان سند Developer Guid



صفحه ۱۵ از ۱۵

بازنگری: 04

کد مدرک: CDI-PAX.PCPOS-

طبقه بندی: محرمانه

}