



- *Active-X Controls and*
- *Business Objects Technical Reference*
- *for ISS45 8.1.1.0*
-
-

ISS45 Active-X Controls and Business Objects Technical Reference

Date of Issue	Product Identification Number	Part Number	Brief Description
June 1999	45001/041	No Part Number	Preliminary
October 2000	45001/041	89000014	Preliminary
April 2004	45001/041	89000176	With 8080
January 2006	45001/041	89000271	With 8110

**© Copyright StoreNext Retail Technologies LLC 1995-2006
All rights reserved**

This publication is protected by federal copyright law. No part of this publication may be reproduced or transmitted into any human or computer language in any form or by any means, stored in a retrieval system, transmitted, redistributed, translated or disclosed to third parties, or de-compiled in any way including, but not limited to, photocopy, photograph, electronic, mechanical, magnetic or manual without the express written permission of StoreNext Retail Technologies LLC or its licensors, if any. All copies, so authorized, shall contain a full copy of this copyright notice.

StoreNext Retail Technologies LLC endeavors to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. StoreNext Retail Technologies LLC makes no representation or warranties with respect to the contents hereof, and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose or non-infringement. No commitments by StoreNext or its suppliers are made from this documentation which is provided for information only.

Development of StoreNext products and documentation is continuous: StoreNext Retail Technologies LLC reserves the right to revise this publication and to make changes from time to time in the contents hereof or in the products herein described or discussed without notice and without any obligation of StoreNext Retail Technologies LLC to notify any person or organization of such revision or changes. Information published in this document will likely become obsolete over time and it is recommended that users regularly check for updates and newer versions.

StoreNext Retail Technologies LLC has prepared this manual for use by users, authorized third parties and personnel of StoreNext Retail Technologies LLC as a guide to the proper installation, operation, customization and/or maintenance of StoreNext Retail Technologies LLC equipment and software. The drawings and specifications contained herein are the property of StoreNext Retail Technologies LLC and/or its licensors.

Third-party products, services, or company names referenced in this document may be trademarked or copyrighted by their respective owners, and are for identification purposes only.

Copyrights, trademarks and license agreements shall be governed and construed in accordance with the laws of the State of Texas and the Federal Arbitration Act, and shall benefit Retailix, its successors, and assigns.

Address comments and corrections to:

StoreNext Retail Technologies LLC
Software Program Director
6100 Tennyson Parkway
Suite 130
Plano, Texas 75024



That's the **NEW** Point

Contents

CONTENTS 2

V8 BUSINESS OBJECTS 3

Overview..... 3

Object Types..... 4

Class Types..... 4

Loading V8 Objects..... 5

Loading V8 Objects with filtering 12

PROGRAMMING CONCEPTS 16

Common Methods and Properties..... 16

Project Settings 20

Reading Single Record..... 21

Updating/Inserting Single Record..... 21

Deleting Single Record..... 22

Exceptions and Errors..... 23

SAMPLE CODE 24

Declare objects..... 24

Initialize objects when loading the form 24

PLU lookup 25

Applying changes to database..... 26

Destroying objects..... 26

Setting Object Properties 27

SAMPLE BUSINESS OBJECTS 28

PLU 28

MEMBER PROMOTION 41

CashOffice..... 48



V8 Business Objects

This reference describes the business objects used by V8. It contains general explanations about those objects, programming overview, examples, and detailed list of methods and properties available by those objects.

Overview

V8 business objects are ActiveX DLL's. Each object is responsible for one or more subjects in the system. Those subjects can be tables (PLU, Promotion etc.), areas (Cash Office) and more.

The objects encapsulate all the business functionality needed to implement the subject. This functionality is usually made of:

- Reading object from database
- Writing object to database
- Updating QDX where needed, including sending maintenance to POS terminals
- Updating related tables.
- Business rules (validations)

V8 Objects infrastructure is based upon Three-Tier Architecture. Each tier can be recognized by its name that composed as follows:

- Tier prefix
 - V8G_ - represent the Presentation tier ("G" for GUI)
 - V8O_ - represent the Business (Middle) tier ("O" for Object)
 - V8P_ - represent the Data tier ("P" for Persist)
- Subject name

Right after the prefix coming the subject name.

For Example:

Presentation object -	V8G_PLU.DLL
Business object -	V8O_PLU.DLL
Data object -	V8P_PLU.DLL



That's the **Point**

Object Types

There are two kinds of V8 Business objects.

First, we have the objects, which we need to load with a list of keys.

(See paragraph – “Loading V8 Objects”)

Second, we have the filtered objects. Means, that in order to load them, we can either, send an empty recordset which represent that we like to get the whole collection, or, a recordset which represent the filtering, in order to get selected result. (We can also send a Sorting recordset)

(See paragraph – “Loading V8 Objects with filtering”)

Class Types

Basically, for each table in the system, there are 2 classes:

- **Single**

Handles a single object (a single PLU, a single department etc.). Single objects are implemented as a class that is usually named CDEP, CPlu, CDiscount etc.

Class name format:

Prefix: “C”

Suffix: Table name

- **Multi**

Handles multiple objects (list of departments, list of discounts etc). Multiple objects are implemented as a class that is usually named CDEPS, CPluS, CDiscountS etc.

Class name format:

Prefix: “C”

Suffix: Table name + “S”



That's the **Point**

Loading V8 Objects

While loading V8 single objects, using the method “Load”, we have to pass a list of keys as parameters.

Basically, the parameters list determines by the fields included in the Primary Key of the table, which linked to the object.

DII's Name	Class's Name	Linked table	Parameters list
V8BusObj	CAuditDetail	AUDIT_DETAIL	ID, FLD_NM
	CAudit	AUDIT	AUDIT_ID
	CCheckoutBankOpenLoan	CHKOUT_BNK_OPEN_LOAN	TND_CD, CHK_BNK_ID
	CCheckoutBank	CHKOUT_BNK	BNK_ID
	CCntrolCheck	CNTRL_CHECK	CC_NBR
	CCommunityPartner	COMUNTY_PRTNR	PRG_ID
	CCoupon	COUPON	COUP_NBR
	CCshrinkBnk	CASHR_CHK_BANK_ASSIGN	BANK_ID, EMP_ID
	CCustomerLnk	CUST_SURV_LNKD	BNK_ID, QST_ID
	CCustomerQst	CUST_SURV_QST	QST_ID
	CDepGroup	DEP_GROUP	GRP_ID
	CDep	DEP	STR_HIER_ID,
	CDiscount	DISCOUNT	DISC_NBR
CEmp	EMP	EMP_ID	



That's the **Point**

	CExcise Tax	EXCISE_TAX	EXCISE_TAX_NBR
	CMsgCenter	MSG_COUNTER	MSG_TYPE_ID
	CMsgQueue	MSG_QUE	ID, MSG_TYPE, EMP_ID, TER_ID, BNK_ID
	CposMenuLine	POS_MENU_LINES	MNU_TYPE, MNU_NBR, LINE_NBR
	CPosMenu	POS_MENU	MNU_TYPE, MNU_NBR

	CpostItemMsg	POST_ITM_MSG	MSG_NBR
	CReturn Type	RTN_TYPE	RTN_NBR
	CSysParameter	SYS_PRM	NAME
	CTareWeight	TARE_WGT	TAR_WGT_NBR
	CTermina	TER	TER_ID
V80_CashOffice	CViewLoanPickup	LOAN_PICKUP	DT, EMP_ID, TER_ID, TND_CD
V80_PLU	CDEAGroup	DEA_GRP	ID
	CPlu	PLU	ITM_ID
	CPluPriceReason	PLU_PRC_RSN	PRC_RSN_ID
	CPriceGroup	PRC_GRP	PRC_GRP_ID
V80_Plu	CPluBat	PLU_BTCH	BTCH_NBR



That's the **Point**

Batch	ch		, DT_CREATED
V8O_Plucategory	CPLUCategoryField	PLU_CTG_FLD	ID_FLD, FLD_NAME_TXT, TBL_NAME, DEF_VAL
	CPLUCategory	PLU_CTG	ID_CTG
	CPLUFilterCategory	PLU_FLTR_CTG	FLTR_ID, CTG_ID
	CPLUFilter	PLU_FLTR	FLTR_ID
	CPLUFilterValue	PLU_FLTR_VAL	VAL_ID, SEQ_NBR, FLD_ID, TBL_NAME
	CPLUReportEnv	PLU_RPT_ENV	RPT_ID, RPT_TYPE
V8O_Alrt	CAlrt	ALRT	ALRT_ID
	CAlrtAssgn	ALRT_ASSGN	ALRT_ID, TRGT_ID
	CAlrtTran	ALRT_TRAN	TRAN_ID, ALRT_ID, POS_PC_FG, POS_PC_ID, GEN_TM
	CAlrtTrgt	ALRT_TRGTS	TRGT_ID
V8O_EmailSend	CEmailSend	EMAIL_SEND	LOYAL_CUST_NUM, TRAN_FILE_PTR
V8O_PostEmail	CPosEmail	POST_EMAIL	MailID
V8O_SystemPrm	CCpnPrnMaint	CPN_PRN_MAINT	ID



That's the **NEW** Point

	CcpnValueCode	CPN_VAL_CODE	ID
	CMultiStampScheme	MULTIPLE_STAMP_SCHEMES	ID
	CQdxFileChecksum	QDX_FILE_CHECKSUM	ID
	CRestrictSales	RSTRCT_SLS	ID
V80_SystemsPrmGeneral	CSysPrmGen	SYS_PRM	ID, NAME
V80_Table	CEcca	ECCA	ECCA_CD
	CTareWgt	TARE_WGT	TAR_WGT_NBR
V80_Tasks	CTasks	TASKS	NAME
V80_EodInt	CEodAutoTimes	EOD_AUTO_TIMES	DAY_ID
	CEodExpDate	EOD_EXP_DATE	DAY_ID
	CPosEodListPersists	POS_EOD_LIST	POS_NBR
V80_ReportSchedule	CReportArchive	ReportArchive	DDateActive, iReportId, iUserDefined
	CReportSource	ReportSource	iReportSourceId
	CRptAttributes	ReportAttributes	iReportId
	CRptParameter	ReportParameter	IReportId, iParamNo
	CRptPrmVal	ReportParamValues	IJobId, iParamNo



That's the **Point**

	CRptSchedule	ReportSchedule	iJobId
	CRptType	ReportType	iReportType
V8O_MmbrProm	CClbCrdScheme	CLB_CRD_SCHM	SCHM_ID
	CCreditProg	CRDT_PROG	CRDT_PROG_ID
	CLoyalCust	LOYAL_CUST	LOYAL_CUST_NUM
	CMmbrPromLnkd	MMBR_PROM_LNKD	MMBR_PROM_ID, LNK_TYP, LNK_ID
	CMmbrProm	MMBR_PROM	MMBR_PROM_ID
	CSegments	SEGMENTS	SEGMENT_ID
V8O_Language	CControls	Controls	lLanguageId, sProject, sFormName , sControlName, iSubControlId, iCollItemId
	CEdtMenu	MenuObjectStrings	lLanguageId, iObjectId
	CEdtMsg	MsgValue	lMessageId, iLanguageId
	CLanguages	Languages	iLanguageId
	CReportStrings	ReportStrings	lLanguageId, lStringId, sReportName
	CStrLng	Strings	lLanguageId, lStringId
V8O_Ser	CFee	FEE	FEE_ID



That's the **Point**

viceFee	CFeeRange	FEE_RANGE	FEE_ID, RANGE_ID
V80_GenBatch	CGenBatchHeader	GEN_BTCH_HDR	BTCH_NBR, DT_CREATED, BTCH_TYPE,
	CGenBatchType	GEN_BTCH_TYPE	ID
	CGenBatchError	GEN_BTCH_ERROR	BTCH_NBR, DT_CREATED, BTCH_TYPE
	CBatchMemberLink	GEN_BTCH_MMBR_PROM_LNKD	GEN_BTCH_NBR, DT_CREATED, OPCODE, MMBR_PROM_ID, LNK_TYP, LNK_ID
	CBatchSubDep	GEN_BTCH_SUB_DEP	GEN_BTCH_NBR, DT_CREATED, OPCODE, SUB_DEP_NBR
	CBatchPluGroup	GEN_BTCH_PLU_GROUP	GEN_BTCH_NBR, DT_CREATED, OPCODE, GRP_ID
	CBatchMemberProm	GEN_BTCH_MMBR_PROM	GEN_BTCH_NBR, DT_CREATED, OPCODE, MMBR_PROM_ID



That's the **Point**

	CbatchLoyalCust	GEN_BTCH_LOYAL_CUST	GEN_BTCH_NBR, DT_CREATED, OPCODE, LOYAL_CUS_NUM
	CbatchBadAccount	GEN_BTCH_BAD_ACCOUNT	GEN_BTCH_NBR, DT_CREATED, OPCODE, ACCT_NBR
	CbatchCrdtProg	GEN_BTCH_CRDT_PROG	GEN_BTCH_NBR, DT_CREATED, OPCODE, CRDT_PROG_ID
	CbatchDep	GEN_BTCH_DEP	GEN_BTCH_NBR, DT_CREATED, OPCODE, STR_HIER_ID
	CBatchFuelGrade	GEN_BTCH_FuelGrades	GEN_BTCH_NBR, DT_CREATED, OPCODE, Gradeld

Loading V8 Objects with filtering

While working with the second type of objects, we can use a filter object in order to send the exact filtering we like to get from the collection (Multi object) by using the V8_DataManager.Dll

V8_DataManager.Dll, has two important classes:

- ☞ **CFilter** – which define a recordset (ADO) representing the filtering, using two methods: ADD and CLEAR.

- ☞ **CSort** – which define a recordset (ADO) representing the sorting,

Using two methods: ADD and CLEAR.

CFilter – “ADD” Method

The “ADD” method has the following parameters list:

Property name –

The name of the field in the object (V8O_Tender, V8O_Dep, etc...) We like to filter.



That's the **Point**

Property Value –

The field's value we like to do filter on.

Operator –

Operator between the field and it's following field.

Following is the operator's enumeration:

- POSCFilterOpNone = 1
- POSCFilterOpEqual = 2
- POSCFilterOpNotEqual = 3
- POSCFilterOpLessOrEqual = 4
- POSCFilterOpGreaterOrEqual = 5
- POSCFilterOpLess = 6
- POSCFilterOpGreater = 7
- POSCFilterOpLike = 8

PreLinkOperator –

logical operator.

Following is the operator's enumeration:

- POSCFilterLinkOpNone = 1
- POSCFilterLinkOpAnd = 2
- POSCFilterLinkOpOr = 3
- POSCFilterLinkOpLeftParenthesis = 4
- POSCFilterLinkOpRightParenthesis = 5
- POSCFilterLinkOpNot = 6
- POSCFilterLinkOpNotAnd = 7
- POSCFilterLinkOpNotOR = 8

CSort – “ADD” Method

The “ADD” method has the following parameters list:

Property name –

The name of the field in the object (V8O_Tender, V8O_Dep, etc...) We like to filter.

Order Type –

Ordering type according the following enumeration:

- POSOOrderTypeDesc = 1
- POSOOrderTypeAsc = 2

CSort & CFilter– “CLEAR” Method

Clearing the filtering and sorting object.

Example:



That's the **Point**

```

Dim objFilter as DataObject.CFilter
Dim objSort as DataObject.CSort
Dim objTender as V8O_Tender.CbadAcct
Set objTender = V8O_Tender.CbadAcct
Set objFilter = New DataObject.CFilter
Set objSort = New DataObject.CSort
objSort.Add "AccountNumber", POSOrderTypeAsc
objFilter.Add "AccountNumber", "981435"
objTender.CastObject.Load objFilter.GetRecordset,
objSort.GetRecordset
  
```

DI l's N a m e	Class's Name	Property list
V 8 O - D ep	cDep	CFTaxRate7, CFTaxRate8
	CDepGr oup	Description, GroupID, NoneMDSE, SouperGroup
	CSubDe p	GroupID, DeparementID, Description SubDepartmentID

	CsuperG roup	SprDescription, sprNonBankAble, sprID
V 8 O - T en de r	CBadAcc t	ACCT_NBR = AccountNumber;CNTRL_C HCK_NBR = CCNumber
	CBadAcc tEnh	AccountID, AmountOWING, AmountTendered, ControlCheckNumber, Description, DateTendered, Initials, OrgStoreNumber, SeqNumber, SpecialDescr, Tender



That's the **Point**

	CCardsRange	AllowedAuthoKeyFlg, AuthoFlg, CardUseage, CashBackSchMax, CashBackSchMin, ClubCardCode, CountFileFlg, CreditCardFlg, CreditRna
	CCrdVld	ControlCheckNumber, CheckMMYYFlg, CardValidCode, Description, DateExpReqFlg, DateFormat, DateManualFlg, ExpirationDate, ExtraIDOption, ExtraC
	CEcca	Address, BadAccountControlCheck, BadAccountFlg, CheckFileFlg, Damount, DcashBack, DcheckAmountFlg, DcheckCashBackFlg, DcheckCountFlg, Dco
	CTender	AcceptRefund, AcceptSales, AccountNumber, AccountOption, AccountPrmpt, AccountRequired, AdditionalTenderInform, AddFloat
	CTenderGroup	TenderGroupName, TenderGroup
V8OPUGr	CPLUGroup	GroupID, Description, QDX_OP_CODE



That's the **NEW** Point

ou p		
---------	--	--

Programming Concepts

This section describes how to write programs that use V8 business objects. Since most objects are similar in their concepts, the concepts described below apply to most V8 business objects. Once you understand them and have managed to use one object, using other should be quite similar.

Common Methods and Properties

The following concepts (object types, method, properties) are common to all objects in the system (PLU, Department, Promotion etc.).

“Single” Object

The single object is made of the following parts:

- Property IsNew
- Property IsDirty
- Property IsDeleted
- Property CastObject
- Properties and methods unique to the specific object.

Common Properties (Read Only Properties)

These properties are common to all single objects and they indicate the object status. These properties are:

Property Name	Description
IsNew	Indicates that the lookup item is not found in the database. This property is set after calling the Load() method, which reads the record from database. If the item was found, IsNew will be FALSE. If the item was not found, IsNew will be TRUE.



That's the **Point**

Property Name	Description
IsDeleted	Indicate intention to delete the object. This property is set after the Delete() method is called.
IsDirty	Indicate that the object one or more of the object properties were modified.

CastObject

CastObject is an object inherited by all business objects. It contains the basic methods and properties common to all objects.

These methods are used to implement the general interface business objects.

Name	Description
Load(ByVal vKey1, Optional ByVal vKey2, Optional ByVal vKey3, Optional ByVal vKey4, Optional ByVal vKey5)	Reads object from database. Pass object keys (PLU number, promotion number etc.) to this method to read the item. The IsNew property will be set true or false if the record was found or not.
Edit()	Allow editing the object. Before you change any of the object properties (price, name, etc.) or add new item you must call this method to switch the object into edit mode.
CancelEdit()	Cancel any changes to the object and restore its initial values (after load or reset)
Apply()	Save changes to database.
Delete()	Put the object in delete state.



That's the **Point**

Name	Description
	After calling Apply the object is deleted from storage.
IsEdit	The object is in edit mode
IsNew	The object is new. See above.
IsValid	The object is valid. Run all business rule validations of this object to make sure the record is valid.
AddNew()	Private to PLU
IsDeleted	The object is marked for delete. See above.
IsDirty	Object is dirty (properties were changed). See above.

Unique Properties (Read/Write Properties)

These properties differ from one object to the other. On the PLU object they will be PLU number, description, price etc. On department they will be department number, name, active etc.

“Multi” Object

The multi object is made of the following parts:

- Property CastObject
- Method Item
- Properties and methods unique to the specific object.

CastObject

CastObject is an object inherited by all business objects. It contains the basic methods and properties common to all objects:

Name	Description
Load()	Read all records into the “Items”



That's the **Point**

Name	Description
	collection (see below)
MoveFirst()	Move to first record in collection
MoveLast()	Move to last record in collection
MoveNext()	Move to next record
MovePrevious()	Move to previous record
Refresh()	Refresh collection
RemoveCurrent ()	Delete current record from the "Item" collection
GetOrder()	Get order as string
Bookmark	Bookmark of current record in collection. Updated as you move through the collection.
Count	Number of records in collection. Set after calling Load()
IsFirst	Is it the 1 st record in collection
IsLast	Is it the last record in collection
OrderBy	Determines if Load() sorts records in ascending/descending order

Item

Item() enables you to access the collection of records read by the Load() method. Records are kept in a collection, which can be accessed using an index number.

For example: DepS.Item(1). Description



Project Settings

When starting a new program, you must first create reference to the objects you want to use.

VB

Add a reference to relevant object. Usually it will be “v8 Business logic objects”, found in POSWare\Office\Components\DII\v8busobj.dll

VC++

- Add the following to your application STDAFX.H:

```
#import "C:\Program Files\POSWare\Office\Components\DII\v8Intf.dll"  
#import "C:\Program Files\POSWare\Office\Components\DII\v8BusObj.dll"
```

- Add the following to your class .H files:

```
using namespace v8BusObj ;
```

Initialization

Create object, single or multi, according to your need.

Define variable of the object type you need. Initialize it. For example:

VB

```
Dim oPlu As Cplu          \ "Single" PLU object  
Dim oDepS as CdepS      \ "Multi" department  
object  
Set oPlu = New Cplu  
Set oDepS = new CDepS
```

VC++

```
_CPluPtr      pPlu;      \ "Single" PLU object  
_CDepSPtr     pDep;     \ "Multi" department  
object  
pPlu = _CPluPtr(__uuidof(CPlu)) ;  
pDep = _CDepSPtr(__uuidof(CDepS)) ;
```



Reading Single Record

In order to read a record, you need to use the Load() method

- Call the Load method in the CastObject object to read single record.
- Check the “IsNew” property to determine if record was found or not.

VB

```
oPlu.CastObject.Load PluNumber
If oPlu.IsNew Then
    ` item not found
Else
    ` item found
endif
```

VC++

```
pPlu->GetCastObject()->Load(PluNumber);
if ( pPlu->IsNew )
{
    // item not found
}
else
{
    // item found
}
```

Updating/Inserting Single Record

Writing changes to the database requires the following steps:

- Calling the Load() method to read the item
- Set the object to edit mode.
- Set object properties
- Validate data



That's the **Point**

- Apply changes

VB

```
If Not oPlu.CastObject.IsEdit Then
    oPlu.CastObject.Edit
End If
oPlu.RetailPrice = txtPrice
If oPlu.CastObject.IsValid Then
    oPlu.CastObject.Apply
Else
    MsgBox "Invalid data"
End If
```

VC++

```
if ( !pPlu->CastObject->GetIsEdit() )
    pPlu->CastObject->Edit() ;
pPlu->PutDepartmentID(m_Dep);
if ( pPlu->CastObject->GetIsValid() )
{
    pPlu->CastObject->Apply();
}
else
{
    AfxMessageBox("Invalid data");
}
```

Deleting Single Record

To delete a record from database:

- Load the record
- Set object state to edit mode
- Delete record
- Apply changes to database



VB

```
oPlu.CastObject.Edit  
oPlu.CastObject.Delete  
oPlu.CastObject.Apply
```

VC++

```
pPlu->CastObject->Edit();  
pPlu->CastObject->Delete();  
pPlu->CastObject->Apply();
```

Exceptions and Errors

All business objects raise errors in case of errors and illegal operations.

Exceptions might be raised when trying to perform illegal operation (like accessing a read only property, deleting a non-existing item). Exception may also be raised due to violation of business rules. For example: setting a zero price to an item.

You must add exception handling to your code and handle those errors.



Sample Code

Below is a piece of source sample written in Visual Basic 5.0.

Full sample code package in VB and VC is also available.

Declare objects

```
Dim oPlu As CPlu
Dim oDeps As CDepS
Dim oDep As CDEPDisplay
Dim oErr As cErrService
Dim cnt As Control
```

Initialize objects when loading the form

```
Private Sub Form_Load()
Set oPlu = New Cplu ' PLU object
Set oDeps = New CdepS ' Departments collection
oDeps.CastObject.Load ' loading the collection
Set cnt = txtDesc
//Populating combo box with department names
iCount = oDeps.CastObject.Count
For i = 1 To iCount
Set oDep = oDeps(i)
cmbDep.AddItem oDep.Description
cmbDep.ItemData(cmbDep.NewIndex) = oDep.Depld
oDeps.CastObject.MoveNext
Next
End sub
```



That's the **Point**

PLU lookup

```
Private Sub cmdLookup_Click()  
Dim pe As PosError  
On Error GoTo errr  
ClearControls ' clear the screen controls  
Form1.MousePointer = 11  
oPlu.CastObject.Load Val(txtId) //loading the PLU with the ID.  
If Not oPlu.IsNew Then ' Item was found. Loading controls from object.  
    txtDesc = oPlu.DisplayDescription  
    txtPrice = oPlu.RetailPrice  
    txtDep = oPlu.DepartmentID  
    cmbDep.ListIndex = oDeps("k" & oPlu.DepartmentID).Position - 1  
    txtDesc.SetFocus  
    lblStatus = "Exists"  
Else ' Item not found  
    MsgBox "PLU not Found"  
    txtId.SetFocus  
    txtId.SelStart = 0  
    txtId.SelLength = Len(txtId)  
    lblStatus = "New"  
End If  
End If  
Form1.MousePointer = 0  
Exit Sub  
errr:  
    Set oErr = New cErrService  
    oErr.ErrorHandle "PLU Demo", Log_Msg, Err  
Exit Sub
```



*That's the **Point***

End Sub

Applying changes to database

```
Private Sub cmdApply_Click()  
On Error GoTo errr  
If Not oPlu.CastObject.IsEdit Then  
    oPlu.CastObject.Edit  
End If  
SetObject  
If oPlu.CastObject.IsValid Then  
    oPlu.CastObject.Apply  
End If  
txtId.SetFocus  
Exit Sub  
errr:  
MsgBox Err.Description  
Exit Sub  
End Sub
```

Destroying objects

```
Private Sub Form_Unload(Cancel As Integer)  
    Set oPlu = Nothing  
    Set oDeps = Nothing  
End Sub  
' Procedure to clear screen  
Private Sub ClearControls()  
    txtDesc = ""  
    txtPrice = "0"  
End Sub
```



*That's the **Point***

Setting Object Properties

```
Private Sub SetObject()
```

```
    oPlu.DisplayDescription = txtDesc
```

```
    oPlu.RetailPrice = txtPrice
```

```
    'oPlu.DepartmentID = txtDep
```

```
    oPlu.DepartmentID = oDeps(cmbDep.ListIndex + 1).DepId
```

```
End Sub
```



Sample Business Objects

This section describes some of V8 business objects. Since most objects share a lot in common, we describe here only the properties and methods that are unique to each object. The common parts were listed above.

PLU

This section describes properties, methods and other things unique to the PLU object only.

Other methods and properties available by all objects are described under "Programming Methods and Properties".

As for now, there is no "Multi" object for the PLU object. This object will be provided with later versions of V8. In order to scan the PLU tables, use standard VB/VC++ database functions.

Unique Properties - Read/Write

Name	Description
ItemID	PLU item id
DepartmentID	Department ID
FromBatch	Source of the change(Internal)
APISourceName	Name of external device ("Telxon")
CastObject	Property that exposes Business objects interface that is a general interface to all POS business object.
PriceReasonID	Price change reason ID.
DefaultReturnLoc	Future V9
MessageId	The POS message Id link to the item.
DisplayDescripti	Full Description of the Item.



That's the **Point**

Name	Description
on	
SaleRestrict	Pointer to sale restriction group.
ReceiptDescription	The description of the item on ticket.
TaxabilityID	V9 pointer to taxability group.
MerchandiseCrossRe	Future V9 pointer to merchandise category.
NoneMerchID	Future V9 pointer to non merchandise category.
UnitOfMeasure	Pointer to Unit of measure table. Not Used.
UnitQty	The quantity for one unit of sale of item.
LineItemCode	Future V9
MarkDown	Future V9
QtyRequired	The sale quantity is required
SubProductCount	Future V9
QtyAllowed	Future V9
SaleAuth	The item can be sold.
FoodStamp	Payment by food-stamps allowed
WIC	WIC item.
PerpetualInventory	Future V9.
RetailPrice	The sale price of the item
RetailPriceDate	Date of the last retail price.
UnitCost	The cost price of item.
ManualPriceLevel	Future V9



Name	Description
MinimumMerch Amount	Future V9
SerializedMerch	Future V9
CenterPrice	The head office price of the item.
CenterPriceDate	The date of the head office price of the item.
MaxMerchAmount	Future V9.
NegativeEntry	Is Negative item.
StoreCoupon	Is store coupon
VendorCoupon	Is vendor coupon.
ManualPrice	The price of the item must entered manually during the sale.
NoneDiscount	Item can't have discount.
CostPlus	The item is cost plus item.
PriceVerify	Future V9
PriceOverride	Future V9
SupplierPromotion	Future V9
SaveDiscount	Future V9
ItemOnSale	The item is on sale.
InhibitQtyFlag	Prohibit quantity flag
DecimalQtyFlag	The item is sold in decimal quantity.
ShelfLabelsRequired	The Item must have shelf label.- Future V9
TaxRate1	Connected to tax id 1



That's the **Point**

Name	Description
TaxRate2	Connected to tax id 2
TaxRate3	Connected to tax id 3
TaxRate4	Connected to tax id 4
TaxRate5	Connected to tax id 5
TaxRate6	Connected to tax id 6
TaxRate7	Connected to tax id 7
TaxRate8	Connected to tax id 8
CostPerCasePrice	Cost per case price.
UnitCase	The number of units of the item in the case.
MixMatchID	The mix and match code id.
ReturnCode	The return code id.
FamilyID	Family code number 1.
SecondFamily	Family code number 2.
SubDepartmentID	Sub department Id.
DiscountID	Discount Id in discount table
LabelQty	Future V9 Label quantity
ScaleFl	Future V9
HeadOfficeDepartmentID	Head Office department ID.
WeightScale	Weight On scale
FreqShopType	Frequent shopper type
FreqShopVal	Frequent shopper value
POSMessage	Id of POS message linked to item.
ShelfLifeDay	Shelf life time in days V9



That's the **Point**

Name	Description
PromotionID	Promotion ID linked to item.
BucketNumber	NR
ExtendedPromotionID	NR
ExtendedBucketID	NR
ReceiptDescription2	Second receipt description for multilingual support
ReceiptDescription3	Third receipt description for multilingual support
ReceiptDescription4	Fourth receipt description for multilingual support
CouponNumber	Coupon number
TareWeightNumber	TARE weight ID number
RestrictionLayout	Restriction layout ID.
InternalID	Internal code ID.
OldPrice	Previous price.
VendorID	Vendor ID
VendorItemID	Vendor item ID.
VendorItemSZ	Vendor item description.
ComparativeUOM	Comparative UOM
BonusBuyOpcode	Bonus buy opcode- always 4.
BonusBuyDescription	Bonus buy description
ComparisonType	(0-5) 0 No comparison to be done 1 Bonus buy saving 2



Name	Description
	Advertised saving 3 Match advertised saving 4 private label saving 5 competitor compare saving.
ComparsionPrice	Comparison price.
ComparsionQty	Comparison quantity
ComparsionDate	Comparision date
CompareQty	Compare quantity
CompareUnit	Compare unit
AssumeQuantity	Assume quantity
DEAGroup	DEA group ID.
FreqShopLimit	Frequent shopper limit
PluHaveCheckDigit	Plu is with check digit.
AllowDiscountGreaterThenPrice	Override business rule and allow discount greater than price.
EnhancedFrequentShopper	NR
ItemOnPromotion	Item has active promotion.
ExciseTaxID	Excise tax id
ZeroPriceAllowed	Allowed Zero Price
UpdateLinkedItemsByInternalID	Update Linked Items By Internal ID
CopyDataFromDepartment	Copy Data From Department
HostDelete	Delete host delete flag



Name	Description
LocalDelete	Delete Local flag
SuperAuthRequired	Super authorization required
SwwCode	
SspProduct	
SkipQdxMaint	Skip QDX
SkipAudit	Skip Audit
MerchandiseCrossRef	Merchandise Cross Reference
BonusCoupon	Bonus coupon
OriginalPrice	Original Price
DeletePLU	Set the selected item for deletion
DeleteDate	The deletion date
OriginalUnitQty	Original Unit Qty
ExcludeMinPurchase	Exclude Min Purchase
PriceGroupID	Price Group ID
ItemPoint	Item Point
FuelFlag	Fuel flag
ShelfLifeDay	Shelf life number of days 0-999
ForceDelete	This flag force to delete PLU ignoring the systems parameter

Methods

The following methods are unique to the PLU object.

Name	Description
ResetObj()	Create new empty objects and reset its properties to empty values.
Validate()	Apply business rules against object. This method is also executed automatically before applying changes.
AddNew()	Create new PLU object.
BR_CheckPLULinkedToSubDep()	Check if current PLU Linked To Sub Department
BR_CheckDigitValidation(ByValue As Integer) As Boolean	Check Digit Validation
BR_ItemIsUPC2(Optional ItemID As Integer) As Boolean	Check if an item is UPC2
BR_ItemID(ByValue As Integer) As Double	Check the item number
BRCheckPromotion() As Integer	Check if the current item is at promotion 0 - No Promotion 1 - Promotion On PLU 2 - Promotion On Dep. 3 - Promotion On Mix & Match



Name	Description
	4 - Promotion On Mix & Match AND Dep
BR_ActiveDepartmentPromotion(iDep As Long)	Check if the given iDep is on promotion
BR_ActiveMixMatchFiveDigitPromotion(IMixMatch As Long)	Check if the given IMixMatch is on active promotion
CopyPLU(ByRef dbItemID As Double)	This function receives a Plu number, checks if given Plu exists and if not, adds a new Plu by this number and returns true , otherwise returns false

Object Validations and Error Codes

The following are error codes and their descriptions. The PLU object raises these errors.

Number	Description
10001	String too long
10002	Invalid item number
10003	Invalid check digit In item number
10004	Invalid Internal Number
10005	Weighted item and decimal



That's the **Point**

Number	Description
10006	Force Prohibit Qty
10007	Invalid Department
10008	Invalid Negative Entry
10009	Display Description Too Long
10010	Receipt Description Too Long
10011	Zero Price Not Allowed
10012	Discount Not Allowed
10013	Frequent shopper Discount Not
10014	Discount Greater than Price
10015	Invalid Store Vendor
10016	Invalid Frequent shopper Discount
10017	Plu Must Be Linked To Sub dep
10018	Invalid Plu Price
10019	Invalid Cost Per Case Price
10020	Department 0 not allowed
1002	Bonus Buy Description Too long



That's the **Point**

Number	Description
1	
1002 2	Wrong Date
1002 3	Item Not Exist
1002 4	The excise tax you selected not exists.
1002 5	Invalid Comparision Type
1002 6	Enhanced Frequent shopper in the
1002 7	Invalid Arguments for compressed code
1002 8	Item on promotion
1002 9	Item on promotion By Department
1003 0	Item on promotion By MixMatch
1003 1	The coupon you selected not exists.
1003 2	The department you selected not exists.
1003 3	The sub department you selected not exists.
1003 4	The tare weight you selected not exists.
1003 5	The discount you selected not exists.
1003 6	The return code you selected not exists.



That's the **NEW** Point

Number	Description
1003 7	The pos item message you selected not exists.
1003 8	The DEA group you selected not exists.
1003 9	The change price reason you selected not exists.
1004 0	Item On sale
1004 1	Active Promotion By MixMatch
1004 2	Active Promotion By Department
1004 3	Invalid Source item number
1004 4	Copy to the same Plu number
1004 5	Invalid Destination item number
1004 6	Item on Promotion by Manufacture
1004 7	Item Linked To Grade
1004 8	Item Must Linked To Grade Department
1004 9	Fuel in the system flag is off
1005 0	Negative Entry flag set
1005 1	Coupon flag set
1005	Prohibit changes to Negative Entry flag



Number	Description
2	
1005 3	Prohibit Manual Changes To Non Merchandise Flag
1005 4	The department you selected is not active
1005 5	Invalid Internal code check digit
1005 6	Shelf life day should be between 0 to 999



MEMBER PROMOTION

This section describes properties, methods and other things unique to the Member promotion and Member promotion linked only.

Other methods and properties available by all objects are described under “Programming Methods and Properties”.

Member Promotion uses the parent - child technique. Which mean that single promotion has multiple child objects and they are linked to single child object. The child objects can also be used as a regular Multi and Single objects.

Member promotion (Parent):

Unique Properties - Read/Write

Name	Description
PromId	Member Promotion id
PromType	Member Promotion type- 1.Immmediate 2.Enhanced 3.groupPromotion
StrtDate	Promotion Start activating date
EndDate	Active Promotion activating date
ActvFg	This property is not used. We use the start date and end date to calculate the activation of the promotion.
Description	Promotion description.
CrddpndncyFg	Member card required.
AllSchemAvail	Promotion available for all card scheme.
StrHierId	Mark down department. A department that takes the reward charge.
RwrTyp	Reward Type.



Name	Description
	1-Amount off(cash). 2-Percentage off. 3-Points. 4-Per item discount. 8-New price. 9-Amount off per pound. 10-Amount off per item. 11-New price per pound. 12-Target message. 13-Percent of order. 14-Amount of order. 15-Free item.
RwrldVal	Reward value.
LmtQty	Limit Quantity of items to buy.
SunFg	Promotion specific day active. Sunday.
SunStrtTm	Promotion specific day start time. Sunday.
SunEndTm	Promotion specific day end time. Sunday.
MonFg	Promotion specific day active. Monday
MonStrtTm	Promotion specific day start time. Monday
MonEndTm	Promotion specific day end time. Monday
TueFg	Promotion specific day active. Tuesday
TueStrtTm	Promotion specific day start time. Tuesday
TueEndTm	Promotion specific day end time. Tuesday
WedFg	Promotion specific day active. Wednesday
WedStrtTm	Promotion specific day strt tm Wednesday



Name	Description
WedEndTm	Promotion specific day end tm Wednesday
ThuFg	Promotion specific day active. Thursday
ThuStrtTm	Promotion specific day start time Thursday
ThuEndTm	Promotion specific day end time Thursday
FriFg	Promotion specific day active. Friday
FriStrtTm	Promotion specific day start time. Friday
FriEndTm	Promotion specific day end time. Friday
SatFg	Promotion specific day active. Saturday
SatStrtTm	Promotion specific day start time. Saturday
SatEndTm	Promotion specific day end time. Saturday
Grp1Type, Grp#Typ 1-10	Group type 1-units. 2-weighted. 3- By Value. 4-Points (in groups it's 1-10). In enhanced the group type field is grp1type.
Trshold#Qty 1-10	Threshold. The delta needed for getting the next promotion. (in groups it's 1-10 according the group type). In enhanced it's 1 and 2.
Trig#Fg 1-10	Triggers. Used in groups. The reward charge will be on the same group number.
StrtTm	Start time of the Promotion



Name	Description
EndTm	End time of the Promotion
Schmld# 1-10	Card scheme Id 1-10.
LowHighFg	The reward will be given on the highest priced item or the lowest priced item
MinValue	Minimum item value.
MaxWeight	Maximum weight value.
MinWeight	Minimum weight value.
MinPurchase	Minimum Purchase value.
DelayedPromFg	Delayed promotion-will be given at the end of the ticket.
ValByCshierFg	Reward value entered by cashiers.
CouponRqrd	Coupon required – PLU code.
LinkedProm	Linked promotion to promotion



Methods

The following methods are unique to the Member Promotion object.

Name	Description
UserID ()	For Audit use.
OperatingSource ()	For Audit use.

Object Validations and Error Codes

The following are error codes and their descriptions. The Member Promotion object raises these errors.

Number	Description
21401	Description is too long
21402	Promotion type is not between 1-6.
21403	Date is not valid
21404	Time is not valid
21405	Invalid Group Type Number
21406	Invalid Reward Type Number
21408	Member Promotion Id Should be between 1 to 9999
21409	Reward value Should be between 0 to 9999.99
2141	Limit Quantity Should be between 0 to



Number	Description
0	429000
2141 1	Threshold Quantity Should be between 0 to 999999
2141 2	Threshold Weight Should be between 0 to 999.999
2141 3	Threshold Value Should be between 0 to 9999.99
2141 4	Invalid Reward value
2141 5	You should choose at least one group.
2141 6	Invalid Minimum Item Value
2141 7	Invalid Minimum Weight
2141 8	Invalid Minimum Purchase
2142 0	Invalid Linked Promotion
2142 1	The about to delete Promotion is also Linked Promotion

Member promotion Linked:

Unique Properties - Read/Write

Name	Description
PromId	Member Promotion id
LnkType	Linked type- 0.PLU 1.Department 2.Mix & Match 3.Manufacture 4.Price group



Name	Description
LnkId	Linked Id
GrpId	If working with groups. 1-10

Methods

The following methods are unique to the Member Promotion object.

Name	Description
UserID ()	For Audit use.
OperatingSource ()	For Audit use.
SendToQDX()	Saving Info to qdx after the saving transaction to the sql DB was over.
SaveToAudit()	writing changes to Audit object after the saving transaction to the sql DB was over.

Object Validations and Error Codes

The following are error codes and their descriptions. The Member Promotion object raises these errors.

Number	Description
21701	Description is too long
21702	Invalid Promotion Linked Item Type
21703	This Plu Item does not exist
21704	This Department Item does not exist



Number	Description
2170 5	This Mix And Match does not exist
2170 6	Group Id should be between 1 to 3

CashOffice

This section describes properties, methods and other things unique to the CashOffice object only.

Other methods and properties available by all objects are described under "Programming Methods and Properties".

Cash Office objects linked to several tables.

CCashOfficeS \ cCashOffice work with:

SHIFT_TER_DATA \ SHIFT_EMP_DATA

SHIFT_TER_MEDIA \ SHIFT_EMP_MEDIA

CCashOfficeS \ cCashOffice work with:



That's the **Point**

LOAN_PICKUP

CCashOfficeS - Unique Properties & Methods

<<Class Module>> CCashOfficeS	
	<<Get>> CastObject() : IPOSCollection
	<<Get>> CastObjectChild() : IChildPOSCollection
	<<Get>> CashOfficeMange() : CCashOfficeMange
	<<Get>> IsEdit() : Boolean
	<<Let>> COType(vData : CO_Type)
	<<Get>> COType() : CO_Type
	<<Let>> ID(vData : Long)
	<<Get>> ID() : Long
	<<Let>> PreviousShift(vData : Boolean)
	<<Get>> PreviousShift() : Boolean
	<<Let>> POSMode(vData : Boolean)
	<<Get>> POSMode() : Boolean
	<<Get>> EnabledPrevShiftPeriod() : Boolean
	<<Get>> AllwCshrDclr() : Boolean
	<<Get>> CashierSignOff() : Boolean
	<<Get>> CashierStatus() : Integer
	<<Get>> ShiftExistance(EmpID : Long, bPOSMode : Boolean) : Boolean
	<<Get>> CashierSignOffByID(EmpID : Long) : Boolean
	<<Get>> CashierPosStatusByID(EmpID : Long) : Integer
	<<Get>> POSErrors() : POSErrors
	Item(ByVal Index : Variant) : CCashOffice

Name	Description
CastObjectChild	CastObject is an inherited object, exposing The entire common properties and methods of another business objects linked to the object itself, except of its child\single object.
CashOfficeMan age	Exposing the cCashOfficeManage interface.
CoType	Returns\Sets the CashOffice transaction type. 1 = AddLoan 2 = Pickup 3 = Declaration
ID	Returns\Sets the POS\Cashier ID



Name	Description
PreviousShift	Returns\Sets whether the POS\Cashier has a previous shift. (Relevant for system that defined as “Delayed Cashier\POS balancing” mode. TRUE = Previous shift does exist FALSE = Previous shift does not exist
PosMode	Returns\Sets the balancing mode. TRUE = POS Balancing FALSE = Cashier balancing
EnablePrevious ShiftPeriod	Determines whether the system allow to enable working with the Previous shift, though such period can be found.
AllwCshrDclr	Returns whether the POS\Cashier can be declared.
CashierSignOff	Returns whether the POS\Cashier is in a SignOff mode (According the period)
CashierStatus	Return the POS\Cashier status: 1 – Sign On 2 – Sign Off 9 – Locked (Balanced)
ShiftExistance	Return TRUE whether the POS\Cashier was active.

Object Validations and Error Codes

Number	Description
1080 1	Maximum fields for ordering is 10.
1080 2	Option is not implement.
1080 3	Need to be in Edit Mode.
1080 4	Item Already Exist!



That's the **NEW** Point

CCashOffice - Unique Properties & Methods

<<Class Module>> CCashOffice	
<<Get>> CastObject() : IBusObject	
<<Let>> LoadObject(ByVal vData : Boolean)	
<<Get>> LoadObject() : Boolean	
<<Let>> ID(ByVal IGetID : Long)	
<<Get>> ID() : Long	
<<Let>> MangerID(ByVal vData : Long)	
<<Get>> MangerID() : Long	
<<Let>> COType(ByVal bGetType : Byte)	
<<Get>> COType() : Byte	
<<Let>> TenderID(ByVal iGetTenderID : Integer)	
<<Get>> TenderID() : Integer	
<<Let>> TenderGroup(ByVal iGetTenderID : Integer)	
<<Get>> TenderGroup() : Integer	
<<Let>> AutoDeclr(ByVal vData : Boolean)	
<<Get>> AutoDeclr() : Boolean	
<<Let>> Description(ByVal sGetDescription : String)	
<<Get>> Description() : String	
<<Let>> Quantity(ByVal IGetQuantity : Long)	
<<Get>> Quantity() : Long	
<<Let>> Amount(ByVal dGetAmount : Double)	
<<Get>> Amount() : Double	
<<Let>> TotalQuantity(ByVal IGetTotalQuantity : Long)	
<<Get>> TotalQuantity() : Long	
<<Let>> TotalAmount(ByVal IGetTotalAmount : Double)	
<<Get>> TotalAmount() : Double	
<<Let>> ExpectedAmount(ByVal IGetTotalAmount : Double)	
<<Get>> ExpectedAmount() : Double	
<<Let>> POSMode(ByVal bGetPOSAccountability : Boolean)	
<<Get>> POSMode() : Boolean	
<<Let>> PreviousShift(ByVal bGetPreviousShift : Boolean)	
<<Get>> PreviousShift() : Boolean	
<<Let>> Position(vData : Long)	
<<Get>> Position() : Long	
<<Let>> IsNew(ByVal vData : Boolean)	
<<Get>> IsNew() : Boolean	
<<Let>> IsDeleted(ByVal vData : Boolean)	
<<Get>> IsDeleted() : Boolean	
<<Let>> IsDirty(ByVal vData : Boolean)	
<<Get>> IsDirty() : Boolean	
<<Get>> POSErrors() : V8Interfaces.POSErrors	

Name	Description
------	-------------



Name	Description
LoadObject	Returns whether the object is in LOAD mode.
ID	Returns\Sets the POS\Cashier ID.
TenderID	Returns\Sets the Tender code.
Tender Group	Returns\Sets the Tender group.
AutoDeclr	Returns\Sets whether Automatic Declaration defined.
Description	Returns\Sets the CashOffice transaction's description.
Amount	Returns\Sets the CashOffice transaction's tender-amount.
Quantity	Returns\Sets the CashOffice transaction's tender-quantity.
TotalAmount	Returns\Sets the transaction's total amount (For all the tenders)
TotalQuantity	Returns\Sets the transaction's total quantity (For all the tenders)
ExpectedAmount	Returns\Sets the expected amount, before declaring.

Object Validations and Error Codes










Number	Description
1050 1	String too long.
1050 2	Option is not implement.



That's the **Point**

Number	Description
10503	Amount Limited to 999999999999.99

CViewLoanPickupS - Unique Properties & Methods

<<Class Module>> CViewLoanPickupS	
	<<Let>> ID(vData : Long)
	<<Get>> ID() : Long
	<<Let>> POSMode(vData : Boolean)
	<<Get>> POSMode() : Boolean
	<<Let>> PreviousShift(vData : Boolean)
	<<Get>> PreviousShift() : Boolean
	<<Get>> CastObject() : IPOSCollection
	<<Get>> POSErrors() : POSErrors
	Item(ByVal Index : Variant) : CViewLoanPickup

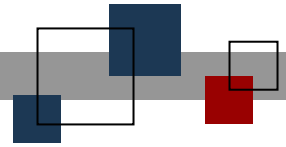
Name	Description
ID	Returns\Sets the POS\Cashier ID
PreviousShift	Returns\Sets whether the POS\Cashier has a previous shift. (Relevant for system that defined as "Delayed Cashier\POS balancing" mode. TRUE = Previous shift does exist FALSE = Previous shift does not exist
PosMode	Returns\Sets the balancing mode. TRUE = POS Balancing FALSE = Cashier balancing



CViewLoanPickup - Unique Properties & Methods

<<Class Module>> CViewLoanPickup	
<<Get>> CastObject() : IBusObject	
<<Let>> LoadObject(ByVal vData : Boolean)	
<<Get>> LoadObject() : Boolean	
<<Let>> TransDate(ByVal dGetTransDate : Date)	
<<Get>> TransDate() : Date	
<<Let>> CashierID(ByVal iGetCashierID : Integer)	
<<Get>> CashierID() : Integer	
<<Let>> PosID(ByVal iGetPosID : Integer)	
<<Get>> PosID() : Integer	
<<Let>> TenderID(ByVal iGetTenderID : Integer)	
<<Get>> TenderID() : Integer	
<<Let>> LoanPickupFG(ByVal bGetLoanPickupFG : Boolean)	
<<Get>> LoanPickupFG() : Boolean	
<<Let>> LoanPickupAmount(ByVal cGetLoanPickupAmount : Currency)	
<<Get>> LoanPickupAmount() : Currency	
<<Let>> LoanPickupQuantity(ByVal lGetLoanPickupQuantity : Long)	
<<Get>> LoanPickupQuantity() : Long	
<<Let>> ManagerID(ByVal iGetManagerID : Integer)	
<<Get>> ManagerID() : Integer	
<<Let>> TransFrom(ByVal bGetTransFrom : Byte)	
<<Get>> TransFrom() : Byte	
<<Let>> CashierShiftNumber(ByVal lGetCashierShiftNumber : Long)	
<<Get>> CashierShiftNumber() : Long	
<<Let>> PosShiftNumber(ByVal lGetPosShiftNumber : Long)	
<<Get>> PosShiftNumber() : Long	
<<Let>> Position(vData : Long)	
<<Get>> Position() : Long	
<<Let>> IsNew(ByVal vData : Boolean)	
<<Get>> IsNew() : Boolean	
<<Let>> IsDeleted(ByVal vData : Boolean)	V8 Business objects
<<Get>> IsDeleted() : Boolean	
<<Let>> IsDirty(ByVal vData : Boolean)	
<<Get>> IsDirty() : Boolean	
<<Get>> POSForm() : V8Interface POSForm	

Name	Description
TransDate	Returns\Sets the transaction date.
CashierID	Returns\Sets the cashier ID.
PosID	Returns\Sets the POS ID.
TenderID	Returns\Sets the Tender code.
LoanPickupFG	Returns\Sets whether it is a Loan or Pickup transaction. True = AddLoan False = Pickup
LoanPickupAmount	Returns\Sets the transaction's tender-amount.
LoanPickupQuantity	Returns\Sets the transaction's tender-quantity.
ManagerID	Returns\Sets the manager ID
CashierShiftNumber	The number of the current period for the specific POS\Cashier.



© StoreNext Retail Technologies LLC 2006

StoreNext Retail Technologies LLC endeavors to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission.

The development of StoreNext products and services is continuous and published information may not be up to date. It is important to check the current position with StoreNext. This document is not part of a contract or license save insofar as may be expressly agreed.