

PRELIMINARY

ISS 45



■ *Active-X Business Objects*
■ *Technical Reference*
■ *Version 8.2*
■
■

ISS45 8.1 Active-X 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

**Copyright® International Computers Limited 1995-2000
All rights reserved.**

This publication is protected by federal copyright law into any human or computer language in any form or by any means, electronic, mechanical, magnetic, manual. No part of this publication may be copied or distributed, stored in a retrieval system, or translated or otherwise, or disclosed to third parties without the express written permission of ICL Retail Systems.

ICL Retail Systems 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. ICL Retail Systems further reserves the right to revise this publication and to make changes from time to time in the contents hereof without obligation of ICL Retail Systems to notify any person or organization of such revision or changes.

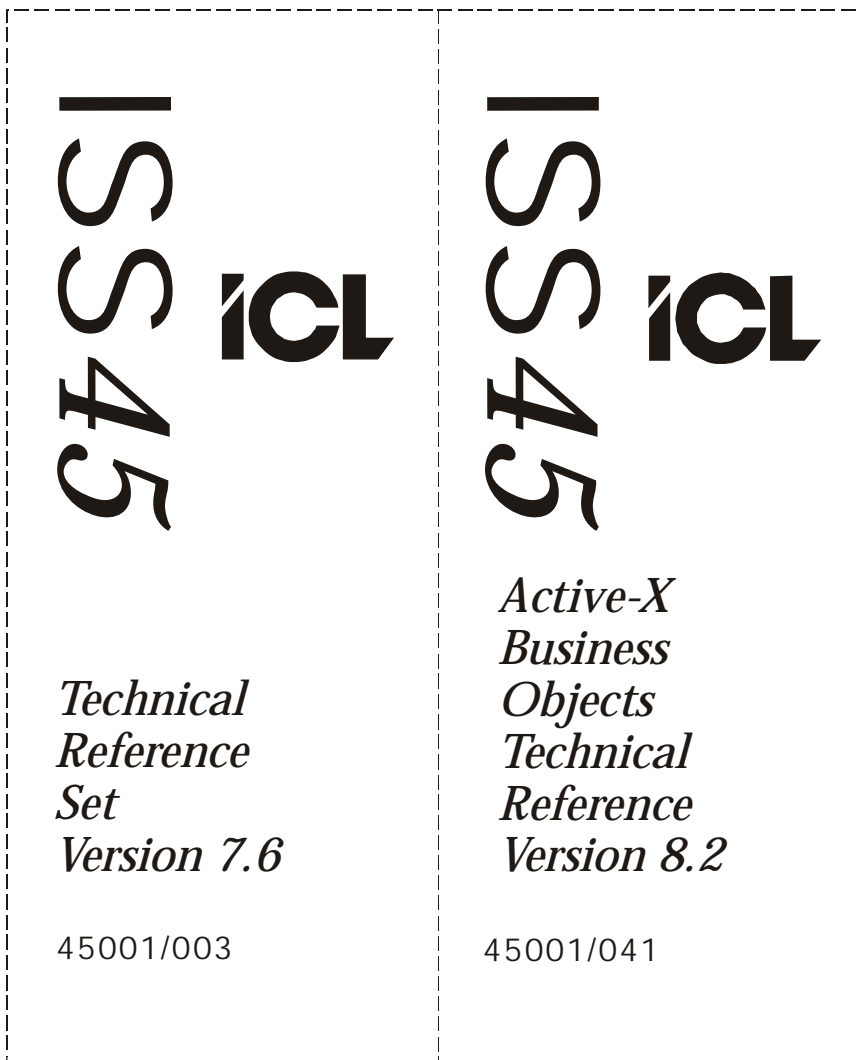
ICL Retail Systems has prepared this manual for use by users, authorized third parties and personnel of ICL Retail Systems as a guide to the proper installation, operation, customization and/or maintenance of ICL Retail Systems equipment and software. The drawings and specifications contained herein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

Address comments and corrections to:

ICL Retail Systems
ISS45 Program Director
2933 Bunker Hill Lane
Suite 101
Santa Clara, CA 95054

This documentation is designed for placement in an ICL binder that can be ordered separately. To order the binder, contact your sales representative. Indicate PIN 45007/002 and/or part number 80192817.

This sheet contains spine cards that can be used to identify the binder for the appropriate documentation. Cut one of the cards along the dotted lines and insert it in the binder's spine pocket. Discard the remaining cards or save them for later use.



PRELIMINARY

ISS 45



- *Active-X Business Objects*
- *Technical Reference*
- *Version 8.1*
-
-
-

ISS45 8.1 Active-X Business Objects Technical Reference

Date of Issue	Product Identification Number	Part Number	Brief Description
June 1999	45001/041	No Part Number	Preliminary

**Copyright[®] International Computers Limited 1995-1999
All rights reserved.**

This publication is protected by federal copyright law into any human or computer language in any form or by any means, electronic, mechanical, magnetic, manual. No part of this publication may be copied or distributed, stored in a retrieval system, or translated or otherwise, or disclosed to third parties without the express written permission of ICL Retail Systems.

ICL Retail Systems 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. ICL Retail Systems further reserves the right to revise this publication and to make changes from time to time in the contents hereof without obligation of ICL Retail Systems to notify any person or organization of such revision or changes.

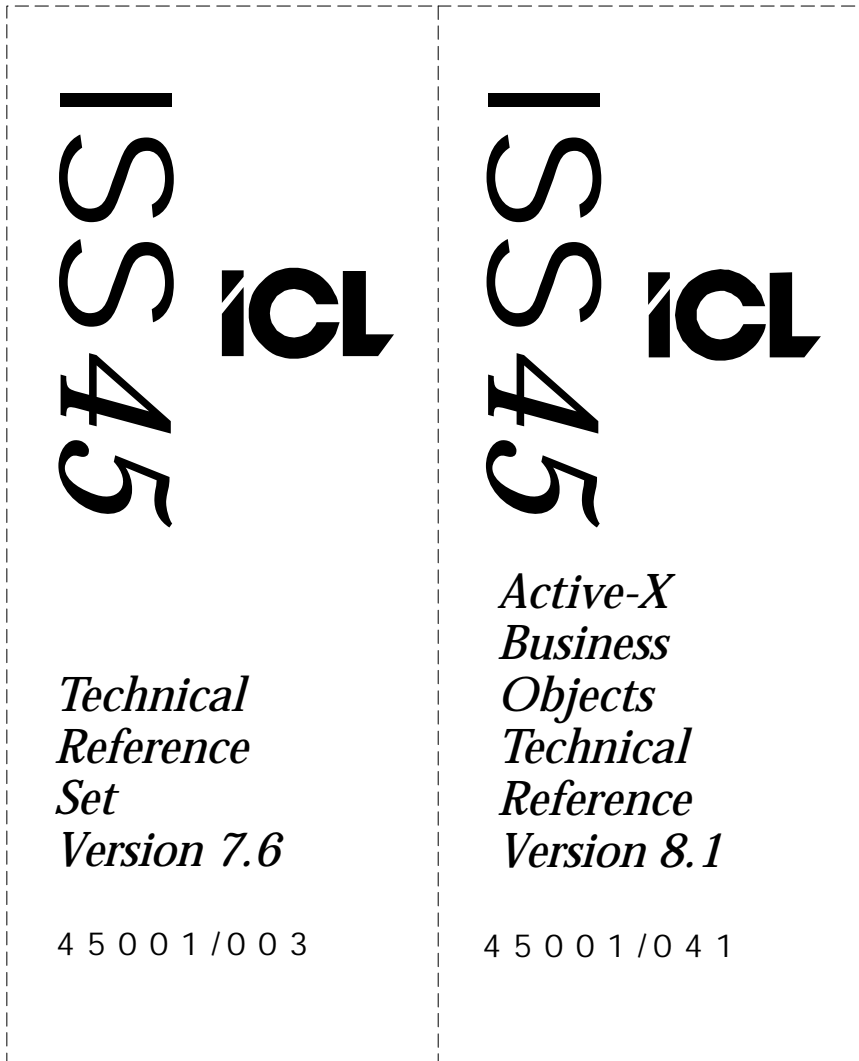
ICL Retail Systems has prepared this manual for use by users, authorized third parties and personnel of ICL Retail Systems as a guide to the proper installation, operation, customization and/or maintenance of ICL Retail Systems equipment and software. The drawings and specifications contained herein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

Address comments and corrections to:

ICL Retail Systems
ISS45 Program Director
2933 Bunker Hill Lane
Suite 101
Santa Clara, CA 95054

This documentation is designed for placement in an ICL binder that can be ordered separately. To order the binder, contact your sales representative. Indicate PIN 45007/002 and/or part number 80192817.

This sheet contains spine cards that can be used to identify the binder for the appropriate documentation. Cut one of the cards along the dotted lines and insert it in the binder's spine pocket. Discard the remaining cards or save them for later use.



Contents

V8 Business Objects.....	1
Overview	1
Object Types.....	1
Programming Concepts	2
Common Methods and Properties.....	2
Project Settings.....	6
Reading a Single Record.....	7
Updating/Inserting Single Record.....	7
Deleting a Single Record	8
Exceptions and Errors	9
Sample Code.....	10
Declare objects	10
Initialize Objects when Loading the Form.....	10
PLU Lookup	11
Applying Changes to Databases.....	12
Destroying Objects.....	12
Setting Object Properties	13
Sample Business Objects.....	14
PLU.....	14

V8 Business Objects

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

Overview

V8 business objects are ActiveX DLLs. 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 usually consists of:

- Reading objects from databases
- Writing objects to databases
- Updating QDX where needed, including sending maintenance to PoS terminals.
- Updating related tables.
- Business rules (validations)

Object Types

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

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.

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.

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)

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.
IsDeleted	Indicates intention to delete the object. This property is set after the Delete() method is called.
IsDirty	Indicates that 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.) 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. 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)

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" 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 first 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 references to the objects you want to use.

VB

Add a reference to the 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 objects, single or multi, according to your needs.

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 a 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
- 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 a 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 operations, like accessing a read-only property, or deleting a non-existent item. Exceptions 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.DepId
    oDeps.CastObject.MoveNext
Next
End sub
```

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  
End Sub
```

Applying Changes to Databases

```
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
```

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 objects.
PriceReasonID	Price change reason ID.
DefaultReturnLoc	Future V9
MessageId	The POS message ID link to the item.
DisplayDescription	Full description of the Item.
SaleRestrict	Pointer to sale restriction group.
ReceiptDescription	The description of the item on ticket.
TaxabilityID	V9 pointer to taxability group.

Name	Description
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
MinimumMerchAmount	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

Name	Description
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.
ShelfLablesRequired	The Item must have shelf label. Future V9
TaxRate1	Connected to tax id 1
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.

Name	Description
SecondFamily	Family code number 2.
SubDepartmentID	Sub department ID.
DiscountID	Discount Id in discount table
LabelQty	Future V9 Label quantity
ScaleFl	Future V9
HeadOfficeDepID	Head Office department ID.
WeightScale	Weight On scale
FreqShopType	Frequent shopper type
FreqShopVal	frequent shopper value
POSMMessage	ID of POS message linked to item.
ShelfLifeDay	Shelf life time in days V9
PromotionID	Promotion ID linked to item.
BucketNumber	NR
ExtendedPromotionID	NR
ExtendedBucketID	NR
RecieptDescription2	Second receipt description for multilingual support
RecieptDescription3	Third receipt description for multilingual support
RecieptDescription4	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.

Name	Description
VendorItemSZ	Vendor item description.
ComparativeUOM	Comparative UOM
BonusBuyOpCode	Bonus buy opcode- always 4.
BonusBuyDescription	Bonus buy dexription
ComparsionType	(0-5) 0 No comparition to be done 1 - Bonus buy saving 2 - Advertised saving 3 - Match advertised saving 4 - Private label saving 5 competitor compare saving.
ComparsionPrice	Comparison price
ComparsionQty	Comparison quantity
ComparsionDate	Comparison date
CompareQty	Compare quantity
CompareUnit	Compare unit
AssumeQuantity	Assume quantity
DEAGroup	DEA group ID
FreqShopLimit	Frequent Shopper limit
PluHaveCheckDigit	PLU is with check digit.
AllowDiscountGreaterThanPrice	Override business rule and allow discount greater than price.
EnhancedFrequentShopper	NR
ItemOnPromotion	Item has active promotion.
ExciseTaxID	Excise tax ID

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.

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 inumber
10005	Weighted item and decimal
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

Number	Description
10014	Discount greater than price
10015	Invalid store vendor
10016	Invalid Frequent shopper Discount
10017	PLU must be linked to subdepartment
10018	Invalid PLU price
10019	Invalid cost per case price
10020	Department 0 not allowed
10021	Bonus Buy description too long
10022	Wrong date
10023	Item does not exist
10024	The excise tax you selected does not exist.
10025	Invalid comparison type
10026	Enhanced Frequent Shopper in the system
10027	Invalid arguments for compressed code
10028	Item on promotion
10029	Item on promotion by department
10030	Item on promotion by mismatch
10031	The coupon you selected does not exist.
10032	The department you selected does not exist.
10033	The sub department you selected does not exist.
10034	The tare weight you selected does not exist.
10035	The discount you selected does not exist.
10036	The return code you selected does not exist.
10037	The pos item message you selected does not exist.
10038	The DEA group you selected does not exist.
10039	The change price reason you selected does not exist.

Number	Description
10040	Item on sale
10041	Active promotion by mixmatch
10042	Active promotion by department

© **International Computers Limited 1997 - 1999**

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

ICL Retail Systems
2933 Bunker Hill Lane, #101
Santa Clara, CA 95054

PIN 45001/041

Contents

CONTENTS	I
V8 BUSINESS OBJECTS	1
<i>Overview.....</i>	<i>1</i>
<i>Object Types.....</i>	<i>1</i>
PROGRAMMING CONCEPTS	2
<i>Common Methods and Properties</i>	<i>2</i>
<i>Project Settings</i>	<i>6</i>
<i>Reading Single Record.....</i>	<i>7</i>
<i>Updating/Inserting Single Record.....</i>	<i>7</i>
<i>Deleting Single Record.....</i>	<i>8</i>
<i>Exceptions and Errors.....</i>	<i>9</i>
SAMPLE CODE	10
<i>Declare objects.....</i>	<i>10</i>
<i>Initialize objects when loading the form</i>	<i>10</i>
<i>PLU lookup</i>	<i>11</i>
<i>Applying changes to database.....</i>	<i>12</i>
<i>Destroying objects.....</i>	<i>12</i>
<i>Setting Object Properties</i>	<i>13</i>
SAMPLE BUSINESS OBJECTS	14
<i>PLU</i>	<i>14</i>
<i>MEMBER PROMOTION.....</i>	<i>24</i>

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)

Object Types

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

- **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.
- **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.

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.
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

Property Name	Description
	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. 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” 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

Name	Description
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\Dll\v8busobj.dll

VC++

- Add the following to your application STDAFX.H:

```
#import "C:\Program Files\POSWare\Office\Components\Dll\V8Intf.dll"
```

```
#import "C:\Program Files\POSWare\Office\Components\Dll\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
- 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
```

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 Sub

Form1.MousePointer = 0
Exit Sub

errr:
    Set oErr = New cErrService
    oErr.ErrorHandle "PLU Demo", Log_Msg, Err
Exit Sub
```

```
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
```

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.
DisplayDescription	Full Description of the Item.
SaleRestrict	Pointer to sale restriction group.
ReceiptDescription	The description of the item on ticket.
TaxabilityID	V9 pointer to taxability group.

Name	Description
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 saled.
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
MinimumMerchAmount	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

Name	Description
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.
ShelfLablesRequired	The Item must have shelf label.-Future V9
TaxRate1	Connected to tax id 1
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.

Name	Description
SecondFamily	Family code number 2.
SubDepartmentID	Sub department Id.
DiscountID	Discount Id in discount table
LabelQty	Future V9 Label quantity
ScaleFl	Future V9
HeadOfficeDepID	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
PromotionID	Promotion ID linked to item.
BucketNumber	NR
ExtendedPromotionID	NR
ExtendedBucketID	NR
RecieptDescription2	Second receipt description for multilingual support
RecieptDescription3	Third receipt description for multilingual support
RecieptDescription4	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.

Name	Description
VendorItemSZ	Vendor item description.
ComparativeUOM	Comparative UOM
BonusBuyOpCode	Bonus buy opcode- always 4.
BonusBuyDescription	Bonus buy dexription
ComparsionType	(0-5) 0 No comparition to be done 1 Bonus buy saving 2 Advertised saving 3 Match advertised saving 4 private label saving 5 competitor compare saving.
ComparsionPrice	Comparison price.
ComparsionQty	Comparison quantity
ComparsionDate	Comparison date
CompareQty	Compare quantity
CompareUnit	Compare unit
AssumeQuantity	Assume quantity
DEAGroup	DEA group ID.
FreqShopLimit	Frequent shopper limit
PluHaveCheckDigit	Plu is with check digit.
AllowDiscountGreaterThan Price	Override business rule and allow discount greater thean 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
LocalDelete	Delete Local flag

Name	Description
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
ExcludeMinPurch	Exclude Min Purch
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(ByVal sNumber) as Boolean	Check Digit Validation
BR_ItemIsUPC2(Optional dGetItemID) as Boolean	Check if an item is UPC2
BR_ItemID(ByVal sGetItemID As String) 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 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	This function receives a Plu number,

Name	Description
dblGetItemID As Double)	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
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

Number	Description
10018	Invalid Plu Price
10019	Invalid Cost Per Case Price
10020	Department 0 not allowed
10021	Bonus Buy Description Too long
10022	Wrong Date
10023	Item Not Exist
10024	The excise tax you selected not exists.
10025	Invalid Comparsion Type
10026	Enhanced Frequent shopper in the
10027	Invalid Arguments for compressed code
10028	Item on promotion
10029	Item on promotion By Department
10030	Item on promotion By MixMatch
10031	The coupon you selected not exists.
10032	The department you selected not exists.
10033	The sub department you selected not exists.
10034	The tare weight you selected not exists.
10035	The discount you selected not exists.
10036	The return code you selected not exists.
10037	The pos item message you selected not exists.
10038	The DEA group you selected not exists.
10039	The change price reasone you selected not exists.
10040	Item On sale
10041	Active Promotion By MixMatch
10042	Active Promotion By Department
10043	Invalid Source item number

Number	Description
10044	Copy to the same Plu number
10045	Invalid Destination item number
10046	Item on Promotion by Manufacture
10047	Item Linked To Grade
10048	Item Must Linked To Grade Department
10049	Fuel in the system flag is off
10050	Negative Entry flag set
10051	Coupon flag set
10052	Prohibit changes to Negative Entry flag
10053	Prohibit Manual Changes To Non Merchandise Flag
10054	The department you selected is not active
10055	Invalid Internal code check digit
10056	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.
CrDpndncyFg	Member card required.
AllSchemAvail	Promotion available for all card scheme.
StrHierId	Mark down department. A department that takes the reward charge.
RwrDTyp	Reward Type. 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-

Name	Description
	Target message. 13-Percent of order. 14-Amount of order. 15-Free item.
RwrVal	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
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.

Name	Description
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
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
21410	Limit Quantity Should be between 0 to 429000
21411	Threshold Quantity Should be between 0 to 999999
21412	Threshold Weight Should be between 0 to 999.999
21413	Threshold Value Should be between 0 to 9999.99
21414	Invalid Reward value
21415	You should choose at least one group.
21416	Invalid Minimum Item Value

Number	Description
21417	Invalid Minimum Weight
21418	Invalid Minimum Purchase
21420	Invalid Linked Promotion
21421	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
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
21705	This Mix And Match does not exist
21706	Group Id should be between 1 to 3

© International Computers Limited 1997 - 2000

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

ICL Retail Systems
2933 Bunker Hill Lane, #101
Santa Clara, CA 95054

P/N 89000014
PIN 45001/041