**QBooksLib Library**

This Library is a sample of a .Net library to interface with QuickBooks Desktop to interact with the following objects:

* InventoryItems
* Non-InventoryItems
* Accounts
* ItemServices
* PriceLevels
* Vendors
* Customers
* ShipMethods
* Invoices
* CreditMemos
* ReceivePayments
* Bills
* BillPaymentChecks
* PurchaseOrders
* Checks
* Deposits
* SalesOrders
* Charges
* CreditCardCharges
* CreditCardCredits
* DataExtensions

The library is programmed in C# and use QuickBooks Foundation Class (QBFC) to access QuickBooks.

The QBFC used in this version is 16.0 64bit or 32bit. With this versions you can access up to QuickBooks Desktop 23.0

The following is a summary of the component classes

The fields names in each class are those used by QuickBooks as it descripted in [QuickBooks Desktop API Reference](https://developer.intuit.com/app/developer/qbdesktop/docs/api-reference/qbdesktop) . There are several field names that have the suffix Ret, Add or Mod according what type of request is used, in this cases this suffix is deleted from the field name. For instance you have in the Invoice query in the API Reference fields named:

OrInvoiceLine**Ret**List (Invoice Query Response).

OrInvoiceLine**Add**List (Invoice Add Request).

OrInvoiceLine**Mod**List (Invoice Modify Request).

In this case the field name declared in QBInvoice is **OrInvoiceLineList.**

**Library Usage**

To use this library you need Quickbooks installed in your PC and QuickBooks Desktop SDK version 16.0 32bit or version 16.0 64bits. You can dowload it from [here](https://developer.intuit.com/app/developer/qbdesktop/docs/get-started/download-and-install-the-sdk) (you need an account in [developer.inituit.com](https://accounts.intuit.com/signup.html?offering_id=Intuit.devx.devx&iux_target_aal=25&iux_sso_mfa=true&redirect_url=https%3A%2F%2Fdeveloper.intuit.com%2Fapp%2Fdeveloper%2Fqbdesktop%2Fdocs%2Fget-started%2Fdownload-and-install-the-sdk%3FdevXlogin%3Dtrue))

**QBooksLib Classes**

**QBConnection Class**

This class has the purposes to open and close the connection with QuickBooks. Normally the open/close connection is handled by the methods described below, but if you want to open a connection and then do the different calls to QuickBooks, you can use the QBConnection class. There is one caveat : if you keep your connection open and your program crash QuickBooks could stuck in your memory and also it could prevent to close it (if you have the UI running) or open a new one connection. To overcome this issue you have to kill the QuickBooks process from the task manager.

class QBConnection

{

IMsgSetRequest RequestMsgSet;

QBSessionManager SessionManager;

string FileName;

bool Status;

string ErrorMsg;

}

RequestMsgSet: Attributes for the request set

SessionManager: The session manager

FileName: File name of the open company.

Status: true if the company file is open successful, otherwise is false.

ErrorMsg: Error message.

**Method Constructor**

Open the connection with QuickBooks

public QBConnection(string CompanyName, , QBENConnectionType ConnectionType = QBENConnectionType.ctLocalQBD, QBENOpenMode OpenMode = QBENOpenMode.omDontCare, QBENRqOnError RqOnError = QBENRqOnError.roeContinue )

**Parameters**

**CompanyName**: Complete route of QuickBooks company file. if CompanyName = “” the company open in

QuickBooks UI is used.

**ConnectionType:** Specify a value of ctUnknown, ctLocalQBD, ctRemoteQBD, or ctLocalQBDLaunchUI

**OpenMode:** The desired access mode. It can be one of three values: omSingleUser (specifies

single-user mode) omMultiUser (specifies multi-user mode) omDontCare (accept whatever

mode is currently in effect, or singleuser mode if no other mode is in effect)

**RqOnError:** Attribute which specifies how to proceed when an error occurs.Could it be one of the

following: roeStop, roeContinue or roeRollback.

**Returns**

**QBConnection object**

**Usage:**

QBConnection QBConn = new QBConnection(@“c:\quickbooksFiles\QB.qbw”);

If(QBConn.status)

Console.WriteLine(“QuickBooks File successfully Open!”);

Qbconn.CloseConnection();

**Method CloseConnection**

Close the connection

public CloseConnection();

**Method Version**

Return a double with the version number of the SessionManager.

Public double Version();

**QBError Class**

This class is used to receive the status and error message when you use the Read, Save and Modify methods of classes QBItemInventory**s ,** QBItemNoInventory**s**, QBCustomer**s**, QBInvoice**s**.

**Properties**

**Status**: Return true if ther are no errors.

**Message**: If Status is false return the error message.

**Code**: If Status is false return the error code. Error codes 200 and up are the

errors code returned by QBFC.

**XmlRequest**: If Debug parameter is true return the XML string of the request.

**XmlResponse**: If Debug parameter is true return the XML string of the response.

**QBItemInventory Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBItemInventory

{

public string ListID

public DateTime?

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public string Name

public string FullName

public string BarCodeValue

public QBBarCode BarCode

public bool? IsActive

public QBRef ClassRef

public QBRef ParentRef

public int? Sublevel

public string ManufacturerPartNumber

public QBRef UnitOfMeasureSetRef

public bool? IsTaxIncluded

public QBRef SalesTaxCodeRef

public string SalesDesc

public double? SalesPrice

public QBRef IncomeAccountRef

public string PurchaseDesc

public double? PurchaseCost

public QBRef PurchaseTaxCodeRef

public QBRef COGSAccountRef

public QBRef PrefVendorRef

public QBRef AssetAccountRef

public double? ReorderPoint

public double? Max

public double? QuantityOnHand

public double? AverageCost

public double? QuantityOnOrder

public double? QuantityOnSalesOrder

public string ExternalGUID

public List<QBDataExt> DataExtList

public DateTime? InventoryDate

public double? TotalValue

public bool? ForceUOMChange

public bool? ApplyIncomeAccountRefToExistingTxns

}

**QBItemInventorys Class**

This class is used to read, save and modify the inventory items . Inherit a list of QBItemInventory objects.

**Method Read**

This method gets a list of the inventory items from QuickBooks.

**Overlays**

public QBError Read(string CompanyName, List<string> ListIDList = null, List<string> FullNameList = null, bool All = false, bool Debug = false);

public QBError Read(QBConnection QBConn, List<string> ListIDList = null, List<string> FullNameList = null, bool All = false, bool Debug = false);

**Parameters**

**CompanyName**: Complete route of QuickBooks company file. if CompanyName = “” the company

open in QuickBooks UI is used.

**ListIDList**: List of ListID of items to be read.

**FullNameList**: List of Full Name of items to be read.

**All**: If it is true this method read all Items in the company file. (Use with

caution, if the items number is too big this method can take several

minutes to complete)

**Debug**: If it is true QBError fields : XmlRequest and XmlResponse returns the xml

request and response of the query.

**IncludeRetElementList:** List of fields included in the query.

When you read the items **ListIDList** has precedence over **FullNameList** and the last one has precedence over **All.**

**Returns**

**Qberror object.** If Qberror.Status is true the read was successful if not QBError.ErrorMsg return a string with error description**.**

**Usage**

QBItemInventorys Items = new QBItemInventorys();

If(Items.Read((@“c:\quickbooksFiles\QB.qbw”).Status)

Console.WriteLine(“Items Read successfully”);

**Method Save**

Add items to QuickBooks Item List

**Overlays**

public QBError Save(string CompanyName, bool Debug = false)

public QBError Save(QBConnection Connection, bool Debug = false)

**Parameters**

**CompanyName**: Complete route of QuickBooks company file. if CompanyName = “” the company

open in QuickBooks UI is used.

**Debug**: If it is true QBError fields : XmlRequest and XmlResponse returns the xml

request and response of the query.

**Returns**

**Qberror object.** If Qberror.Status is true the read was successful if not QBError.ErrorMsg return a string with error description**.**

**Usage**

QBItemInventory Item = new QBItemInventory()

//Setup all the required fields of Item

//i.e. Item.Name = “NN”;

QBItemInventorys Items = new QBItemInventorys();

Items.Add(Item);

If(Items.Save((@“c:\quickbooksFiles\QB.qbw”).Status)

Console.WriteLine(“Items Add successfully”);

**Method Modify**

Add items to QuickBooks Item List

**Overlays**

public QBError Save(string CompanyName, bool Debug = false)

public QBError Save(QBConnection Connection, bool Debug = false)

**Parameters**

**CompanyName**: Complete route of QuickBooks company file. if CompanyName = “” the company

open in QuickBooks UI is used.

**Debug**: If it is true the field name XmlRequest and XmlResponse returns the xml

request and response of the query.

**Returns**

**Qberror object.** If Qberror.Status is true the read was successful if not QBError.ErrorMsg return a string with error description**.**

**Usage**

QBItemInventorys Items = new QBItemInventorys();

Item[0].Price = 10.0; //modify the price of item number 0.

If(Items.Save((@“c:\quickbooksFiles\QB.qbw”).Status)

Console.WriteLine(“Items Saved successfully”);

**QBItemNonInventory**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBItemNonInventory {

public DateTime? TimeModified

public string EditSequence

public string Name

public string FullName

public string BarCodeValue

public bool? IsActive

public QBRef ClassRef

public QBRef ParentRef

public int? Sublevel

public string ManufacturerPartNumber

public QBRef UnitOfMeasureSetRef

public bool? IsTaxIncluded

public QBRef SalesTaxCodeRef

public QBORSalesPurchase ORSalesPurchase

public string ExternalGUID

public List<QBDataExt> DataExtList

public QBBarCode BarCode

public bool? ForceUOMChange

}

**QBItemNonInventorys Class**

This class is used to read, save and modify the ItemNonInventory. Inherit a list of QBItemNonInventory objects. Properties, methods an example similar to QBItemInventory.

**QBAccount**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBItemNonInventory {

public string ListID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public string Name

public string FullName

public bool? IsActive

public QBRef ParentRef

public int? Sublevel

public QBENAccountType? AccountType

public QBENSpecialAccountType? SpecialAccountType

public bool? IsTaxAccount

public string AccountNumber

public string BankNumber

public string Desc

public double? Balance

public double? TotalBalance

public QBRef SalesTaxCodeRef

public QBTaxLineInfo TaxLineInfoRet

public QBENCashFlowClassification? CashFlowClassification

public QBRef CurrencyRef

public List<QBDataExt> DataExtList

public double? OpenBalance

public DateTime? OpenBalanceDate

public int? TaxLineID

}

**QBAccounts Class**

This class is used to read, save and modify the Accounts . Inherit a list of QBAccount objects.

Properties, methods an example similar to QBItemInventory.

**QBItemService**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBItemService{

public string ListID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public string Name

public string FullName

public string BarCodeValue

public QBBarCode BarCode

public bool? IsActive

public QBRef ClassRef

public QBRef ParentRef

public int? Sublevel

public QBRef UnitOfMeasureSetRef

public bool? ForceUOMChange

public bool? IsTaxIncluded

public QBRef SalesTaxCodeRef

public QBORSalesPurchase ORSalesPurchase

public string ExternalGUID

public List<QBDataExt> DataExtList

}

**QBItemServices Class**

This class is used to read, save and modify the ItemServices . Inherit a list of QBItemService objects.

Properties, methods an example similar to QBItemInventory.

**QBPriceLevel**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBPriceLevel{

public string ListID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public string Name

public bool? IsActive

public QBENPriceLevelType? PriceLevelType

public QBORPriceLevel ORPriceLevel

public QBRef CurrencyRef

}

**QB****PriceLevels Class**

This class is used to read, save and modify the PriceLevels. Inherit a list of QBPriceLevel objects.

Properties, methods an example similar to QBItemInventory.

**QBVendor**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBVendor{

public string ListID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public string Name

public bool? IsActive

public QBRef ClassRef

public bool? IsTaxAgency

public string CompanyName

public string Salutation

public string FirstName

public string MiddleName

public string LastName

public string JobTitle

public QBAddress VendorAddress

public QBAddressBlock VendorAddressBlock

public QBAddress ShipAddress

public string Phone

public string AltPhone

public string Fax

public string Email

public string Cc

public string Contact

public string AltContact

public List<QBAdditionalContact> AdditionalContactRefList

public List<QBContact> ContactsList

public string NameOnCheck

public string AccountNumber

public string Notes

public List<QBAdditionalNote> AdditionalNotesList

public QBRef VendorTypeRef

public QBRef TermsRef

public double? CreditLimit

public string VendorTaxIdent

public bool? IsVendorEligibleFor1099

public double? Balance

public QBRef BillingRateRef

public string ExternalGUID

public QBRef SalesTaxCodeRef

public QBENSalesTaxCountry? SalesTaxCountry

public bool? IsSalesTaxAgency

public QBRef SalesTaxReturnRef

public string TaxRegistrationNumber

public QBENReportingPeriod? ReportingPeriod

public bool? IsTaxTrackedOnPurchases

public QBRef TaxOnPurchasesAccountRef

public bool? IsTaxTrackedOnSales

public QBRef TaxOnSalesAccountRef

public bool? IsTaxOnTax

public List<QBRef> PrefillAccountRefList

public QBRef CurrencyRef

public List<QBDataExt> DataExtList

public double? OpenBalance

public DateTime? OpenBalanceDate

}

**QBVendors Class**

This class is used to read, save and modify the Vendors. Inherit a list of QBVendor objects.

Properties, methods an example similar to QBItemInventory.

**QBCustomer**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBCustomer {

public string ListID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public string Name

public string FullName

public bool? IsActive

public QBRef ClassRef

public QBRef ParentRef

public int? Sublevel

public string CompanyName

public string Salutation

public string FirstName

public string MiddleName

public string LastName

public string JobTitle

public QBAddress BillAddress

public QBAddressBlock BillAddressBlock

public QBAddress ShipAddress

public QBAddressBlock ShipAddressBlock

public List<QBAddress> ShipToAddressList

public string Phone

public string AltPhone

public string Fax

public string Email

public string Cc

public string Contact

public string AltContact

public List<QBAdditionalContact> AdditionalContactRefList

public List<QBContact> ContactsList

public QBRef CustomerTypeRef

public QBRef TermsRef

public QBRef SalesRepRef

public double? Balance

public double? TotalBalance

public QBRef SalesTaxCodeRef

public QBRef ItemSalesTaxRef

public QBENSalesTaxCountry? SalesTaxCountry

public string ResaleNumber

public string AccountNumber

public double? CreditLimit

public QBRef PreferredPaymentMethodRef

public QBCreditCardInfo CreditCardInfo

public QBRef JobTypeRef

public string Notes

public List<QBAdditionalNote> AdditionalNotesList

public QBENPreferredDeliveryMethod? PreferredDeliveryMethod

public QBRef PriceLevelRef

public string ExternalGUID

public string TaxRegistrationNumber

public QBRef CurrencyRef

public List<QBDataExt> DataExtList

public double? OpenBalance

public DateTime? OpenBalanceDate

public List<string> IncludeRetElementList

public QBENJobStatus? JobStatus

public DateTime? JobStartDate

public DateTime? JobProjectedEndDate

public DateTime? JobEndDate

public string JobDesc

}

**QBCustomers Class**

This class is used to read, save and modify the Customers. Inherit a list of QBCustomer objects.

Properties, methods an example similar to QBItemInventory.

**QBShipMethod**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBShipMethod{

public string ListID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public string Name

public bool? IsActive

}

**QB****ShipMethods Class**

This class is used to read, save and modify the ShipMethods. Inherit a list of QBShipMethod objects.

Properties, methods an example similar to QBItemInventory only that QBShipMethods class doesn’t have Modify Method.

**QBInvoice Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBInvoice {

public string TxnID;

public DateTime? TimeCreated;

public DateTime? TimeModified;

public string EditSequence;

public int? TxnNumber;

public QBRef CustomerRef;

public QBRef ClassRef;

public QBRef ARAccountRef;

public QBRef TemplateRef;

public DateTime? TxnDate;

public string RefNumber;

public QBAddress BillAddress;

public QBAddressBlock BillAddressBlock;

public QBAddress ShipAddress;

public QBAddressBlock ShipAddressBlock;

public bool? IsPending;

public bool? IsFinanceCharge;

public string PONumber;

public QBRef TermsRef;

public DateTime? DueDate;

public QBRef SalesRepRef;

public string FOB;

public DateTime? ShipDate;

public QBRef ShipMethodRef;

public double? Subtotal;

public QBRef ItemSalesTaxRef;

public double? SalesTaxPercentage;

public double? SalesTaxTotal;

public double? AppliedAmount;

public double? BalanceRemaining;

public QBRef CurrencyRef;

public double? ExchangeRate;

public double? BalanceRemainingInHomeCurrency;

public string Memo;

public bool? IsPaid;

public QBRef CustomerMsgRef;

public bool? IsToBePrinted;

public bool? IsToBeEmailed;

public bool? IsTaxIncluded;

public QBRef CustomerSalesTaxCodeRef;

public double? SuggestedDiscountAmount;

public DateTime? SuggestedDiscountDate;

public string Other;

public string ExternalGUID;

public List<QBLinkedTxn> LinkedTxnList;

public List<QBORInvoiceLine> ORInvoiceLineList;

public List<QBDataExt> DataExtList;

public List<QBSetCredit> SetCreditList;}

**QBInvoices Class**

This class is used to read, save and modify the Invoice . Inherit a list of QBInvoice objects.

**Properties**

string XmlRequest: the XML string of the query used, used for debugging purposes when the parameter

Invoice is true.

string XmlResponse: the XML string with the response of the query used, used for debugging purposes when

the parameter Invoice is true.

**Method Read**

This method gets a list of Invoices from QuickBooks.

**Overlays**

public QBError Read(string CompanyName, DateTime? FromDate = null, DateTime? ToDate = null, List<string> TxnIDList = null, List<string> RefNumberList = null, string Entity = "", bool All = false, bool debug = false)

public QBError Read(QBConnection QBConn, DateTime? FromDate = null, DateTime? ToDate = null, List<string> TxnIDList = null, List<string> RefNumberList = null, string Entity = "", bool All = false, bool debug = false)

**Parameters**

**CompanyName**: Complete route of QuickBooks company file. if CompanyName = “” the company

open in QuickBooks UI is used.

**FromDate:** Start date of the search.

**ToDate:** End Date of the search.

**TxnIDList**: List of TxnID of invoices to be read.

**RefNumberList**: List of RefNumber of Invoices to be read.

**Entity:** List of Entity FullNames (in this case Customers Name).

**All**: If it is true this method read all Invoices in the company file. (Use with

caution, if the number of Invoices is too big, this method can take

several minutes to complete)

**Debug**: If it is true QBError fields : XmlRequest and XmlResponse returns the xml

request and response of the query.

When you read the invoices **FromDate and ToDate** has precedence over **TxnIDList,** and this one has precedence over **RefNUmberList,** and this last one has precedence over **All.**

**Returns**

**Qberror object.** If Qberror.Status is true the read was successful if not QBError.ErroMsg return a string with error description**.**

**Usage**

QBInvoices Invoices = new QBInvoices();

If(Invoices.Read((@“c:\quickbooksFiles\QB.qbw”).Status)

Console.WriteLine(“Invoices Read successfully”);

**Method Save**

Add an Invoice to QuickBooks Invoices List

**Overlays**

public QBError Save(string CompanyName, bool Debug = false)

public QBError Save(QBConnection Connection, bool Debug = false)

**Parameters**

**CompanyName**: Complete route of QuickBooks company file. if CompanyName = “” the company

open in QuickBooks UI is used.

**Debug**: If it is true QBError fields : XmlRequest and XmlResponse returns the xml

request and response of the query.

**Returns**

**Qberror object.** If Qberror.Status is true the read was successful if not QBError.ErrorMsg return a string with error description**.**

**Usage**

QBInvoice Inv = new QBInvoice()

QBInvoices Invs = new QBInvoices();

//Setup the required fields of Inv

//i.e. Inv.CustomerRef.FullName = “Joe”;

Invs.Add(Inv);

If(Invs.Save((@“c:\quickbooksFiles\QB.qbw”).Status)

Console.WriteLine(“Invoices Add successfully”);

**Method Modify**

Modify Invoices of QuickBooks Invoices List

**Overlays**

public QBError Save(string CompanyName, bool Debug = false)

public QBError Save(QBConnection Connection, bool Debug = false)

**Parameters**

**CompanyName**: Complete route of QuickBooks company file. if CompanyName = “” the company

open in QuickBooks UI is used.

**Debug**: If it is true QBError fields : XmlRequest and XmlResponse returns the xml

request and response of the query.

**Returns**

**Qberror object.** If Qberror.Status is true the read was successful if not QBError.ErrorMsg return a string with error description**.**

**Usage**

QBInvoices Invs = new QBInvoices();

Invs[0].RefNumber = “10”; //modify the RefNUmber of Invoice number 0.

If(Invs.Save((@“c:\quickbooksFiles\QB.qbw”).Status)

Console.WriteLine(“Invoices Modified successfully”);

**QBCreditMemo Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBCreditMemo {

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef CustomerRef

public QBRef ClassRef

public QBRef ARAccountRef

public QBRef TemplateRef

public DateTime? TxnDate

public string RefNumber

public QBAddress BillAddress

public QBAddressBlock BillAddressBlock

public QBAddress ShipAddress

public QBAddressBlock ShipAddressBlock

public bool? IsPending

public string PONumber

public QBRef TermsRef

public DateTime? DueDate

public QBRef SalesRepRef

public string FOB

public DateTime? ShipDate

public QBRef ShipMethodRef

public double? Subtotal

public QBRef ItemSalesTaxRef

public double? SalesTaxPercentage

public double? SalesTaxTotal

public double? TotalAmount

public double? CreditRemaining

public QBRef CurrencyRef

public float? ExchangeRate

public double? CreditRemainingInHomeCurrency

public string Memo

public QBRef CustomerMsgRef

public bool? IsToBePrinted

public bool? IsToBeEmailed

public bool? IsTaxIncluded

public QBRef CustomerSalesTaxCodeRef

public string Other

public string ExternalGUID

public List<QBLinkedTxn> LinkedTxnList

public List<QBORCreditMemoLine> ORCreditMemoLineList

public List<QBDataExt> DataExtList

}

**QBCreditMemos Class**

This class is used to read, save and modify the CreditMemo. Inherit a list of QBCreditMemo objects.

Properties, methods an example similar to QBInvoices.

**QBReceivePayment Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBReceivePayment {

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef CustomerRef

public QBRef ARAccountRef

public DateTime? TxnDate

public string RefNumber

public double? TotalAmount

public QBRef CurrencyRef

public float? ExchangeRate

public double? TotalAmountInHomeCurrency

public QBRef PaymentMethodRef

public string Memo

public QBRef DepositToAccountRef

public QBCreditCardTxnInfo CreditCardTxnInfo

public List<QBAppliedToTxn> AppliedToTxnList

public List<QBDataExt> DataExtList

public double? UnusedPayment

public double? UnusedCredits

public string ExternalGUID

public QBORApplyPayment ORApplyPayment

}

**QBReceivePayments Class**

This class is used to read, save and modify the ReceivePayments. Inherit a list of QBReceivePayment objects.

Properties, methods an example similar to QBInvoices.

**QBBillClass**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBBill {

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef VendorRef

public QBAddress VendorAddress

public QBRef APAccountRef

public DateTime? TxnDate

public DateTime? DueDate

public double? AmountDue

public QBRef CurrencyRef

public float? ExchangeRate

public double? AmountDueInHomeCurrency

public string RefNumber

public QBRef TermsRef

public string Memo

public bool? IsTaxIncluded

public QBRef SalesTaxCodeRef

public bool? IsPaid

public string ExternalGUID

public List<QBLinkedTxn> LinkedTxnList

public bool? ClearExpenseLines

public bool? ClearItemLines

public List<QBExpenseLine> ExpenseLineList

public List<QBORItemLine> ORItemLineList

public double? OpenAmount

public List<QBDataExt> DataExtList

}

**QBBills Class**

This class is used to read, save and modify the Bills. Inherit a list of QBBill objects.

Properties, methods an example similar to QBInvoices.

**QBBillPaymentCheck Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBBillPaymentCheck {

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef PayeeEntityRef

public QBRef APAccountRef

public DateTime? TxnDate

public QBRef BankAccountRef

public double? Amount

public QBRef CurrencyRef

public float? ExchangeRate

public double? AmountInHomeCurrency

public string RefNumber

public string Memo

public QBAddress Address

public QBAddressBlock AddressBlock

public bool? IsToBePrinted

public string ExternalGUID

public List<QBAppliedToTxn> AppliedToTxnList

public List<QBDataExt> DataExtList

public QBORCheckPrint ORCheckPrint

}

**QBBillPaymentChecks Class**

This class is used to read, save and modify the BillPaymentChecks. Inherit a list of QBBillPaymentCheck objects . Properties, methods an example similar to QBInvoices.

**QBPurchaseOrder Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBPurchaseOrder {

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef VendorRef

public QBRef ClassRef

public QBRef InventorySiteRef

public QBRef ShipToEntityRef

public QBRef TemplateRef

public DateTime? TxnDate

public string RefNumber

public QBAddress VendorAddress

public QBAddressBlock VendorAddressBlock

public QBAddress ShipAddress

public QBAddressBlock ShipAddressBlock

public QBRef TermsRef

public DateTime? DueDate

public DateTime? ExpectedDate

public QBRef ShipMethodRef

public string FOB

public double? TotalAmount

public QBRef CurrencyRef

public double? ExchangeRate

public double? TotalAmountInHomeCurrency

public bool? IsManuallyClosed

public bool? IsFullyReceived

public string Memo

public string VendorMsg

public bool? IsToBePrinted

public bool? IsToBeEmailed

public bool? IsTaxIncluded

public QBRef SalesTaxCodeRef

public string Other1

public string Other2

public string ExternalGUID

public List<QBLinkedTxn> LinkedTxnList

public List<QBORPurchaseOrderLine> ORPurchaseOrderLineList

public List<QBDataExt> DataExtList

public QBORInventorySiteORShipToEntity ORInventorySiteORShipToEntity

}

**QBPurchaseOrders Class**

This class is used to read, save and modify the PurchaseOrders. Inherit a list of QBPurchaseOrder objects . Properties, methods an example similar to QBInvoices.

**QBCheck Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBCheck {

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef AccountRef

public QBRef PayeeEntityRef

public string RefNumber

public DateTime? TxnDate

public double? Amount

public QBRef CurrencyRef

public double? ExchangeRate

public double? AmountInHomeCurrency

public string Memo

public QBAddress Address

public QBAddressBlock AddressBlock

public bool? IsPending

public bool? IsToBePrinted

public bool? IsTaxIncluded

public QBRef SalesTaxCodeRef

public string ExternalGUID

public List<QBLinkedTxn> LinkedTxnList

public List<QBExpenseLine> ExpenseLineList

public List<QBORItemLine> ORItemLineList

public List<QBDataExt> DataExtList

public List<QBAppliedToTxn> ApplyCheckToTxnList;

public bool? ClearItemLines { get; set; }

public bool? ClearExpenseLines { get; set; }

}

**QBChecks Class**

This class is used to read, save and modify the Checks. Inherit a list of QBCheck objects . Properties, methods an example similar to QBInvoices.

**QBDeposit Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBDeposit {

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public DateTime? TxnDate

public QBRef DepositToAccountRef

public string Memo

public double? DepositTotal

public QBRef CurrencyRef

public float? ExchangeRate

public double? DepositTotalInHomeCurrency

public QBCashInfo CashBackInfo

public string ExternalGUID

public List<QBDepositLine> DepositLineList

public List<QBDataExt> DataExtList

}

**QBDeposits Class**

This class is used to read, save and modify the Deposits. Inherit a list of QBDeposit objects . Properties, methods an example similar to QBInvoices.

**QBSalesOrder Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBSalesOrder

{

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef CustomerRef

public QBRef ClassRef

public QBRef TemplateRef

public DateTime? TxnDate

public string RefNumber

public QBAddress BillAddress

public QBAddressBlock BillAddressBlock

public QBAddress ShipAddress

public QBAddressBlock ShipAddressBlock

public string PONumber

public QBRef TermsRef

public DateTime? DueDate

public QBRef SalesRepRef

public string FOB

public DateTime? ShipDate

public QBRef ShipMethodRef

public double? Subtotal

public QBRef ItemSalesTaxRef

public double? SalesTaxPercentage

public double? SalesTaxTotal

public double? TotalAmount

public QBRef CurrencyRef

public float? ExchangeRate

public double? TotalAmountInHomeCurrency

public bool? IsManuallyClosed

public bool? IsFullyInvoiced

public string Memo

public QBRef CustomerMsgRef

public bool? IsToBePrinted

public bool? IsToBeEmailed

public bool? IsTaxIncluded

public QBRef CustomerSalesTaxCodeRef

public string Other

public string ExternalGUID

public List<QBLinkedTxn> LinkedTxnList

public List<QBORSalesOrderLine> ORSalesOrderLineList

public List<QBDataExt> DataExtList

public string FulfillmentStatus

public string ShippingDetails

public QBShippingDetailsLine ShippingDetailsLineRet

public QBENSOChannel? SOChannel

public string StoreName

public string StoreType

}

**QBSalesOrders Class**

This class is used to read, save and modify the SalesOrders. Inherit a list of QBSalesOrder objects . Properties, methods an example similar to QBInvoices.

**QBCharge Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBCharge

{

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef CustomerRef

public DateTime? TxnDate

public string RefNumber

public QBRef ItemRef

public QBRef InventorySiteRef

public QBRef InventorySiteLocationRef

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public double? Rate

public double? Amount

public double? BalanceRemaining

public string Desc

public QBRef ARAccountRef

public QBRef ClassRef

public DateTime? BilledDate

public DateTime? DueDate

public bool? IsPaid

public string ExternalGUID

public List<QBLinkedTxn> LinkedTxnList

public List<QBDataExt> DataExtList

public QBRef OverrideItemAccountRef

public QBENOptionForPriceRuleConflict? OptionForPriceRuleConflict

}

**QBCharges Class**

This class is used to read, save and modify the Charges. Inherit a list of QBCharge objects . Properties, methods an example similar to QBInvoices.

**QBCreditCardCredit Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBCreditCardCredit

{

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef AccountRef

public QBRef PayeeEntityRef

public DateTime? TxnDate

public double? Amount

public QBRef CurrencyRef

public float? ExchangeRate

public double? AmountInHomeCurrency

public string RefNumber

public string Memo

public bool? IsTaxIncluded

public QBRef SalesTaxCodeRef

public string ExternalGUID

public List<QBExpenseLine> ExpenseLineList

public List<QBORItemLine> ORItemLineList

public List<QBDataExt> DataExtList

public bool? ClearExpenseLines

}

**QBCreditCardCredits Class**

This class is used to read, save and modify the CreditCardCredits. Inherit a list of QBCreditCardCredit objects . Properties, methods an example similar to QBInvoices.

**QBCreditCardCharge Class**

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBCreditCardCharge

{

public string TxnID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public int? TxnNumber

public QBRef AccountRef

public QBRef PayeeEntityRef

public DateTime? TxnDate

public double? Amount

public QBRef CurrencyRef

public float? ExchangeRate

public double? AmountInHomeCurrency

public string RefNumber

public string Memo

public bool? IsTaxIncluded

public QBRef SalesTaxCodeRef

public string ExternalGUID

public List<QBExpenseLine> ExpenseLineList

public List<QBORItemLine> ORItemLineList

public List<QBDataExt> DataExtList

public bool? ClearExpenseLines

public bool ClearItemLines

}

**QBCreditCardCharges Class**

This class is used to read, save and modify the CreditCardCharges. Inherit a list of QBCreditCardCharge objects . Properties, methods an example similar to QBInvoices.

**QBDataExtADM Class**

This Class is for Add, Delete and Modify DataExtension. The name doesn’t follow the used convention to disambiguate with QBDataExt class, defined in Auxiliary Classes.

Classes and enums in blue are defined below in Auxiliary Classes and Enums:

Variable types bool, int, float, double, DateTime are defined as nullable.

public class QBDataExtADM

{

public string OwnerID

public string DataExtName

public QBORListTxn ORListTxn

public string DataExtValue

}

**QBDataExtADM Class**

This class is used to read, save and modify the DataExtensions. Inherit a list of QBDataExtADM objects . Properties, methods an example similar to QBInvoices.

**Auxiliary Classes**

public class QBRef

{

public string ListID

public string FullName

}

public class QBError

{

public bool Status;

public string Message;

public Object Other;

public int? Code;

}

public class QBRef

{

public string ListID

{ get; set; }

public string FullName

{ get; set; }

}

public class QBAddress

{

public string Name

public string Addr1

public string Addr2

public string Addr3

public string Addr4

public string Addr5

public string City

public string State

public string PostalCode

public string Country

public string Note

public bool? DefaultShipTo

}

public class QBAddressBlock

{

public string Addr1

public string Addr2

public string Addr3

public string Addr4

public string Addr5

}

public class QBORRate

{

public double? Rate

public double? RatePercent

}

public class QBORRatePriceLevel

{

public double? Rate

public double? RatePercent

public QBRef PriceLevelRef

}

public class QBAppliedToTxn

{

public string TxnID

public double? PaymentAmount

public List<QBSetCredit> SetCreditList

public QBENTxnType? TxnType

public DateTime? TxnDate

public string RefNumber

public double? BalanceRemaining

public double? Amount

public double? DiscountAmount

public QBRef DiscountAccountRef

public QBRef DiscountClassRef

public List<QBLinkedTxn> LinkedTxnList

}

public class QBLinkedTxn

{

public string TxnID

public QBENTxnType? TxnType

public DateTime? TxnDate

public string RefNumber

public QBENLinkType? LinkType

public double? Amount

}

public class QBExpenseLine

{

public string TxnLineID

public QBRef AccountRef

public double? Amount

public string Memo

public QBRef CustomerRef

public QBRef ClassRef

public QBRef SalesTaxCodeRef

public QBENBillableStatus? BillableStatus

public QBRef SalesRepRef

public List<QBDataExt> DataExtList

public bool ClearItemLines

}

public class QBDataExt

{

public string OwnerID

public string DataExtName

public QBENDataExtType? DataExtType

public string DataExtValue

}

public class QBORSerialLotNumber

{

public string SerialNumber

public string LotNumber

}

public class QBTaxLineInfo

{

public int? TaxLineID

public string TaxLineName

}

public class QBLinkToTxn

{

public string TxnID

public string TxnLineID

public string LinkToTxnID

}

public class QBSetCredit

{

public string CreditTxnID

public double? AppliedAmount

public bool? Override

}

public class QBBarCode

{

public string BarCodeValue

public bool? AssignEvenIfUsed

public bool? AllowOverride

}

public class QBORInventorySiteORShipToEntity

{

public QBRef InventorySiteRef

public QBRef ShipToEntityRef

}

public class QBORPrice

{

public double? Price

public double? PricePercent

}

//PurchaseOrder

public class QBORPurchaseOrderLine

{

public QBPurchaseOrderLine PurchaseOrderLine

public QBPurchaseOrderLineGroup PurchaseOrderLineGroup

}

public class QBPurchaseOrderLine

{

public string TxnLineID

public QBRef ItemRef

public string ManufacturerPartNumber

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public double? Rate

public QBRef ClassRef

public double? Amount

public QBRef InventorySiteLocationRef

public QBRef CustomerRef

public DateTime? ServiceDate

public QBRef SalesTaxCodeRef

public double? ReceivedQuantity

public double? UnbilledQuantity

public bool? IsBilled

public bool? IsManuallyClosed

public string Other1

public string Other2

public QBRef OverrideItemAccountRef

public List<QBDataExt> DataExtList

}

public class QBPurchaseOrderLineGroup

{

public string TxnLineID

public QBRef ItemGroupRef

public string ListID

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public bool? IsPrintItemsInGroup

public double? TotalAmount

public List<QBPurchaseOrderLine> PurchaseOrderLineList

public List<QBDataExt> DataExtList

public QBRef InventorySiteLocationRef

}

//Invoice

public class QBORInvoiceLine

{

public QBInvoiceLine InvoiceLine

public QBInvoiceLineGroup InvoiceLineGroup

}

public class QBInvoiceLine

{

public string TxnLineID

public QBRef ItemRef

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public QBORRatePriceLevel ORRatePriceLevel

public QBORRate ORRate

public QBRef ClassRef

public double? Amount

public QBRef InventorySiteRef

public QBRef InventorySiteLocationRef

public QBORSerialLotNumber ORSerialLotNumber

public string ExpirationDateForSerialLotNumber

public DateTime? ServiceDate

public QBRef SalesTaxCodeRef

public string Other1

public string Other2

public List<QBDataExt> DataExtList

public QBRef PriceLevelRef;

public QBENOptionForPriceRuleConflict? OptionForPriceRuleConflict

public QBLinkToTxn LinkToTxn

public QBRef OverrideItemAccountRef

}

public class QBInvoiceLineGroup

{

public string TxnLineID

public QBRef ItemGroupRef

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public bool? IsPrintItemsInGroup

public double? TotalAmount

public List<QBInvoiceLine> InvoiceLineList

public List<QBDataExt> DataExtList

public QBRef InventorySiteLocationRef

public QBRef InventorySiteRef

}

//Bill

public class QBORItemLine

{

public QBItemLine ItemLine

public QBItemGroupLine ItemGroupLine

}

public class QBItemLine

{

public string TxnLineID

public QBRef ItemRef

public QBRef InventorySiteRef

public QBRef InventorySiteLocationRef

public QBORSerialLotNumber ORSerialLotNumber

public string SerialNumber

public string LotNumber

public string ExpirationDateForSerialLotNumber

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public double? Cost

public double? Amount

public QBRef CustomerRef

public QBRef ClassRef

public QBRef SalesTaxCodeRef

public QBENBillableStatus? BillableStatus

public QBRef SalesRepRef

public List<QBDataExt> DataExtList

public QBRef OverrideItemAccountRef

public QBLinkToTxn LinkToTxn;

public string ManufacturerPartNumber

public double? Rate

public DateTime? ServiceDate

}

public class QBItemGroupLine

{

public string TxnLineID

public QBRef ItemGroupRef

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public double? TotalAmount

public List<QBItemLine> ItemLineList

public List<QBDataExt> DataExtList

public QBRef InventorySiteRef

public QBRef InventorySiteLocationRef

}

//NonInventoryItem

public class QBORSalesPurchase

{

public QBSalesAndPurchase SalesAndPurchase

public QBSalesOrPurchase SalesOrPurchase

}

public class QBSalesOrPurchase

{

public string Desc

public QBORPrice ORPrice

public QBRef AccountRef

public bool? ApplyAccountRefToExistingTxns

}

public class QBSalesAndPurchase

{

public string SalesDesc

public double? SalesPrice

public QBRef IncomeAccountRef

public string PurchaseDesc

public double? PurchaseCost

public QBRef PurchaseTaxCodeRef

public QBRef ExpenseAccountRef

public QBRef PrefVendorRef

public bool? ApplyIncomeAccountRefToExistingTxns

public bool? ApplyExpenseAccountRefToExistingTxns

}

//Customer

public class QBAdditionalContact

{

public string ContactName

public string ContactValue

}

public class QBContact

{

public string ListID

public DateTime? TimeCreated

public DateTime? TimeModified

public string EditSequence

public string Contact

public string Salutation

public string FirstName

public string MiddleName

public string LastName

public string JobTitle

public List<QBAdditionalContact> AdditionalContactRefList

}

public class QBCreditCardInfo

{

public string CreditCardNumber

public int? ExpirationMonth

public int? ExpirationYear

public string NameOnCard

public string CreditCardAddress

public string CreditCardPostalCode

}

public class QBAdditionalNote

{

public int? NoteID

public DateTime? Date

public string Note

}

//PriceLevel

public class QBORORCustomPrice

{

public double? CustomPrice { get; set; }

public double? CustomPricePercent { get; set; }

}

public class QBORPriceLevel

{

public double? PriceLevelFixedPercentage { get; set; }

public List<QBPriceLevelPerItem> PriceLevelPerItemList { get; set; }

}

public class QBPriceLevelAdjustment

{

public double? AdjustPercentage { get; set; }

public QBENAdjustRelativeTo? AdjustRelativeTo { get; set; }

}

public class QBPriceLevelPerItem

{

public QBRef ItemRef { get; set; }

public QBORORCustomPrice ORORCustomPrice { get; set; }

public QBPriceLevelAdjustment PriceLevelAdjustment;

}

//ReceivePayment

public class QBCreditCardTxnInputInfo

{

public string CreditCardNumber

public int? ExpirationMonth

public int? ExpirationYear

public string NameOnCard

public string CreditCardAddress

public string CreditCardPostalCode

public string CommercialCardCode

public QBENTransactionMode? TransactionMode

public QBENCreditCardTxnType? CreditCardTxnType

}

public class QBCreditCardTxnResultInfo

{

public int? ResultCode

public string ResultMessage

public string CreditCardTransID

public string MerchantAccountNumber

public string AuthorizationCode

public QBENAVSStreet? AVSStreet

public QBENAVSZip? AVSZip

public QBENCardSecurityCodeMatch? CardSecurityCodeMatch

public string ReconBatchID

public int? PaymentGroupingCode

public QBENPaymentStatus? PaymentStatus

public DateTime? TxnAuthorizationTime

public int? TxnAuthorizationStamp

public string ClientTransID

}

public class QBCreditCardTxnInfo

{

public QBCreditCardTxnInputInfo CreditCardTxnInputInfo

public QBCreditCardTxnResultInfo CreditCardTxnResultInfo

}

public class QBORApplyPayment

{

public bool? IsAutoApply

public List<QBAppliedToTxn> AppliedToTxnList

}

//BillPaymentCheck

public class QBORCheckPrint

{

public bool? IsToBePrinted

public string RefNumber

}

//CreditMemo

public class QBORCreditMemoLine

{

public QBCreditMemoLine CreditMemoLine

public QBCreditMemoLineGroup CreditMemoLineGroup

}

public class QBCreditMemoLine

{

public string TxnLineID

public QBRef ItemRef

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public QBORRate ORRate

public QBORRatePriceLevel ORRatePriceLevel

public QBRef ClassRef

public double? Amount

public QBRef InventorySiteRef

public QBRef InventorySiteLocationRef

public QBORSerialLotNumber ORSerialLotNumber

public string ExpirationDateForSerialLotNumber

public DateTime? ServiceDate

public QBRef SalesTaxCodeRef

public string Other1

public string Other2

public QBCreditCardTxnInfo CreditCardTxnInfo

public List<QBDataExt> DataExtList

public QBRef OverrideItemAccountRef

}

public class QBCreditMemoLineGroup

{

public string TxnLineID

public QBRef ItemGroupRef

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public bool? IsPrintItemsInGroup

public double? TotalAmount

public List<QBCreditMemoLine> CreditMemoLineList

public QBRef InventorySiteLocationRef

public QBRef InventorySiteRef

public List<QBDataExt> DataExtList

}

//Deposit

public class QBCashInfo

{

public string TxnLineID

public QBRef AccountRef

public string ListID

public string FullName

public string Memo

public double? Amount

public string ExternalGUID

}

public class QBORDepositLine

{

public QBPaymentLine PaymentLine

public QBDepositInfo DepositInfo

}

public class QBDepositInfo

{

public QBRef EntityRef

public string ListID

public string FullName

public QBRef AccountRef

public string Memo

public string CheckNumber

public QBRef PaymentMethodRef

public QBRef ClassRef

public double? Amount

}

public class QBPaymentLine

{

public string PaymentTxnID

public string PaymentTxnLineID

public string OverrideMemo

public string OverrideCheckNumber

public QBRef OverrideClassRef

}

public class QBDepositLine

{

public QBENTxnType? TxnType

public string TxnID

public string TxnLineID

public string PaymentTxnLineID

public QBRef EntityRef

public QBRef AccountRef

public string Memo

public string CheckNumber

public QBRef PaymentMethodRef

public QBRef ClassRef

public double? Amount

public QBORDepositLine ORDepositLine

}

//SalesOrder

public class QBSalesOrderLine

{

public string TxnLineID

public QBRef ItemRef

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef OverrideUOMSetRef

public QBORRate ORRate

public QBORRatePriceLevel ORRatePriceLevel

public QBRef ClassRef

public double? Amount

public QBENOptionForPriceRuleConflict? OptionForPriceRuleConflict

public QBRef InventorySiteRef

public QBRef InventorySiteLocationRef

public QBORSerialLotNumber ORSerialLotNumber

public string ExpirationDateForSerialLotNumber

public QBRef SalesTaxCodeRef

public string ListID

public string FullName

public double? Invoiced

public bool? IsManuallyClosed

public string Other1

public string Other2

public List<QBDataExt> DataExtList

}

public class QBSalesOrderLineGroup

{

public string TxnLineID

public QBRef ItemGroupRef

public string Desc

public double? Quantity

public string UnitOfMeasure

public QBRef InventorySiteRef

public QBRef InventorySiteLocationRef

public QBRef OverrideUOMSetRef

public bool? IsPrintItemsInGroup

public double? TotalAmount

public List<QBSalesOrderLine> SalesOrderLineList

public List<QBDataExt> DataExtList

}

public class QBShippingDetailsLine

{

public string TrackingID

public string CarrierName

public string ShippingMethod

public double? ShippingCharges

public string SOChannel

public string StoreName

public string StoreType

}

public class QBORSalesOrderLine

{

public QBSalesOrderLine SalesOrderLine

public QBSalesOrderLineGroup SalesOrderLineGroup

}

//DataExtensions

public class QBORListTxn

{

public QBListDataExt ListDataExt

public QBTxnDataExt TxnDataExt

public QBENOtherDataExtType? OtherDataExtType

}

public class QBListDataExt

{

public QBENListDataExtType? ListDataExtType

public QBRef ListObjRef

}

public class QBTxnDataExt

{

public QBENTxnDataExtType? TxnDataExtType

public string TxnID

public string TxnLineID

}

**Enumerations**

public enum QBENDataExtType

{

detAMTTYPE = 0,

detDATETIMETYPE = 1,

detINTTYPE = 2,

detPERCENTTYPE = 3,

detPRICETYPE = 4,

detQUANTYPE = 5,

detSTR1024TYPE = 6,

detSTR255TYPE = 7

}

public enum QBENTxnType

{

ttARRefundCreditCard = 0,

ttBill = 1,

ttBillPaymentCheck = 2,

ttBillPaymentCreditCard = 3,

ttBuildAssembly = 4,

ttCharge = 5,

ttCheck = 6,

ttCreditCardCharge = 7,

ttCreditCardCredit = 8,

ttCreditMemo = 9,

ttDeposit = 10,

ttEstimate = 11,

ttInventoryAdjustment = 12,

ttInvoice = 13,

ttItemReceipt = 14,

ttJournalEntry = 15,

ttLiabilityAdjustment = 16,

ttPaycheck = 17,

ttPayrollLiabilityCheck = 18,

ttPurchaseOrder = 19,

ttReceivePayment = 20,

ttSalesOrder = 21,

ttSalesReceipt = 22,

ttSalesTaxPaymentCheck = 23,

ttTransfer = 24,

ttVendorCredit = 25,

ttYTDAdjustment = 26

}

public enum QBENLinkType

{

ltAMTTYPE = 0,

ltQUANTYPE = 1

}

public enum QBENBillableStatus

{

bsBillable = 0,

bsNotBillable = 1,

bsHasBeenBilled = 2

}

public enum QBENOptionForPriceRuleConflict

{

ofprcZero = 0,

ofprcBasePrice = 1

}

public enum QBENAccountType

{

atAccountsPayable = 0,

atAccountsReceivable = 1,

atBank = 2,

atCostOfGoodsSold = 3,

atCreditCard = 4,

atEquity = 5,

atExpense = 6,

atFixedAsset = 7,

atIncome = 8,

atLongTermLiability = 9,

atNonPosting = 10,

atOtherAsset = 11,

atOtherCurrentAsset = 12,

atOtherCurrentLiability = 13,

atOtherExpense = 14,

atOtherIncome = 15

}

public enum QBENSpecialAccountType

{

satAccountsPayable = 0,

satAccountsReceivable = 1,

satCondenseItemAdjustmentExpenses = 2,

satCostOfGoodsSold = 3,

satDirectDepositLiabilities = 4,

satEstimates = 5,

satExchangeGainLoss = 6,

satInventoryAssets = 7,

satItemReceiptAccount = 8,

satOpeningBalanceEquity = 9,

satPayrollExpenses = 10,

satPayrollLiabilities = 11,

satPettyCash = 12,

satPurchaseOrders = 13,

satReconciliationDifferences = 14,

satRetainedEarnings = 15,

satSalesOrders = 16,

satSalesTaxPayable = 17,

satUncategorizedExpenses = 18,

satUncategorizedIncome = 19,

satUndepositedFunds = 20

}

public enum QBENCashFlowClassification

{

cfcNone = 0,

cfcOperating = 1,

cfcInvesting = 2,

cfcFinancing = 3,

cfcNotApplicable = 4

}

public enum QBENSalesTaxCountry

{

stcAustralia = 0,

stcCanada = 1,

stcUK = 2,

stcUS = 3

}

public enum QBENJobStatus

{

jsAwarded = 0,

jsClosed = 1,

jsInProgress = 2,

jsNone = 3,

jsNotAwarded = 4,

jsPending = 5

}

public enum QBENPreferredDeliveryMethod

{

pdmNone = 0,

pdmEmail = 1,

pdmFax = 2

}

public enum QBENReportingPeriod

{

rpMonthly = 0,

rpQuarterly = 1

}

public enum QBENConnectionType

{

ctUnknown = 0,

ctLocalQBD = 1,

ctRemoteQBD = 2,

ctLocalQBDLaunchUI = 3,

ctRemoteQBOE = 4

}

public enum QBENOpenMode

{

omSingleUser = 0,

omMultiUser = 1,

omDontCare = 2

}

public enum QBENRqOnError

{

roeStop = 0,

roeContinue = 1,

roeRollback = 2

}

public enum QBENPriceLevelType

{

pltFixedPercentage,

pltPerItem

}

public enum QBENAdjustRelativeTo

{

artStandardPrice,

artCost,

artCurrentCustomPrice

}

public enum QBENTransactionMode

{

tmCardNotPresent = 0,

tmCardPresent = 1

}

public enum QBENCreditCardTxnType

{

ccttAuthorization = 0,

ccttCapture = 1,

ccttCharge = 2,

ccttRefund = 3,

ccttVoiceAuthorization = 4

}

public enum QBENAVSStreet

{

avssPass = 0,

avssFail = 1,

avssNotAvailable = 2

}

public enum QBENAVSZip

{

avszPass = 0,

avszFail = 1,

avszNotAvailable = 2

}

public enum QBENCardSecurityCodeMatch

{

cscmPass = 0,

cscmFail = 1,

cscmNotAvailable = 2

}

public enum QBENPaymentStatus

{

pssUnknown = 0,

pssCompleted = 1

}

public enum QBENSOChannel

{

socBlank = 0,

socEcommerce =1

}

public enum QBENListDataExtType

{

ldetAccount = 0,

ldetCustomer = 1,

ldetEmployee = 2,

ldetItem = 3,

ldetOtherName = 4,

ldetVendor = 5

}

public enum QBENTxnDataExtType

{

tdetARRefundCreditCard = 0,

tdetBill = 1,

tdetBillPaymentCheck = 2,

tdetBillPaymentCreditCard = 3,

tdetBuildAssembly = 4,

tdetCharge = 5,

tdetCheck = 6,

tdetCreditCardCharge = 7,

tdetCreditCardCredit = 8,

tdetCreditMemo = 9,

tdetDeposit = 10,

tdetEstimate = 11,

tdetInventoryAdjustment = 12,

tdetInvoice = 13,

tdetItemReceipt = 14,

tdetJournalEntry = 15,

tdetPurchaseOrder = 16,

tdetReceivePayment = 17,

tdetSalesOrder = 18,

tdetSalesReceipt = 19,

tdetSalesTaxPaymentCheck = 20,

tdetVendorCredit = 21

}

public enum QBENOtherDataExtType

{

odetCompany = 0

}