

Anchor Webservice

© 2018 Anchor Distributors



Table of Contents

| | |
|-------------------------------------|----------|
| Foreword | 0 |
| Part I Introduction | 3 |
| 1 About | 3 |
| 2 What's new..... | 4 |
| Part II Vendor Documentation | 4 |
| 1 Overview..... | 4 |
| 2 General Concepts..... | 4 |
| Anchor WebService | 4 |
| Login & Authentication | 5 |
| Structures and Functions | 6 |
| Function Calls Explanation | 11 |
| VB Samples | 13 |
| More Samples | 15 |
| XML Sample | 17 |
| Minimum Functions | 20 |
| 3 Initial database..... | 21 |
| Files | 21 |
| Index | 0 |

1 Introduction

1.1 About

About

[Top](#) [Next](#)

Web service is an interface for 3rd party vendor to do business with us. (B2B partner).

Mainly used to populate their own website with our Database Information (items) and to submit order to us.

Typically, the vendors are Anchor's Bill to customer and Vendor's customers are Anchor's ship to.

Here's the quick overview of the way our web service works:

1. Populate your own database with our database.
(updated automatically on your end, via tab delimited file, uploaded to your FTP server or via email)
2. Create the end customer account on our end
3. Create shopping cart on your website for customers
4. Access our web services for shipping rate (Flat or Rated)
5. Submit orders to us via web services
6. end of day - query status of orders via web services (for tracking numbers, and refund of out of stock items).

QUICK FAQ

The web service can be described as a series of Function Calls to Anchor Distributor's database.

THERE IS NO user interface for it from our end. If you need a user interface, your programmer have to create it for you.

Note: since each order placed will be from your main account, you can log into our website and see all the orders. this is the only user interface you get from us.

Each web service Function calls can be used by your programmer, to create a user interface, customized to your liking, even linked to your own website/database.

Note: Anchor's web service is only available for vendors with their own programmers.

here's a more detailed steps on the process of using Anchor Webservices:

1. An initial file to build a database is needed (this comes from Anchor), this file can be sent via FTP each week or each night to your FTP server from our end.
This file contain information on all the items that you want to carry in our warehouse.
Why everynight/week? For changes and additions to Anchor database.
2. With that file, your programmer have to do create a database on your end.
Your programmer also have to create a daily/weekly update to read the file that we FTP-ed to update your database.
3. When a user login to your site and place an order:
 - a. Your programmer can check the availability of the item to see that it is available (a call to one of the web services).
 - b. A shipping rate function should also be called to show all available shipping methods.
(this can be called as they add more items, or at the end of an order).
 - c. when the user finalize the order, your programmer will call the function to create a SHIP TO account (with their name and address, etc2), then call the function to submit the order to Anchor. (this function will return an invoice number, which your programmer have to capture and save in your own database, so that you can track this later).
4. Once an order comes to Anchor, depending on the status of your account, the order either

- goes straight to the warehouse for dispatch, or Credit department will review the order first.
5. You can ask your programmer to create a nice user interface to check each order using the invoice numbers that is captured with each submission. (this will in turn show the status of invoices and tracking numbers(if available)). Again, this is all done by calling one of the webservice functions. OR, your programmer can use the same call to create a “tracking” for your customer so they can check their order themselves.
 6. Your programmer also have to check/compare the order that is being submitted to the one that is being shipped because that item might be out of stock(for any reason).. and your programmer have to make sure that the customer/end user is aware that their order has been changed.(This might involve a refund process to your customer as well).
 7. When our connection to the web is severed (which can happen), your programmer have to make it so that each order that hasn't come to us, has to be automatically submitted (maybe within next hour, 15 minutes, etc2). Also, a webservice function call can be failed too(for any reason). The programmer have to be aware of this and make sure to retry so the call is good and the order goes through.

1.2 What's new

What's new

[Top](#) [Previous](#) [Next](#)

- 4th Party Store name and information can be passed on to us just for Pick-Slip printing purposes.
- Pick-slip for Vendors will be printed from separate printer on the pick-server.

2 Vendor Documentation

2.1 Overview

Overview

[Top](#) [Previous](#) [Next](#)

This Section of the documentation is aimed for the Vendor who will use Anchor Webservice as to consume Anchor Distributor's database.

2.2 General Concepts

2.2.1 Anchor WebService

Anchor WebService

[Top](#) [Previous](#) [Next](#)

Anchor Distributor Webservice was developed using Microsoft Visual Basic .NET 2003 and reside on a Microsoft IIS 6.0 machine.

Here's the Address of the WebService:

- Deprecated(end of 2018): <http://ws.anchor distributors.com/anchorwebservice.asmx>

Current: <https://soap.anchor distributors.com/anchorwebservice.asmx>

Below is the Snapshot of what the WebService looks like..

AnchorWebservice

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [DeleteOrder](#)
Return Boolean, given an Invoice Seq ID, Customer Seq ID, Item Seq ID as input.
Return TRUE if successful.
- [GetCountries](#)
Return an Array of ListValue2 Structure with Country Seq ID, Country Name, and Country Code.
- [GetItemAvailable](#)
Return ListValue3 Structure of Item Seq Id, Unit Available, Item_Status_Code, Item_Status_Description given an array of Item Seq Id.
Useful when querying Item(s) availability.
- [GetOrderStatus](#)
Return SalesOrderStatus Structure, given an Invoice Seq Id input.
Useful to see the Status of Sales Order, Tracking number(s) is included for Shipped Invoice.
- [GetProcessedInvoices](#)
Return Integer of Invoice Seq Id, given an Account Number(bill to), Date Start and Date end.
Returns Array of Shipped Invoice(s) for specific date_shipped range.
- [GetProcessedOrders](#)
Return ProcessedOrder Structure, given an Account Number(bill to), Date Start and Date end.
Returns Array of Processed Invoice(s) and tracking number(s) for specific date range.
- [GetProductAuthors](#)
Return an Array of ListValue2 Structure with Author SEQ ID, Author Name, and Total Number of Titles for that Author.
- [GetProductBestsellers](#)
Return an Array of ListValue Structure with BestSeller Category SEQ ID and BestSeller Description.
- [GetProductBindings](#)
Return an Array of ListValue2 Structure with Binding SEQ ID, Binding Description, and Binding Code.
- [GetProductBlurb](#)
Return ListValue Structure of Item Seq Id and Blurb(Description), given an array of Item Seq Id.
Use this to grab Blurb/Description of an item (if available).
- [GetProductByDate](#)
Return Array of Standard_Product_Database, given a Date Addition Start and End.
Used to Populate Web Service Consumer's Item Database.
- [GetProductCategories](#)
Return an Array of ListValue Structure with Category SEQ ID and Category Description.
- [GetProductExtended](#)
Return a Product Extended Structure given an Item Seq Id input, to get Product Extended Information.
- [GetProductPublishers](#)
Return an Array of ListValue2 Structure with Publisher SEQ ID, Publisher Name, and Total Number of Titles for that Publisher.
- [GetProductRecordset](#)
Return an Array of ListValue Structure with BestSeller Category SEQ ID and BestSeller Description.
- [GetProductSubjects](#)
Return an Array of ListValue Structure with Subject SEQ ID and Subject Description.
- [GetProductSubjects_Code](#)
Return an Array of ListValue Structure with Subject SEQ ID and Subject Code.
- [GetProductSubjects_Complete](#)
Return an Array of ListValue2 Structure with Subject SEQ ID, Subject Code, and Subject Description.

2.2.2 Login & Authentication

Login & Authentication

[Top](#) [Previous](#) [Next](#)

In order to use Anchor Webservice, a Special Account needs to be established with Anchor.

note the term special here means if vendor wish for us to do fulfilment, we will then create a company profile and logos on pick-slip.

Once an account is created, Vendor can try to login to Anchor Webservice using SOAP Header Authentication Methods.

here's an example of how to login: ****PLEASE CHECK on MORE SAMPLES for the latest example****

- in Visual Basic .Net:

```
Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    btnLogin.Click
```

```
Dim wsTest As ADWS.AnchorWebservice = New ADWS.AnchorWebservice
```

```
Dim shTest As ADWS.SecurityHeader
```

```
shTest = New ADWS.SecurityHeader
```

```
shTest.Username = Val(tbUsername.Text)
shTest.Password = tbPassword.Text
wsTest.SecurityHeaderValue = shTest
```

```
lbOutput.Text = wsTest.SecurityTest() ' This is a Function that will return "Login Successful"
```

```
End sub
```

- in PHP:
<?PHP

```
ini_set("max_execution_time",3000000);
require("soaplib/nusoap.php");
```

```
// No params required for this soap call
```

```
$parms = array();
```

```
$soapclient = new soapclient('https://soap.anchor distributors.com/AnchorWebservice.asmx?WSDL',
"wsdl");
```

```
$hdr = "<SecurityHeader xmlns='https://tempuri.org/AnchorWebservice/AnchorWebservice'>
<Username>USERNAME</Username>
<Password>PASSWORD</Password>
</SecurityHeader>";
```

```
$result = $soapclient->call( 'SecurityTest',
$parms,
'https://soap.anchor distributors.com/',
'https://soap.anchor distributors.com/AnchorWebservice/AnchorWebservice/SecurityTest',
$hdr,
false,
'rpc',
'literal'
);
```

```
die('HERE IT IS ' . $soapclient->response . ' <BR><BR><BR>' . print_r($result, true));
```

```
?>
```

Note that the **Username** and **Password** will be given when an Account is created and To access a web service function you must always pass a username and password in the soap security header.

2.2.3 Structures and Functions

Structures and Functions

[Top](#) [Previous](#) [Next](#)

in order to use the Webservice, please note that there are various Structures and Functions throughout the systems.

Visit: **<https://soap.anchor distributors.com/anchorwebservice.asmx>** for a list of current functions available from Anchor Webservice.

Here's the list of Data Structures that might be passed in or returned from a function:

```
Public Structure ListValue
Public SEQ_ID As Integer
Public Description As String
End Structure
```

```
Public Structure QueryCriteria
Public Page As Integer
```

```
Public Items_Per_Page As Integer
Public Lookup_Code As String
Public Title As String
Public Title_3_Part As String
Public Author As String
Public Publisher As String
Public Subject As Integer
Public Category As Integer
Public Best_Sellers As Integer
Public Binding As Integer
End Structure

Public Structure ListValue2
Public SEQ_ID As Integer
Public Description_1 As String
Public Description_2 As String
End Structure

Public Structure ListValue3
Public SEQ_ID As Integer
Public Description_1 As String
Public Description_2 As String
Public Description_3 As String
Public Description_4 As String
End Structure

Public Class QueryResult
Public Page_Count As Integer
Public Row_Count As Integer
Public Items As Product()
End Class

<Serializable(> Public Class Product
Public Seq_Id As Integer
Public Status As String
Public Title As String
Public ISBN As String
Public SpeedyCode As String
Public UPC As String
Public Price As Decimal
Public Discount As Decimal
Public Author As String
Public Publisher As String
Public Subject As String
Public Binding As String
Public Available As String
End Class

<Serializable(> Public Class ProductExtended
Inherits Product
Public CaseQty As Decimal
Public Blurb As String
Public TOC As String
Public Excerpt As String
Public Image_Filename As String
```

```
Public Date_First_Published As String
Public Date_Due As String
Public BackCover As String
Public Weight As Decimal
Public Height As Decimal
End Class
<Serializable(> Public Class ProductPOD
Public Seq_Id As Integer
Public SpeedyCode As String
Public ISBN13 As String
Public Title As String
Public Width As Decimal
Public Length As Decimal
Public Height As Decimal
Public PageCount As Decimal
Public Author_First_Name As String
Public Author_Last_Name As String
Public Publisher_Name As String
Public Blurb As String
Public Image_Filename As String
End Class
```

```
Public Structure SalesOrderStatus
Public Status As String
Public Tracking As String()
End Structure
```

```
Public Structure ShipToCustomer
Public Bill_To_Seq_Id As Integer
Public Ship_To_Seq_Id As Integer
Public Name As String
Public Street1 As String
Public Street2 As String
Public City As String
Public State As String
Public ZipCode As String
Public Country As String
Public Telephone As String
Public Fax As String
Public Email As String
Public Contact As String
Public Customer_Type_Seq_Id As Integer
Public Ship_method_Seq_Id As Integer
Public Country_Seq_Id As Integer
End Structure
```

' NOTE: salesorderimprint and salesorderdetailimprint is needed as the structure changes = some customer can't submit order.

```
Public Structure SalesOrderDetail
Public Product_Seq_Id As Integer
Public Order_Quantity As Integer
Public Ship_Quantity As Integer
Public Unit_Price As Decimal
Public Discount As Decimal
Public Extension As Decimal
```

```
Public Customer_ID As String
End Structure
```

```
Public Structure SalesOrderDetailImprint
Public Product_Seq_Id As Integer
Public Order_Quantity As Integer
Public Ship_Quantity As Integer
Public Unit_Price As Decimal
Public Discount As Decimal
Public Extension As Decimal
Public Customer_ID As String
Public imprint_font_style_id As Integer
Public imprint_text_line1 As String
Public imprint_text_line2 As String
Public indexing_color_id As Integer
End Structure
```

```
<Serializable(> Public Structure SalesOrder
Public Invoice_Seq_Id As Integer
Public Bill_to_Seq_Id As Integer
Public Ship_to_Seq_Id As Integer
Public PO_Number As String
Public Net As Decimal
Public Flag_Rush_Order As String
Public Date_Ship_By As String
Public Shipping_Charge As Decimal
Public SO_Detail As SalesOrderDetail()
Public Flag_All_Complete As String
Public Ship_method_Seq_Id As Integer
Public Store_Name As String
Public Store_Message As String
Public Store_Street As String
Public Store_City As String
Public Store_State As String
Public Store_ZIP As String
Public Store_Country As String
Public Intl_Tax_Number As String
Public Intl_Tax_Description As String
Public Intl_Tax_Amount As Decimal
Public Special_Instruction As String
Public Date_Shipped As String
End Structure
```

```
<Serializable(> Public Structure SalesOrderImprint
Public Invoice_Seq_Id As Integer
Public Bill_to_Seq_Id As Integer
Public Ship_to_Seq_Id As Integer
Public PO_Number As String
Public Net As Decimal
Public Flag_Rush_Order As String
Public Date_Ship_By As String
Public Shipping_Charge As Decimal
Public SO_Detail As SalesOrderDetailImprint()
Public Flag_All_Complete As String
Public Ship_method_Seq_Id As Integer
```

```
Public Store_Name As String
Public Store_Message As String
Public Store_Street As String
Public Store_City As String
Public Store_State As String
Public Store_ZIP As String
Public Store_Country As String
Public Intl_Tax_Number As String
Public Intl_Tax_Description As String
Public Intl_Tax_Amount As Decimal
Public Special_Instruction As String
Public Date_Shipped As String
End Structure
```

```
Public Structure ProcessedOrder
Public Invoice As String()
Public Tracking As String()
End Structure
```

```
Public Structure QueryTest
Public Invoice As Integer
Public Flag_Rush As String
End Structure
```

```
Public Structure Shipping_Rate
Public Ship_rate As Decimal
Public Message As String
End Structure
```

```
<Serializable(> Public Structure Standard_Product_Database
Public Item_seq_id As Integer
Public Speedy_Code As String
Public ISBN As String
Public Title As String
Public Author_seq_id As Integer
Public Price As Decimal
Public UPC As String
Public Primary_CBC As String
Public Sub_CBC As String
Public Super_CBC As String
Public Weight As Decimal
Public Discount As Decimal
Public Flag_New As String
Public Date_First_Sale As Date
Public Vendor_Code As String
Public Binding_Seq_Id As Integer
Public Category_Seq_Id() As Integer 'Note that this is actually Catalog_section_seq_id!!
Public Available As Integer
Public CaseQty As Integer
Public Blurb As String
Public TOC As String
Public Excerpt As String
Public Image_Filename As String
Public Date_First_Published As Date
Public Date_Due As String
```

```

    Public BackCover As String
End Structure

Public Structure POD_Order
    Public Invoice_seq_id As Integer
    Public POD_Order_Status_seq_id As Integer
    Public Date_Started As Date
    Public Date_Acknowledged As Date
    Public Date_Printed As Date
    Public Date_Picked As Date
    Public Date_Cancelled As Date
    Public Item_seq_id As Integer
    Public Item_counter As Integer
    Public Quantity As Integer
    Public Location_Code As String
    Public POD_Type_Id As Integer
    Public Putaway_Code As Integer
    Public PO_SEQ_ID As Integer
    Public POD_SEQ_ID As Integer
End Structure

Public Structure ShippingCharges
    Public Invoice As String()
    Public Shipping_Charge As Decimal()
End Structure

```

2.2.4 Function Calls Explanation

Function Calls Explanation

[Top](#) [Previous](#) [Next](#)

Here is the explanation of how the Functions call works:

1. Products related:

Functions started with [GetProduct...](#) are used to populate search criteria (combo boxes). Usually it will return [Listvalue](#), [Listvalue2](#), or [Listvalue3](#) structure (IDs and Descriptions).

[QueryProduct](#) Function is used to get items given a set of search criteria.
(Input is [QueryCriteria](#) structure, output is [QueryResult](#) structure).

2. Shipping related:

[GetCountries](#) will return Country ID and Name. This is needed for Non-USA orders.
- Use this to create a lookup tables on your database.

[GetShipMethods](#) will return Ship Method ID and ship method offered by Anchor distributors.
- Use this to create a lookup tables on your database

[GetVariousShipRates](#) will return Various Ship Rates, given weight, zipcode, and country code
- Use this function to get rated shipping cost,
it will return an array of shipping methods available given the weight, zipcode, and country code.

[GetVariousFlatRates](#) will return Various FLAT shipping Rates,
given total number of items, weight(use 1 if N/A), zipcode(use blank space if N/A), and country ID.
- Use this function to get a flat shipping cost, it will return an array of flat rate shipping methods

available.

Below is an alternative ship rate functions:

(not recommended as you need the exact Shipping Method ID, will only return 1 value for that specific ship method).

[GetShippingRate](#) is used to quote a shipping charge.(USA only).
(Input is weight, ship method ID, zip code)

[GetShippingRate2](#) is used to quote a shipping charge (all orders).
(Input is weight, ship method ID, zip code, country ID)

*note that weight is the sum of products weight.(each product weight is returned back by ProductExtended structure)

3. Setting up an Order:

The first step would be to create a SHIP TO customer using [SubmitShipToAccount](#) function. The Bill To ID is your account number.

If the SHIP TO customer is new, leave it blank and we will return the Customer Ship To ID.

For a Returning Customer, submit the ID to avoid creating a new duplicated account.

Build the Product shopping cart (an array of [SalesOrderDetail](#) structure), and when the customer is ready for checkout, sum the weight of the products and let the customer select their ship method preference. (a Shipping rate then obtained).

Call the [SubmitOrder](#) function(Input is [SalesOrder](#) Structure). Remember to submit the Shipping method ID and Shipping Charge field as selected by the customer. Failure to do this will result in order being sent by the default for the Ship To customer (default shipping method is saved when creating or updating SHIP TO customer).

*note that Customer ID field inside the SalesOrderDetail Structure can be used as a Purchase Order reference for each product(if necessary).

If there is a need for a Bill to name or address changes or custom message (such as 4th party store name), a [SubmitOrderWithStore](#) function can be called instead. This will have the Store Name, Street, City, State, Zip, Country, and a Message(used for birthday wishes for example) to be shown on the Pick-Slip being sent to customer.

4. Checking Order Status, Tracking number:

Once an order is submitted, you can check its status by calling these functions:

- [GetOrderStatus](#)

Return [SalesOrderStatus](#) Structure, given an Invoice Seq Id input.

Useful to see the Status of Sales Order, Tracking number(s) is included for Shipped Invoice.

- [GetProcessedInvoices](#)

Return [Integer](#) of Invoice Seq Id, given an Account Number(bill to), Date Start and Date end.

Returns Array of Shipped Invoice(s) for specific date_shipped range

- [GetProcessedOrders](#)

Return [ProcessedOrder](#) Structure, given an Account Number(bill to), Date Start and Date end.

Returns Array of Processed Invoice(s) and tracking number(s) for specific date range

- [GetShippedInvoiceDetails](#)

Return [SalesOrder](#) Structure, given an Invoice Seq ID.

Returns Order information and items.

Useful to check if any items is unfortunately not shipped due to out of stock.

The difference between the above are:

For [GetOrderStatus](#), the input is a single Invoice, returning the status of that 1 invoice with tracking number(s).

For [GetProcessedInvoices](#), the input is your account number and a date range, returning shipped Invoice(s) for the date range.

For [GetProcessedOrders](#), the same as [GetProcessedInvoices](#), but it will also return tracking number(s) for those invoices fall on the date range.

[GetShippedInvoiceDetails](#) is useful to query what really shipped on an Invoice. (possible out of stock items). If you want to give a refund on out of stock items, this can be done by comparing the order on your end and the final shipped invoice detail using this function.

2.2.5 VB Samples

VB Samples

[Top](#) [Previous](#)

Sample VB Code to Get Authors:

```
Dim wsTest As ADWS.AnchorWebservice = New ADWS.AnchorWebservice
Dim shTest As LOCAL_ADWS.SecurityHeader
'Always set SOAP Header as Authentication Methods
shTest = New LOCAL_ADWS.SecurityHeader
shTest.Username = Val(tbUsername.Text)
shTest.Password = tbPassword.Text
wsTest.SecurityHeaderValue = shTest

'Get Author
Dim AuthorArray As LOCAL_ADWS.ListValue2()
AuthorArray = wsTest.GetProductAuthors()

Dim i As Integer
Dim sString As String
If Not IsNothing(AuthorArray) Then
    For i = 1 To UBound(AuthorArray) - 1
        String = sString & AuthorArray(i).SEQ_ID & " " & AuthorArray
            (i).Description_1 & " " & AuthorArray(i).Description_2 &
                vbCrLf
    Next
Else
    sString = "Array Returns Nothing"
End If
lbOutput.Text = sString
```

Sample VB Code to access Query Product:

```
Dim wsTest As ADWS.AnchorWebservice = New ADWS.AnchorWebservice
Dim shTest As ADWS.SecurityHeader
Dim QCTest As ADWS.QueryCriteria
Dim QPResult As ADWS.QueryResult

'Always set SOAP Header as Authentication Methods
```

```

shTest = New ADWS.SecurityHeader

shTest.Username = "12345"
shTest.Password = "test"
wsTest.SecurityHeaderValue = shTest

'Test Query a Product
QCTest = New ADWS.QueryCriteria
QCTest.Page = 1
QCTest.Items_Per_Page = 10
QCTest.Lookup_Code = ""
QCTest.Title = ""
QCTest.Title_3_Part = ""
QCTest.Author = ""
QCTest.Publisher = ""
QCTest.Subject = "286"
QCTest.Category = 0
QCTest.Best_Sellers = 0
QCTest.Binding = 0

QPResult = New ADWS.QueryResult

QPResult = wsTest.QueryProduct(QCTest)

' Output data collection array of items.

lblResultSummary.Text = "Page Count = " & QPResult.Page_Count.ToString
& vbCrLf & "Row Count = " & QPResult.Row_Count.ToString & " " &
QPResult.Items(0).Title & " " & QPResult.Items(0).Discount

```

Sample VB Code to access SubmitOrderWithStore:

```

Dim wsTest As ADWS.AnchorWebservice = New ADWS.AnchorWebservice

Dim shTest As ADWS.SecurityHeader
Dim b As Boolean
'Always set SOAP Header as Authentication Methods
shTest = New ADWS.SecurityHeader
shTest.Username = Val(tbUsername.Text)
shTest.Password = tbPassword.Text

wsTest.SecurityHeaderValue = shTest

Dim SO As New ADWS.SalesOrder
Dim SOD As New ADWS.SalesOrderDetail

Dim al As New ArrayList
Dim i As Integer

Dim TempValue As ADWS.SalesOrderDetail

SO.Bill_to_Seq_Id = 12345
SO.Ship_to_Seq_Id = 54321
SO.Store_Name = "My Store name"
SO.Store_Message = "Happy Birthday to you.." & vbCrLf & "Wishing you
the best year ahead.."
SO.Store_Street = "12345 Street Ave"

```

```

SO.Store_City = "NewBurgh"
SO.Store_State = "PA"
SO.Store_ZIP = "15069"
SO.Store_Country = ""
SO.PO_Number = "TEST ON 29 FEB 2000"
SO.Flag_Rush_Order = "N"
SO.Flag_All_Complete = Nothing
SO.Shipping_Charge = 3.0
SO.Ship_method_Seq_Id = 8
SO.Net = 2 * 0.355
SO.Invoice_Seq_Id = 0
SO.Date_Ship_By = Format(DateTime.Now, "dd-MMM-yyyy")

TempValue = New ADWS.SalesOrderDetail
TempValue.Product_Seq_Id = 304660
TempValue.Order_Quantity = 1
TempValue.Customer_ID = Nothing
TempValue.Discount = 0.42
TempValue.Unit_Price = 9.97
TempValue.Ship_Quantity = 0
TempValue.Extension = 0

al.Add(TempValue)
TempValue = Nothing

SO.SO_Detail = al.ToArray(GetType(ADWS.SalesOrderDetail))

b = wsTest.SubmitOrderWithStore(SO)

If b Then
    lbOutput.Text = SO.Invoice_Seq_Id
Else
    lbOutput.Text = "False was returned, failed to Submit Order!!"
End If

```

2.2.6 More Samples

Below is some quick parameters to Test our Anchor Webservice.

Simple Login.

1. Check <https://soap.anchor distributors.com/anchorwebservice.asmx>
2. Click on the SecurityTest. It will show you that this is a very simple test of using your username and password.

<https://soap.anchor distributors.com/anchorwebservice.asmx?op=SecurityTest>

IF successful, it will return STRING values of "Login Successful".

VB.NET sample:

```

Dim wsTest As LIVE_ADWS.AnchorWebservice = New LIVE_ADWS.AnchorWebservice
Dim shTest As LIVE_ADWS.SecurityHeader
shTest = New LIVE_ADWS.SecurityHeader

shTest.Username = Val(tbUsername.Text)
shTest.Password = tbPassword.Text
wsTest.SecurityHeaderValue = shTest

```

```
lbOutput.Text = wsTest.SecurityTest()
```

Create Ship-To(your end customer where we will drop ship the items to)

1. Check [https://soap.anchor distributors.com/anchorwebservice.asmx?](https://soap.anchor distributors.com/anchorwebservice.asmx?op=SubmitShipToAccountWithError)

op=SubmitShipToAccountWithError

2. Notice the STRUCTURE of ShipToCustomer, which consist of various values of Customer data.

IF Successful, it will return the Customer_Seq_id assigned under your Account(which is the BILL TO).

sample values:

```
[ShipToCustomer] => Array
(
  [Bill_To_Seq_Id] => YOUR ACCOUNT HERE
  [Ship_To_Seq_Id] => LEAVE BLANK, IT WILL BE RETURNED if you successfully execute this
call.
  [Name] => John Doe
  [Street1] => 1014 TEST STREET
  [Street2] =>
  [City] => Springfield
  [State] => PA
  [ZipCode] => 15068
  [Country] => US
  [Telephone] => 1113334444
  [Fax] =>
  [Email] => store@yourdomain.com
  [Contact] => John Doe
  [Customer_Type_Seq_Id] => 1
  [Ship_method_Seq_Id] => 8
  [Country_Seq_Id] => 282
)
```

NOTE: leave customer_type_seq_id to 1, it will be assigned according to your Billing Default.

Ship_method_seq_id also will follow the default, which usually is Fedex or UPS ground.

country_seq_id default is USA, its 282. Use GetCountries function to find out the correct

IDs.

Create TEST order:

1. <https://soap.anchor distributors.com/anchorwebservice.asmx?op=SubmitOrderWithStoreandError>

2. Notice the STRUCTURE SO and SO_DETAIL. SO consist of Order information data, and

SO_DETAIL consists of ITEMS of the orders.

IF Successful, it will return your INVOICE number. Please save this number as you need to use it to call other functions.

(like getting Tracking number, checking if all items are shipped, which you need to do yourself, and give refund to your end customer if needed.)

sample values:

```
SO.Bill_to_Seq_Id = YOUR ACCOUNT NUMBER HERE
SO.Ship_to_Seq_Id = THE SHIP_TO_ID YOU GET FROM SAVING THE Ship-TO above
SO.Store_Name = "TEST STORE"
SO.Store_Message = "Thank You!"
SO.Store_Street = "111 Plaza Valenzia"
SO.Store_City = "Lake Elsinore"
SO.Store_State = "CA"
SO.Store_ZIP = "92532"
SO.Store_Country = "US"
```

```

SO.PO_Number = "TEST DO NOT RELEASE"
SO.Flag_Rush_Order = "N"
SO.Shipping_Charge = 4.17
SO.Ship_method_Seq_Id = 8
SO.Net = 25.35
SO.Invoice_Seq_Id = 0
SO.Date_Ship_By = "18-APR-2019" 'Format(DateTime.Now, "dd-MMM-yyyy")
SO.Intl_Tax_Amount = 0

```

```

TempValue = New LIVE_ADWS.SalesOrderDetail
TempValue.Product_Seq_Id = 466792
TempValue.Order_Quantity = 1
TempValue.Ship_Quantity = 1
TempValue.Unit_Price = 0
TempValue.Discount = 0
TempValue.Extension = 0
al.Add(TempValue)
TempValue = Nothing

```

```

TempValue = New LIVE_ADWS.SalesOrderDetail
TempValue.Product_Seq_Id = 436681
TempValue.Order_Quantity = 1
TempValue.Ship_Quantity = 1
TempValue.Unit_Price = 0
TempValue.Discount = 0
TempValue.Extension = 0
TempValue.Customer_ID = Nothing
al.Add(TempValue)
TempValue = Nothing

```

```
SO.SO_Detail = al.ToArray(GetType(LIVE_ADWS.SalesOrderDetail))
```

```
b = wsTest.SubmitOrderWithStoreandError(SO, sError)
```

NOTE: you must put "TEST" in the PO_NUMBER field, otherwise your TEST order will be processed.

2.2.7 XML Sample

Webservice Functions can also be called with XML Post.

note the xmlns soap, xsi, xsd values. THIS is crucial for the call to work. Below is a sample:

1. Create a new Ship to account under your Bill to Account.

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <SecurityHeader xmlns="http://tempuri.org/AnchorWebservice/AnchorWebservice">
      <Username>YOUR BILL TO ID</Username>
      <Password>YOUR PASSWORD</Password>
    </SecurityHeader>

```

```

</soap:Header>
<soap:Body>
  <SubmitShipToAccountWithError xmlns="http://tempuri.org/AnchorWebservice/AnchorWebservice">
    <ShipToCustomer>
      <Bill_To_Seq_Id>YOUR BILL TO ID</Bill_To_Seq_Id>
      <Ship_To_Seq_Id></Ship_To_Seq_Id>
      <Name>Drop Ship Customer Name</Name>
      <Street1>2889 Beach Street</Street1>
      <Street2>side porch</Street2>
      <City>Sacramento</City>
      <State>CA</State>
      <ZipCode>94273-0001</ZipCode>
      <Country>US</Country>
      <Telephone></Telephone>
      <Fax></Fax>
      <Email>test@test.com</Email>
      <Contact></Contact>
      <Customer_Type_Seq_Id>1</Customer_Type_Seq_Id>
      <Ship_method_Seq_Id>8</Ship_method_Seq_Id>
      <Country_Seq_Id>282</Country_Seq_Id>
    </ShipToCustomer>
  </SubmitShipToAccountWithError>
</soap:Body>
</soap:Envelope>

```

2. Get ShipMethods values.

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <SecurityHeader xmlns="http://tempuri.org/AnchorWebservice/AnchorWebservice">
      <Username>YOUR BILL TO ID</Username>
      <Password>YOUR PASSWORD</Password>
    </SecurityHeader>
  </soap:Header>
  <soap:Body>
    <GetShipMethods xmlns="http://tempuri.org/AnchorWebservice/AnchorWebservice">
    </GetShipMethods>
  </soap:Body>
</soap:Envelope>

```

3. Submit order

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

```

```

    <soap:Header>
      <SecurityHeader xmlns="http://tempuri.org/AnchorWebservice/
AnchorWebservice">
        <Username>YOUR BILL TO ID</Username>
        <Password>Your Password here</Password>
      </SecurityHeader>
    </soap:Header>
    <soap:Body>
      <SubmitOrderWithImprint xmlns="http://tempuri.org/AnchorWebservice/
AnchorWebservice">>
        <SOI>
          <Invoice_Seq_Id></Invoice_Seq_Id>
          <Bill_to_Seq_Id>YOUR BILL TO ID</Bill_to_Seq_Id>
          <Ship_to_Seq_Id>Your Password here</Ship_to_Seq_Id>
          <!--Optional:-->
          <PO_Number>TEST IMPR</PO_Number>
          <Net>15</Net>
          <!--Optional:-->
          <Flag_Rush_Order>N</Flag_Rush_Order>
          <!--Optional:-->
          <Date_Ship_By>09-DEC-2018</Date_Ship_By>
          <Shipping_Charge>5</Shipping_Charge>
          <!--Optional:-->
          <SO_Detail>
            <!--Zero or more repetitions:-->
            <SalesOrderDetailImprint>
              <Product_Seq_Id>638118</Product_Seq_Id>
              <Order_Quantity>1</Order_Quantity>
              <Ship_Quantity>1</Ship_Quantity>
              <Unit_Price>12</Unit_Price>
              <Discount>0</Discount>
              <Extension>0</Extension>
              <Customer_ID>YOUR BILL TO ID</Customer_ID>
              <imprint_font_style_id>1</imprint_font_style_id>
              <imprint_text_line1>Line1 TEST</imprint_text_line1>
              <imprint_text_line2>Line2 test</imprint_text_line2>
              <indexing_color_id>1</indexing_color_id>
            </SalesOrderDetailImprint>
          </SO_Detail>
          <!--Optional:-->
          <Flag_All_Complete>N</Flag_All_Complete>
            <Ship_method_Seq_Id>8</Ship_method_Seq_Id>
            <Store_Name>Eileen Anne</Store_Name>
            <Store_Message>test</Store_Message>
            <Store_Street>2121 ALLEN PRKWY APT 4073</Store_Street>
            <Store_City>HOUSTON</Store_City>
            <Store_State>TX</Store_State>

```

```

        <Store_ZIP>77019</Store_ZIP>
        <Store_Country>United States</Store_Country>
        <Intl_Tax_Number>0</Intl_Tax_Number>
        <Intl_Tax_Description>test desc</Intl_Tax_Description>
        <Intl_Tax_Amount>0</Intl_Tax_Amount>
        <Special_Instruction></Special_Instruction>
        <Date_Shipped>25-DEC-2018</Date_Shipped>
    </SOI>
    <!--Optional:-->
    <sErrorCode></sErrorCode>
</SubmitOrderWithImprint>
</soap:Body>
</soap:Envelope>

```

2.2.8 Minimum Functions

Here's the Minimum Functions that you will need to use for placing an order, once you populate your database with our items, plus all the lookup necessary.

1. Create shopping cart on your website for customers to start adding items.

Once they are ready to checkout, here's the things you can do:

- a. Access our webservices for shipping rate (Flat or Rated) -> you can populate a dropdown box or radio button.

[GetVariousFlatRates](#)

Return Various FLAT shipping Rates, given total number of items, weight(use 1 if N/A), zipcode(use blank space if N/A), and country code ID

[GetVariousShipRates](#)

Return Various Ship Rates, given weight, zipcode, and country code

- b. Create the customer account on our end

[SubmitShipToAccount](#)

Return Integer, given a ShiptoCustomer Structure as input.

Return the Customer Ship To Account number.

[SubmitShipToAccountWithError](#)

Return Integer, given a ShiptoCustomer Structure as input.

Return the Customer Ship To Account number. THIS ONE HAVE ERROR Parameter

- c. Check item availability upon customer checkout to reduce the possibility of NO STOCK.

[GetItemAvailable](#)

Return ListValue3 Structure of Item Seq Id, Unit Available, Item_Status_Code, Item_Status_Description given an array of Item Seq Id.

Useful when querying Item(s) availability.

[GetItemAvailablebyTime](#)

Return ListValue3 Structure of Item Seq Id, Unit Available, Item_Status_Code, Item_Status_Description given number of Minutes since Inventory is updated.

Useful to get Item availability based on time. A Maximum of 120 minutes allowed

- d. Submit the order to us.

[SubmitOrderWithStoreandError](#)

Return Boolean and Error(if applicable), given SalesOrder Structure and a String as input.

Return TRUE if successful.

Note: 4Th party Store Information(Name, Street, Address, etc) MUST be submitted

2. End of day (usually 7:30PM EST) query status of orders via webservises
(for tracking numbers, and check for out of stock items so you can handle the refund to your customer).

[GetOrderDetails](#)

Return SalesOrder Structure, given an Invoice Seq ID. Returns Order information and items.
Useful to check if any items is unfortunately will not shipped due to out of stock or backordered.

[GetOrderStatus](#)

Return SalesOrderStatus Structure, given an Invoice Seq Id input.
Useful to see the Status of Sales Order, Tracking number(s) is included for Shipped Invoice.

[GetTrackingNumbersURL](#)

Return URL with tracking Numbers, given an Invoice number

Note: you must check the

3. These are optional, if you want to see orders within date ranges.

[GetProcessedInvoices](#)

Return Integer of Invoice Seq Id, given an Account Number(bill to), Date Start and Date end.
Returns Array of Shipped Invoice(s) for specific date_shipped range

[GetProcessedOrders](#)

Return ProcessedOrder Structure, given an Account Number(bill to), Date Start and Date end.
Returns Array of Processed Invoice(s) and tracking number(s) for specific date range

2.3 Initial database

Enter topic text here.

2.3.1 Files

A Tab delimited file "Webservice.txt" needs to be imported to your own database.

---- You can also request Webservice file with header information on the first row (WebserviceLegend.txt)----

This file consist of these fields:

ITEM ID --> This must be submitted as the item id for orders

SPEEDY CODE

ISBN

TITLE

AUTHOR

PRICE

UPC

WEIGHT

DISCOUNT

NEW FLAG

DATE_FIRST_SALE

VENDOR_CODE
 BINDING_TYPE_CODE
 CATEGORY
 AVAILABLE
 ITEM_RANK
 ONHAND
 ONORDER
 VENDOR_STOCK_NUMBER
 BIBLE_BINDING
 BIBLE_CLOSURE
 BIBLE_COLOR
 BIBLE_FLAG_CONCORDANCE
 BIBLE_FLAG_GIANT
 BIBLE_FLAG_INDEXED
 BIBLE_FLAG_RED_LETTER
 BIBLE_POINT_SIZE
 BIBLE_STYLE
 BIBLE_VERSION_SHORT_NAME
 FULLSIZE_COVER_URL
 FULL_DUE_DATE
 PRIMARY_CBC
 SUB_CBC
 SUPER_CBC
 EAN
 ITEM_TYPE_ID
 CATALOG_SECTION_ID --> This is also known as Product Category Codes
 FLAG_DISCONTINUED
 FLAG_SPECIAL_ORDER_ITEM
 PUBLISHER_CODE
 ITEM_STATUS_TYPE
 BISAC_CATEGORY
 FLAG_ENGRAVABLE
 FLAG_IMPRINTABLE
 FLAG_INDEXABLE

Beside the initial Tab delimited file, there are lookup file for the Binding Codes, Category Codes, Vendor Lookup, and Misc Lookup. (vendor and publisher use the same lookup).
 ie: item ID 19, Publisher code is WHIT, if you look at the vendor lookup file, it is "Whitaker House".
 You don't need the 4 lookup files, but at least you will need the Binding code. This Binding code, Category Codes can be invoked via webservice function also.
 this is the preferred method since the function will always return the latest binding code available.
 Below is the lookup table function calls:

[GetProductBindings](#)

Return an Array of ListValue2 Structure with Binding SEQ ID, Binding Description, and Binding Code.

[GetProductCategories](#)

Return an Array of ListValue Structure with Category SEQ ID and Category Description.

[GetProductPublishers](#)

Return an Array of ListValue2 Structure with Publisher SEQ ID, Publisher Name, and Total Number of Titles for that Publisher.

[GetProductSubjects_Complete](#)

Return an Array of ListValue2 Structure with Subject SEQ ID, Subject Code, and Subject Description.

Your database should be updated periodically by either re-importing this initial file (which is updated daily), or by calling a webservice function. (this will at least update the current quantity available). For new products, you have to re-import this file, which we can upload automatically each night to your FTP server.