Introduction:

We have implemented a set of web services that allow interaction between our backend system and our customers'.

High-level overview of available functionality:

- Placing orders, including invoiced orders
- Querying open or shipped orders
- Specific order status/details
- Current inventory list
- Specific product search & details
- Setting and retrieving "BOL Comments" to be displayed on the BOL

A note on programming languages:

Our web services are programmed in the Microsoft .Net framework. For maximum compatibility and greatest ease of use, we recommend any development based on our web services to use .Net if possible. Other languages may be used but will not fully be supported. When using our web services without .Net, you will be interacting with the service in XML format. In this case we ask that all development adhere to the extensibility of XML so that we may continue to add additional information about orders or inventory when we feel it would be of benefit to our customers. For those customers who will be using .Net, we have a sample application available from which code may be studied or copied. For XML users, we will be able to provide support based only on the actual XML exchange and not code.

Methods provided by the web service:

public Order[] getOpenOrders()

Returns the list of all open orders with basic information such as destination address. To retrieve a list of items for each order, a call to getOrder will be required.

public void placeOrder2_0(ref Order2_0 pOrder, ref List<OrderItem2_0> pItems, ref List<InvoiceMiscCharge> pMiscCharges)

Places an order, optionally with an invoice. We highly recommend use of this method over our original placeOrder method, even if invoice support is not desired. There is no return value for this method. If an error occurs an exception will be thrown. This method does return the objects back to you in which the status may be checked, or you can use the checkOrderStatus method to verify a successful order . This method has the following constraints:

- BOL_Num (shipper number) is required and must be no more than 13 characters including the prefix (e.g. WCD). Your customer prefix will be added automatically by the service if it isn't included, and will still count against the character limit.
- Carrier name is required and must be no more than 40 characters in length.
- Requested ship date is required and must have a value of the current date or later.
- Shipto_address1 is required and must be no more than 30 characters in length. Shipto_address2 and 3 are optional.
- Shipto_city, optional, has a maximum of 15 characters in length. Typically this will be used.
- Shipto_state, optional, may not be more than 2 characters in length. Typically this will be used.

- Shipto_zip, optional, may not be more than 5 characters in length. Typically this will be used.
- For invoiced orders (create_invoice = true), please also fill in the “Soldto” address fields with the billing address.
- Special instructions, optional, have a maximum of 55 characters per field.
- Customer PO Number, optional, has a maximum length of 20 characters.
- Invoice payment terms is required for invoiced orders and has a maximum length of 20 characters.
- Invoice external license number is optional for invoiced orders and has a maximum length of 20 characters.
- Total_invoice_amount must equal the total amount of all items and misc charges/credits when placing an invoice.
- Date_invoice_due is also required for invoices.
- Order must contain at least one item. Misc charges/credits are optional for invoices and ignored for non-invoiced orders. For .Net, pass a null value for misc charges as required.
- The order will be considered prepaid when the Prepaid_shipping field is true, and collect otherwise.
- For each item, the required fields are CodeNumber, Cases, and Bottles. For invoices, CasePrice and BottlePrice are also required. Note that CasePrice and BottlePrice are unit prices. CasePrice is the price of one case and BottlePrice of one bottle. Also use TransactionType (tax status e.g. TR or TAXPAID, BR or TTBBOND, CR or CUSTOMSBOND) when known/required. For our Western Wine Services customers, please note that TransactionType is always required.
- Invoice_form_num, optional, may be used to specify an alternate invoice form for some customers. Use this field only when certain that your account has multiple invoice forms available. Omit it if unsure.
- The optional flag Save_order_only will save an order to our database without placing it. This order can then be later modified on our interactive website. Saved orders cannot be loaded or placed through web services. The order number may still be used with web services as long as the saved order is not placed via the website. If the order number is reused via web services, then the saved order will require a new order number.
- All date objects should be standard .Net System.DateTime objects or otherwise a string conforming to ISO 8601.

public bool placeOrder(ref OrderDetail pOrder) [obsolete]

Places an order provided all the fields in the OrderDetail object are passed properly. This is the original method which is still supported but it does not support invoices or other new features. It is not recommended for use in new development, whether invoice support is desired or not. This method returns a boolean that represents success in placing the order, however, errors in placing the order will throw an exception anyway. This method has the following limitations and constraints:

- Invoiced Orders may not be placed with this method.
- ShipperNo (BOL number) must be present and no more than 13 characters including the prefix (e.g. WCD)
- The order must contain at least one item. Use only the item CodeNumber, Cases and Bottles fields as well as Transaction Type when known/required.
- PO Number, if used, has a maximum length of 20 characters. This field was previously required but is now optional.
- PrepaidCollect is a required string that should consist of “Prepaid” or “P” or “Collect” or “C”
- RequestedShipDate is a string that should be in “yyyy/MM/dd”. Recent versions also accept dashes. The date must be the current day or later.
- EnteredDate and ActualShipDate should be ignored as they will be set automatically.
- The To field is the recipient of the shipment. It is required and has a maximum length of 30 characters.
- The Name2 field is an optional second line with a maximum length of 30 characters.
- Street is the street address and it has a maximum length of 30 characters.
- State, if used, has a maximum length of 2 characters.
- Comments (AKA “Special Instructions”) fields are optional and have a maximum of 55 characters each.

public InventoryItems getInventory(DateTime asOfDate)

Returns a list of products with the available number of cases and bottles. The list does NOT include products with zero quantity of cases and bottles; therefore, if a product is not in the list, you should assume that you don't have that item in stock.

Note that while there are quantity fields returned with this function, these do not take into account all factors such as allocated product. If you need detailed availability reports, you must use the getProductDetail function on each item specifically.

public ProductDetail getProductDetail(string pProductCode)

Returns information for a product, such as # of duty paid cases and bottles, # of inbound cases and bottles, description, size, etc. This is the function to use if you need detailed inventory status for an item. It returns detailed item availability and allocation info. If using this function to get inventory numbers for a list of products, please query only one item at a time.

public OrderDetail2_0 getOrder2_0(string pOrderNumber)

Returns the order detail based on the order number passed as parameter. This function will return any order, invoiced or not, with any items, and in the case of invoices, also any misc charges on the order. This function will also return delivery status information and any BOL comments if present. This function will not return the actual invoice or BOL document itself.

public OrderDetail getOrder(string pOrderNumber) [obsolete]

Returns the order detail based on the order number passed as parameter. This function will continue to be supported but should not be used for new development. This function will not provide invoice information or misc charges for invoices. It will return information for invoiced orders, but it will be incomplete. Please use getOrder2_0 instead.

public Order[] getShippedOrders(DateTime pFrom, DateTime pTo)

Returns the list of shipped orders in a maximum of 1 month time span and up to 6 months in the past. All date objects should be standard .Net System.DateTime objects or otherwise a string conforming to ISO 8601.

public Order[] checkOrderStatus (ref List<Order> pOrders)

Returns the status (OPEN,SHIPPED,ON_HOLD) of a list of orders. These status types may additionally be followed by a dash and more information e.g. ON_HOLD-DATE. Note that while an Order object is returned for each order, only the OrderNumber and Status fields should be checked. Please handle both an item missing from the returned list OR an item returned with status NOT_FOUND as an order not being found.

List of objects used:

The SecurityHeader object must contain the Authentication Key that has been assigned to your company. The key can be obtained by contacting us at nj_support@westerncarriers.com.

All objects use the XML/SOAP namespace https://www3.westerncarriers.com/Webservices.

```
public class SecurityHeader : SoapHeader {
        public string AuthenticationKey;
  }
```

```
public class OrderDetail {
        public OrderHeader OrderHeader;
        public OrderItem[] Items;
}


public class Order2_0 {
   //Note: This class has several fields for which input while placing an order is ignored, e.g. From address, customer
  //    number, status, source_code, western_division.
        public string Bol_num;                      //Required
        public string Carrier;                      //Required
        public bool Create_invoice;                 //Required for all orders. False if not an invoiced order.
        public string Customer_po;                  //Optional
        public DateTime? Date_invoice_due;          //Required for invoiced orders, not used in non-invoiced orders
        public DateTime Date_ship_requested;        //Required
        public string External_license;             //Optional for invoiced orders, not used in non-invoiced orders
        public string Payment_terms;                //Required for invoiced orders, not used in non-invoiced orders
        public bool Prepaid_shipping;               //Required
        public bool  Cod                            //Optional: Cash on delivery
        public string Shipto_address1;              //Required
        public string Shipto_address2;              //Optional
        public string Shipto_address3;              //Optional
        public string Shipto_city;                  //Optional for NJ accounts, Required for CA, typically used
        public string Shipto_state;                 //Optional for NJ accounts, Required for CA, typically used
        public string Shipto_zip;                   //Optional for NJ accounts, Required for CA, typically used
        public string Soldto_address1;              //Required for invoiced orders, not used in non-invoiced orders
        public string Soldto_address2;              //Optional for invoiced orders, not used in non-invoiced orders
        public string Soldto_address3;              //Optional for invoiced orders, not used in non-invoiced orders
        public string Soldto_city;                  //Required for CA invoiced orders, optional for NJ invoiced orders
        public string Soldto_state;                 //Required for CA invoiced orders, optional for NJ invoiced orders
        public string Soldto_zip;                   //Required for CA invoiced orders, optional for NJ invoiced orders
        public string Special_instructions;         //Optional
        public string Special_instructions2;        //Optional
        public int? Invoice_form_num;               //Optional for invoiced orders, but only use if specifically advised
        public decimal? Total_invoice_amount        //Required for invoiced orders, not used in non-invoiced orders
        public bool? Save_order_only; //If true, order will not be placed immediately and will be saved for use with our website.
}

public class OrderInfo2_0 : Order2_0 {
        public DateTime Date_entered;
        public DateTime? Date_shipped; //Nullable DateTime, no value = not shipped
}

public class Order {
        public string EnteredDate;
        public string RequestedShipDate;
```

```
    public string ActualShipDate;
    public string Status;
    public string Destination;
    public string City;
    public string State;
    public string OrderNumber;
    public int TotalCases;
    public int TotalBottles;
    public string Carrier;
    public string HoldStatus; //null or zero-length string = not on hold
}
public class OrderHeader : Order {
    public string Street;
    public string PONumber;
    public string Zip;
    public string To;
    public string Name2;
    public string Comments1; //optional, no more than 55 characters
    public string Comments2; //optional, no more than 55 characters
    public string EnteredBy;
    public string PrepaidCollect;
}

public class Product {
    public string CodeNumber;
    public string Description;
    public string Size;
    public int BottlesPerCase;
    public long Vintage;
}

public class OrderItem : Product {
    public string Lot;
    public int Cases;
    public int Bottles;
    public int Weight;
    public string TransactionType; //tax status e.g. TR, BR, or CR. This is required for CA.
}

public class OrderItem2_0 : OrderItem {
    public decimal? BottlePrice;
    public decimal? CasePrice;
}


public class ProductDetail : Product {
    public int DutyPaidCases;
    public int DutyPaidBottles;
    public int InBondCases;
```

```
        public int InBondBottles;
        public int AllocatedCases;
        public int AllocatedBottles;
}

public class InventoryItem : Product {
        public int Cases;
        public int Bottles;
        public int Weight;
}



public class BOLComments {
        public string BOLNum;
        public string[] EndComments; // Comments at the end of the BOL
        public string[] HeadComments; // Comments at the top of the BOL
        public bolcomdet[] lines; //Index is the item line number of the order (not the item code)
}

public class bolcomdet {
        //Externally this class basically is just an ArrayList that can contain
        //  multiple comments per line (string for line comment, double or decimal for price comment)
        public object[] data;
}

public class RtsDeliveryInfo {
        public string RouteId;
        public int MinutesDelayed;
        public bool OrderIsDelivered;
        public DateTime ActualArrivalDateAndTime;
        public DateTime ActualDepartureDateAndTime;
}
```

Web Service Exceptions

Western Carriers exceptions are broken up into several types with associated codes for each group. The groups are: (M)issing Data, (B)ad Data, (G)eneral, and (A)uthorization errors. Inside of the data error groups, they are further broken down into the object type: (O)rder, (P)roduct, (C)ontainer, (I)nvoice, (M)iscellaneous. The error codes that are returned are in the format of: First Character is the Group Code, Second Character is Object Type, then a three-digit error number. The error number is followed by a pipe character (|) Followed by a description. The pipe character is included so you can programmatically split out the error code from the user readable description.
These error codes are subject to additions in the future and are not intended to be a permanent and complete listing. Please code accordingly and include code to handle additional unknown error codes that are not listed here. The user readable description is subject to change in wording at any time and for any reason and should not be relied on for programmatic comparison.

General Errors:
G0001|Generic Global Exception
G0002|Not Implemented Exception

Authorization Errors:
AM001|SecurityException: The Authentication Key is missing.
AM002|SecurityException: Unable to read authentication key.
AM003|SecurityException: Invalid configuration for given authentication key.

Missing Data Errors:
MO001|The order must contain at least one item.
MO002|The order number was not specified.
MO003|The prepaid or collect specification was not specified.
MO004|The carrier was not specified.
MO005|The address was not specified.
MO006|The requested ship date was not specified.

MI005|The sold to address was not specified.
MI007|The total amount was not specified.
MI008|The invoice due date was not specified.
MI009|The payment terms were not specified.

Bad Data Errors:

BO001|Maximum timespan allowed is 31 days, up to a maximum of 6 months in the past.
BO002|The order number (including prefix) must be no longer than 13 characters.
BO003|The PO Number must be no longer than 20 characters.
BO004|The carrier must be no longer than 40 characters.
BO005|The address must be no longer than 30 characters.
BO006|The address line 2, if used, must be no longer than 30 characters.
BO007|The requested ship date cannot be prior to today.
BO008|The state specified is invalid.
BO009|The special instructions (comments) line 1 must be no longer than 55 characters.
BO010|The special instructions (comments) line 2 must be no longer than 55 characters.
BO011|The address line 3, if used, must be no longer than 30 characters.
BO012|The city specified must be no longer than 15 characters.
BO013|The zip/postal code must be no longer than 5 characters.

BI005|The sold to address must be no longer than 30 characters.
BI006|The sold to address line 2, if used, must be no longer than 30 characters.
BI008|The sold to state specified is invalid.
BI011|The sold to address line 3, if used, must be no longer than 30 characters.
BI012|The sold to city specified must be no longer than 15 characters.
BI013|The sold to zip/postal code must be no longer than 5 characters.
BI014|The payment terms must be no longer than 20 characters.
BI015|The external license number must be no longer than 20 characters.

BP001|Product not found.
BP002|Product (___)o contains a zero quantity of cases and bottles.
BP003|Product has cases but no case price (invoiced orders only)
BP004|Product has bottles but no bottle price (invoiced orders only)
BP005|Transaction type for product not recognized