

Protocol for XML Web control interface

Prerequisites for the operation: Module
WebConnect

The Veloce server has an open TCPIP port for sending queries regarding customers and orders to move automatically. To do this simply to connect to this port and send requests to the XML format as described below. The port number Port is configurable in module WebConnect

Online order application (Step 1):

In the customer section there are 2 possibilities:

1- The website we mailings a customer number: If there is a customer number (The customer number is provided by Veloce when creating a new folder, it could be kept in mind in the web client plug), Veloce charge the corresponding customer record and if other fields provided, will Veloce update the client record as the new information provided in these fields.

2 - The website does not mailings customer number: If there have no issue card, then the phone number, name, house number and apartment will be used to locate a particular customer in the customer file Veloce. If there is no apartment, this box can be omitted, but not the other 3.

Should the customer does not exist (new client), it will be created and the new number client will be returned in the response.

```

<VeloceRequest>
  <Oper> 1 </ Oper> // 1 = Ordering (allows the same
                      blow to update or create a
                      customer file.

  <Customer>
    <Record> 12 </ Record> // If there are a number here, it will be used
                            instead of looking for Phone #
                            and the file will be updated as needed
    <Name> Resto Finance Inc. </ Name> // If it does not provide number, #
                                      such, Name, Street, House number and
                                      apartment will be used to
                                      trace the customer record

    <Tel> 4504362192 </ Tel> // Phone Number
    <StNumber> 700 </ StNumber> // Door number

```

<Street> Lamontagne </ Street> // Street name (24 characters max)
 <StreetCode> Lamo </ StreetCode> // Code of the street (9 characters)
 // If a street code is present, the name
 Street // is ignored
 <Apt> </ Apt> // Apartment (4 characters)
 <Near> </ Near> // Optional Near (24 characters)
 <City> St-Jerome </ City> // Optional City (24 characters)
 <Postal 1> J5L1V1 </ Postal> // Optional Postal code (6 characters)
 <Field # 1> Go backward </ Field # 1> // Optional programmable Champs
 (24 characters)

Page 2

<Field # 2> </ Field # 2> // Optional programmable Champs (24
 characters max)
 <Field # 3> </ Field # 3> // Optional programmable Champs (24
 characters max)
 <Special # 1> Jester </ Special # 1> // Optional Fields that special
 company does not appear on the invoice (24
 characters)
 <Special # 2> </ Special # 2> // Optional special fields
 company does not appear on the invoice (24
 characters)
 </ Client>
 <Order>
 <Type> takeout </ Type> // Type of control (Takeout or Delivery)
 <ExpectedTime> HH: MM </ ExpectedTime> // Optional desired Hour
 order, if not to control
 for now
 <Seat> // Can be repeated multiple times
 <Item> // Maximum 200 to 200 repetitions
 items per "seat"
 <Product> 1 </ Product> //Product number
 <Quantity> 1 </ Quantity> // Product Quantity
 <UnitPrice> 2.95 </ UnitPrice> // Optional: Unit Award
 product. It is preferable to
 the register so there will be no
 misunderstanding between the price and the web
 Veloce price for a product.
 <Text> Post </ Text> // Optional: Text message-will
 used only if an item of
 Veloce message type is specified
 (19 characters)
 </ Item>
 <Invoice> // Optional: To payment
 Automatic invoice, one per customer
 (seat).
 <TAX1> 0.15 </ TAX1> // Optional: Amount of tax Forced
 overriding the automatic calculation
 tax per Veloce. Ensures
 corresponding to what the customer sees on
 Web and invoice it will receive.
 <Tax2> 0.26 </ Tax2>
 <Payment> // Maximum of 5 repetitions for 5
 Payments
 <Digital> 1 </ Digital> // # Veloce payment method
 <Amount> 3.36 </ Amount> // Amount of payment

```

    </ Invoice></ Payment>
  </ Seat>
</ Order>
</ VeloceRequest>

```

Page 3

Veloce server Response to this request:

```

<VeloceResponse>
  <Oper> 1 </ Oper>
  <Result> 0 </ Result>

```

in Item

```

// 0 = Success
// 1 = Error type not specified command
// 2 = Internal error
// 3 = Error: Match with multiple customer
// 4 = Error: AT least one product is non-existent
// 5 = Error: missing data fields

// 6 = Error: No item on the command
// 8 = Error has internally when trying to
place the order (can hardly
happen if the system have too much control
waiting (900))
// 9 = Error: Error with client record
(Normally that would arrive if the file
Customer is full (more than 1 million customer)
// 10 = Command accepted but possible problem
with the impression thereof, it should be
call the store to confirm
control in this case
// 11 = Invalid street Code
// 12 = found street Code

```

```

<ResultMsg> Order posted </ ResultMsg>
<ClientRecord> 12 </ ClientRecord>

```

```

// Error Message or success
// If this is the # of the file
Veloce client, it would be useful for
the web service to store this number to
use in a future
order for this customer

```

```

</ VeloceResponse> [EOT]

```

Veloce query request of the client file: (Step 2)

If the search criteria are not precise enough it is possible that

multiple client file be returned (to the same customer several phone # for example)

```
<VeloceRequest>
  <Oper> 2 </ Oper>
  <Record> 12 </ Record> // Optional
  <Tel> 4504362192 </ Tel> // Used if you do not use the Record
  <Name> Resto Finance Inc. </ Name> // Optional to refine search
  <StNumber> 700 </ StNumber> // Optional to refine search
  <Street> Lamontagne </ Street> // Optional to refine search
  <Apt> </ Apt> // Optional to refine search
</ VeloceRequest>
```

Page 4

Veloce server Response to this request:

```
<VeloceResponse>
  <Oper> 2 </ Oper>
  <Result> 1 </ Result> // 0 = nonexistent Client
                        // 1 = Success
                        // 2 = Internal error
                        // 3 = Invalid street Code
                        // 4 = street code not found

  <Customer> // Can be repeated multiple times
    <Record> 12 </ Record> // Record number of that customer
    <Name> Resto Finance Inc. </ Name> //Client Name
    <Tel> 4504362192 </ Tel> // Phone Number
    <StNumber> 700 </ StNumber> // Door number
    <Street> Lamontagne </ Street> // Name of the street
    <StreetCode> Lamo </ StreetCode> // Street Code
    <Apt> </ Apt> // Apartment
    <Near> </ Near> //Close to
    <City> St-Jerome </ City> //City
    <Postal> J5L1V1 </ Postal> //Postal code
    <Field # 1> Go backward </ Field # 1> // Programmable Champs
    <Field # 2> </ Field # 2> // Programmable Champs
    <Field # 3> </ Field # 3> // Programmable Champs
    <Special # 1> Jester </ Special # 1> // Special Fields that company does not appear
                                        the bill
    <Special # 2> </ Special # 2> // Special Fields that company does not appear
                                        the bill

  </ Client>
</ VeloceResponse> [EOT]
```

EXAMPLE

delivery control for a specific time

-with payment

-Amount forced to avoid tax difference from the method of rounding

Veloce

-A Items with forced unit price (can be used to avoid a difference with prices programmed into Veloce)

-A Items with a text message (requires a corresponding message in item Veloce course)

```
<? xml version = "1.0" encoding = "ISO-8859-1"?>
<VeloceRequest>
  <Oper> 1 </ Oper>
  <Customer>
    <Tel> 6696862 </ Tel>
    <Name> Web Test </ Name>
    <StNumber> 123 </ StNumber>
    <Street> Street XYZ </ Street>
    <Special # 1> Simple Test </ Special # 1>
  </ Client>
  <Order>
    <Type> Delivery </ Type>
    <ExpectedTime> 4:25 p.m. </ ExpectedTime>
    <Seat>
      <Item>
        <Product> 1 </ Product>
        <Quantity> 1 </ Quantity>
        <UnitPrice> 1.23 </ UnitPrice>
      </ Item>
      <Item>
        <Product> 2 </ Product>
```

```
<Quantity> 3 </ Quantity>
</ Item>
<Item>
<Product> 999 </ Product>
<Quantity> 1 </ Quantity>
<Text> Message </ Text>
</ Item>
<Invoice>
<TAX1> 1.41 </ TAX1>
<Tax2> 2.22 </ Tax2>
  <Payment>
    <Media> 2 </ Digital>
    <Amount> 31.83 </ Amount>
  </ Payment>
</ Invoice>
</ Seat>
</ Order>
</ VeloceRequest>
```

** Responce of veloce:

```
<? xml version = "1.0" encoding = "UTF-8"?>
<VeloceResponse>
  <Oper> 1 </ Oper>
  <Result> 0 </ Result>
  <ResultMsg> Order posted </ ResultMsg>
```

Page 6

```
<ClientRecord> 12 </ ClientRecord>
<OrderNumber> 22 </ OrderNumber>
</ VeloceResponse> [EOT]
```

Result - Print Veloce

Request for information is "online / offline" from the cashier who will print the bill: (Operation # 3)

```
<VeloceRequest>  
  <Oper> 3 </ Oper>  
</ VeloceRequest>
```

```
<VeloceResponse>  
  <Oper> 3 </ Oper>  
  <Result> 1 </ Result>
```

not

happen)

happen)

```
</ VeloceResponse> [EOT]
```

```
// 0 = OnLine  
// 1 = Internal error (file access error,  
//           should never happen in principle)  
// 2 = OffLine  
// 3 = internal error (should never  
  
// 4 = Internal Timeout (should never  
  
// 999 = Error: missing Control module
```

Request to find out the status of an order: (Operation # 4)

<VeloceRequest>

<Oper> 4 </ Oper>

<OrderNumber> 2 </ OrderNumber>

returned

<OrderType> Delivery </ OrderType>

because the same
between a mode
other.

</ VeloceRequest>

// Order number as

by webconnect

// Type of drive (or takeout

Delivery) very important

number can be repeated

control and

<VeloceResponse>

<Oper> 4 </ Oper>

<Result> 2 </ Result>

// 0 = Command not found

// 1 = Command has not printed invoice

// 2 = Command presents printed invoice

// 3 = Command from ready (delivery)
// 4 = control part (delivery)
// 10 = Internal error (Never Should

happen)

// 11 = Value of the request absent OrderNumber
// 12 = Value of the request absentee OrderType
// 999 = Error: missing Control module
// Optional depending Result Time.

<Time> 1:45:01 p.m. </ Time>

</ VeloceResponse> [EOT]

Page 9

Update query menu (Step 5)

This request may not return the entire list in one operation
it will then proceed to subsequent calls of this request as described below.

<VeloceRequest>

<Oper> 5 </ Oper>

<Number> 1 </ Number>

// Specific item number

or

<StartFrom> 1 </ StartFrom>

// Offset from which the list

begins (1 = first item of

the list)

</ VeloceRequest>

```

<VeloceResponse>
  <Oper> 5 </ Oper>
  <Result> 1 </ Result>                                // 0 = No item in the requested list
                                                         (no item list follows)
                                                         // 1 = transmitted item list
                                                         // 999 = Error: missing Control module
<Item>                                                 // Can be repeated as many times as
                                                         there are items in the list
  <Number> 25 </ Number>                                // Item Number
  <Description> SOUP </ description>                    // Item description
  <Price>
    <Mode # 1>                                         // We still have 4 modes
      <Level # 1> 2.29 </ Level # 1> // You can have up to 4
      ...                                              // Price level
      <Level # 4> 2.29 </ Level # 4>
    </ Fashion # 1>
    ...
    <# 4 mode>
      <Level # 1> 2.29 </ Level # 1>
      ...
      <Level # 4> 2.29 </ Level # 4>
    </ Fashion # 4>
  </ Price>                                             // Item's price
  <TAX1> 1 </ TAX1>                                    // Taxable (TAX1, tax2 ...)
                                                         0 = Non taxable 1 = Taxable
  ...
  <Tax5> 1 </ Tax5>
  <Sector # 1> // info by Sector (central Corporate alone.)
    <Price> 2.29 </ Price>
      <Mode # 1>                                       // We still have 4 modes
        <Level # 1> 2.29 </ Level # 1> // You can have up to 4
        ...                                           // Price level
        <Level # 4> 2.29 </ Level # 4>
      </ Fashion # 1>
      ...
      <# 4 mode>
        <Level # 1> 2.29 </ Level # 1>
        ...
        <Level # 4> 2.29 </ Level # 4>
      </ Fashion # 4>
    </ Price>                                         // Item's price
    <TAX1> 1 </ TAX1>
    ...
    <Tax5> 1 </ Tax5>
  </ Sector # 1>
</ Item>

```

```
<Continue> No </ Continue> // To see if the list is complete or
petition // must be continued in another
// (List Continued)
// No: The list is complete
// [Numeric value] is the value that
must pass in the next call for this query parameter
<StartFrom> for following the list
// Note: the request must réappeler
as long as we do not get the value "No" here to obtain a list
Complete
</ VeloceResponse> [EOT]
```

Street add request or a street block (Step 6)

This request must not exceed 100 blocks at a time.

If we send two address block in sequence on the same street and the area and the shipping costs are identical, the block will Fusions by WebConnect automatically.

```

<VeloceRequest>
  <Oper> 6 </ Oper>
  <Street>
    Can be repeated several times
    <Code> damemtl </ code> Street Code
    <Clear> 1 </ Clear> Optional: If '1', deletes the existing street
    <StNumFrom> 202 </ StNumFrom> starting number
    <StNumTo> 438 </ StNumTo> End of number
    <Description> Boulevard ND </ description> Description
    <Zone> 4 </ Area> Delivery zone
    <Fee> 0.00 </ Fee> Shipping cost
  </ Street>
</ VeloceRequest>

<VeloceResponse>
  <Oper> 6 </ Oper>
  <Street>
    Can be repeated (once a street)
    <Code> damemtl </ code> Street code
    <Result> 0 </ Result> Result:
      0 = Success
      1 = Code and / or invalid description
      2 = File open streets
      3 = No. of max block. exceeded the Street
      10 = Internal error
    <ResultMsg> Success </ ResultMsg> Message of the result
  </ Street>
</ VeloceResponse>

```

Sample query 6:

Replacing the existing street and adding a 2nd address block
a different delivery area.

```
<VeloceRequest>
  <Oper> 6 </ Oper>
  <Street>
    <Code> damemtl </ code>
    <Clear> 1 </ Clear>
    <StNumFrom> 202 </ StNumFrom>
    <StNumTo> 438 </ StNumTo>
    <Description> Boulevard ND </ description>
    <Zone> 4 </ Area>
    <Fee> 0.00 </ Fee>
  </ Street>
  <Street>
    <Code> damemtl </ code>
    <StNumFrom> 439 </ StNumFrom>
    <StNumTo> 600 </ StNumTo>
    <Description> Boulevard ND </ description>
    <Zone> 5 </ Area>
    <Fee> 0.00 </ Fee>
  </ Street>
</ VeloceRequest>
```

```
<VeloceResponse>
  <Oper> 6 </ Oper>
  <Street>
    <Code> damemtl </ code>
    <Result> 0 </ Result>
    <ResultMsg> Success </ ResultMsg>
  </ Street>
  <Street>
    <Code> damemtl </ code>
    <Result> 0 </ Result>
    <ResultMsg> Success </ ResultMsg>
  </ Street>
</ VeloceResponse>
```

Serves to receive a street folder if it is present in veloce.

```
<VeloceRequest>
  <Oper> 7 </ Oper>
  <Code> damemtl </ code>          Street code
</ VeloceRequest>

<VeloceResponse>
  <Oper> 7 </ Oper>
  <Street>
    <Code> damemtl </ code>          Street code
    <Description> Boulevard ND </ description> Description of the road
    <Block>                          // Can repeat
      <StNumFrom> 202 </ StNumFrom>    // Start
      <StNumTo> 600 </ StNumTo>        // End
      <Zone> 4 </ Area>                // Zone
      <Fee> 0.00 </ Fee>               //Fresh
    </ Block>
    <Result> 0 </ Result>             Result:
                                      0 = Success
                                      1 = Invalid Code
                                      Rue found 2 =
                                      10 = Internal error
    <ResultMsg> Success </ ResultMsg> Message of the result
  </ Street>
</ VeloceResponse>
```

Example:

File consultation to an existing street with two blocks.

```
<VeloceRequest>
  <Oper> 7 </ Oper>
  <Code> damemtl </ code>
</ VeloceRequest>

<VeloceResponse>
  <Oper> 7 </ Oper>
  <Result> 0 </ Result>
  <ResultMsg> Success </ ResultMsg>
  <Street>
    <Code> damemtl </ code>
    <Description> Boulevard ND </ description>
    <Block>
      <StNumFrom> 202 </ StNumFrom>
      <StNumTo> 600 </ StNumTo>
      <Zone> 4 </ Area>
      <Fee> 0.00 </ Fee>
    </ Block>
    <Block>
      <StNumFrom> 601 </ StNumFrom>
      <StNumTo> 800 </ StNumTo>
      <Zone> 6 </ Area>
      <Fee> 0.00 </ Fee>
    </ Block>
  </ Street>
</ VeloceResponse>
```