

**Release**

**2.6**

CONTAINERWORLD FORWARDING SERVICES INC.

---

Advanced Solutions Group

Revised April 1<sup>st</sup>, 2015

xOrder  
Administration

ADVANCED SOLUTIONS GROUP

# **xOrder Administration Guide**

---

© 2012 ContainerWorld Forwarding Services Inc.  
16133 Blundell Road • Richmond • BC  
V6W 0A3  
Phone 604 276.1300 • Fax 604 276.1301  
[www.ContainerWorld.com](http://www.ContainerWorld.com)

---

# Table of Contents

<b>WHAT IS XORDER?</b> .....	<b>1</b>
XORDER JAVA SERVLET.....	2
XORDER APPLICATION.....	2
XORDER DIRECT TRANSMISSION .....	2
<b>INSTALLING XORDER</b> .....	<b>3</b>
CONFIGURING WEB START .....	4
<b>USING XORDER</b> .....	<b>5</b>
PROCESSING FILES .....	7
<b>XORDER DIRECT TRANSMISSION API</b> .....	<b>10</b>
LOGONSERVLET API.....	10
XORDERSERVLET API.....	11
TRANSMISSION RESULTS.....	12
<b>REPORTS &amp; CONFIRMATION</b> .....	<b>13</b>
TRANSACTION SUMMARY REPORT.....	13
REVERSE CONFIRMATION – INVENTORY SYNCHRONIZATION .....	14
<b>APPENDIX A: XML DOCUMENT STRUCTURE</b> .....	<b>16</b>
<b>APPENDIX B: XML DOCUMENT DEFINITIONS</b> .....	<b>19</b>
<b>APPENDIX C: XORDER SCHEMA (A)</b> .....	<b>22</b>
<b>APPENDIX C: XORDER SCHEMA (B)</b> .....	<b>23</b>
<b>APPENDIX C: XORDER SCHEMA (C)</b> .....	<b>24</b>
<b>APPENDIX C: XORDER SCHEMA (D)</b> .....	<b>24</b>
<b>APPENDIX C: XORDER SCHEMA (E)</b> .....	<b>26</b>
<b>APPENDIX D: XML DOCUMENT, SAMPLE A</b> .....	<b>27</b>
<b>APPENDIX D: XML DOCUMENT, SAMPLE B</b> .....	<b>28</b>
<b>APPENDIX E: DIRECT TRANSMISSION CODE SNIPPET</b> .....	<b>29</b>
<b>APPENDIX F: TESTING YOUR XML FILE</b> .....	<b>31</b>
<b>APPENDIX G: CONFIRMATION STRUCTURE</b> .....	<b>32</b>
<b>CUSTOMER SERVICE/TECHNICAL SUPPORT</b> .....	<b>33</b>

---

## What is xOrder?

*Faced with the fact that today's companies operate around custom systems yet share common data between partners, xOrder bridges the gap into a seamless, secure solution.*

As part of ContainerWorld's Business to Business(B2B) initiative the xOrder system has been developed to facilitate the transmission of orders into ContainerWorld's Product Distribution System via the internet. By using both a standardized template and interface inherent to XML (Extensible Markup Language) and Java along with the Internet, systems can confidently exchange information regardless of the operating environment or variations of data structure. XML (Extensible Markup Language) defines a universal standard for electronic data exchange and many software vendors have developed tools for creating XML documents. xOrder is ContainerWorld's custom software used to securely transmit and process XML documents created by their data exchange partners.

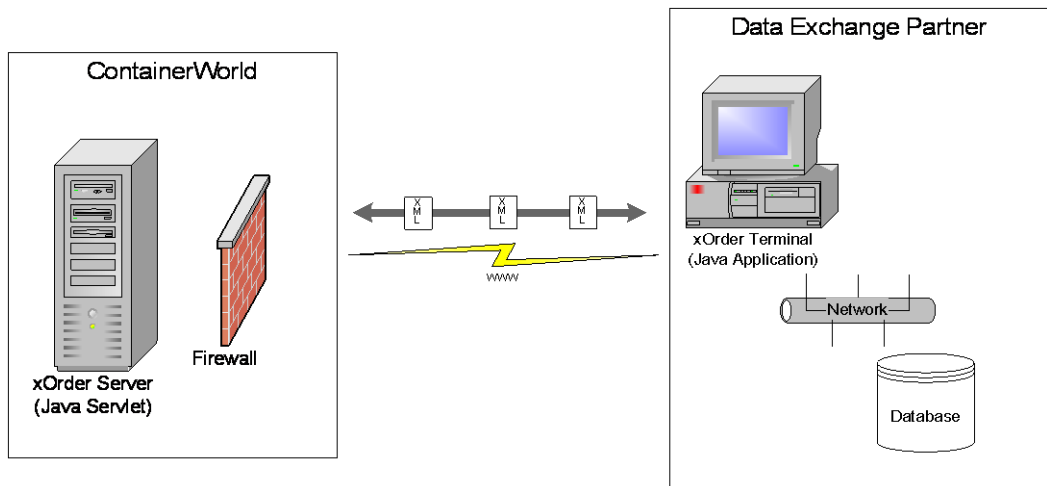


Figure 1: Technical layout of xOrder

## **xOrder Components**

### **xOrder Java Servlet**

The server side component of the xOrder system is a Java Servlet. This technology allows the use of an internet web server making the system very standardized and available 24/7. The Servlet is built using the Java Development Kit (JDK) 1.4, which provides increased functionality for XML handling. The java Servlet runs on ContainerWorld's internet server and will process the orders transmitted in an XML document format via the xOrder Application.

### **xOrder Application**

The client side component of the xOrder system is the xOrder application. The application runs on ContainerWorld's customer's P.C. using [Java™ Web Start](#). Java Web Start Technology will automatically insure you are running the latest version of the xOrder application by downloading any changes upon start up. The xOrder application is the user interface between the customer's system and ContainerWorld's system. xOrder will validate and transmit an XML document to ContainerWorld via the internet.

### **xOrder Direct Transmission**

As an alternative for customers who prefer to customize their systems to process orders without using the xOrder client application, they can take advantage of the xOrder web services, which provide an open API to login and transmit XML documents directly. This solution will create completely automated background integration for a seamless EDI link between two order management systems.

## Installing xOrder

*xOrder uses Java Web Start™ technology an innovative approach to launching applications. Every execution of the program verifies the version for updates as well as the integrity of code distribution. No longer do you need to worry about application updates.*

The installation process for xOrder is quick and easy, all you need is an internet connection and we'll do the rest! Our Advanced Solutions Group (ASG) decided to deploy everything for xOrder with one click from our website, [www.xOrder.ca](http://www.xOrder.ca). All you have to do is follow some simple directions and you will be up and running in minutes! <sup>1</sup>

### Java Web Start™

Java Web Start includes the security features of the Java™ 2 platform, so the integrity of your data and files is never compromised. In addition, Java Web Start technology enables you to use the latest Java 2 plug-in technology, eliminating any concerns of compatibility between different web browsers.

To use Java Web Start, you launch applications by clicking on a Web page link. If the application is not present on your computer, Java Web Start automatically downloads all necessary files. It then caches the files on your computer so the application is always ready to be re-launched anytime you want without having to re-download the program. Whether launched from an icon or a web link, the most current version of the application is always presented to you.

To install Web Start point your browser to <http://www.xOrder.ca>

On this page you will see a link stating, "Click here to download and install Web Start". When the dialog appears asking whether to run from the current location or save to disk, choose the run option. The download has a signed certificate from Sun Microsystems to ensure the application can be trusted. After following the installation instructions Web Start will be installed on your computer.

---

<sup>1</sup> Install times are estimated based on high-speed internet connections. Actual times will vary with connection speeds.

## Configuring Web Start

After successfully installing Java Web Start, the application needs to be configured to run xOrder. To do this, click the link in the xOrder web page labeled “click here to setup xOrder” or follow the link below:

<http://www.xOrder.ca/xOrder/jar/xOrder.jnlp>

Upon following the link, Web Start will download all of the required information to run xOrder. After Web Start configures itself the xOrder application will be launched.

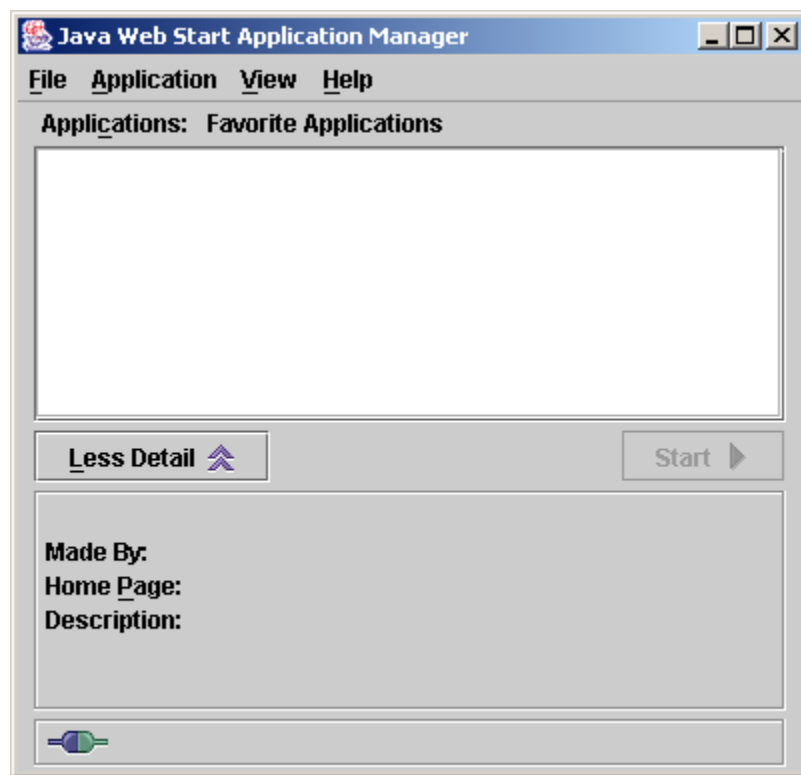


Figure 2: Java Web Start screen

## Using xOrder

*xOrder provides a simple, intuitive interface to the end user; just follow the steps. We've removed any unnecessary options and procedures to ensure that we save time rather than take time.*

Now that xOrder is installed and configured, users are ready to begin transmitting data. The first step, after you've created your XML file, is to logon to the xOrder system with the ContainerWorld supplied username and password. After successfully logging in, the extended features for xOrder are enabled. Figure 3 shows the two disabled menu items – 'View' and 'Reports'. These tools are password protected to ensure that no system settings are altered without permission and to customize the report data to your organization.



Figure 3: xOrder logon screen



After having logged in for the first time, the system preferences should be configured. These can be accessed from 'View >> Preferences' in the xOrder menu. Figure 4, below, shows the 'Directories' window for setting the default input data file directory and the default output file directory. The output directory is used to store system log files generated by the xOrder client software.

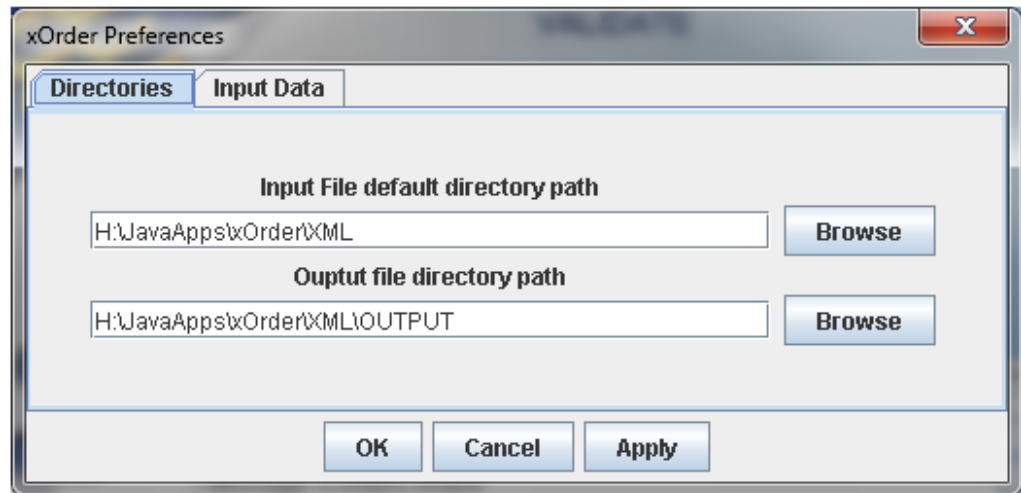


Figure 4: xOrder preferences - directories screen

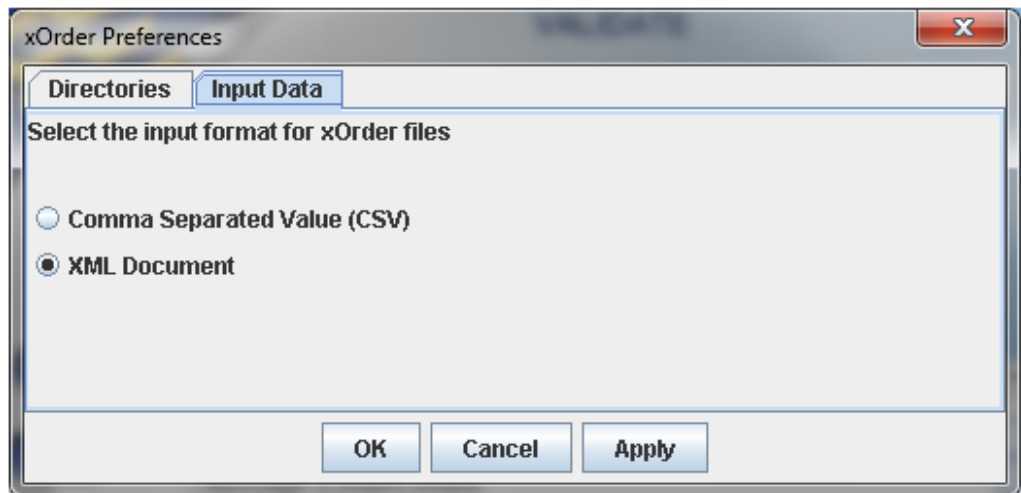


Figure 5: xOrder preferences - input data screen

The xOrder application requires the user to specify what type of data file format will be used. By default the xOrder software is configured for XML as shown in Figure 5 above. Currently XML and CSV files are the supported data input formats, however further releases of xOrder will support ODBC SQL interfaces. It should be noted that for prospective implementations where the input file format is CSV, a customization

software patch is required by ContainerWorld to manipulate the data into a xOrder recognized format; whereas the XML document is an already known data model.

## Processing Files

To begin using xOrder to process orders there are two steps: validation and transmission. The validation step is optional; however it provides a convenient tool for new users or system administrators to test the validity of a data document. There is no risk of error by skipping the validation step because any data errors will be detected through transmission as well. All warning and error messages will be displayed in the status output window (Figure 6 below) and detailed in the xOrder log file.

Following the optional validation is the transmission of the data file. This procedure will validate the data, transmit the file, and process the data into ContainerWorld's system. The status output and logging will be completed in the same manner as it was with validation.

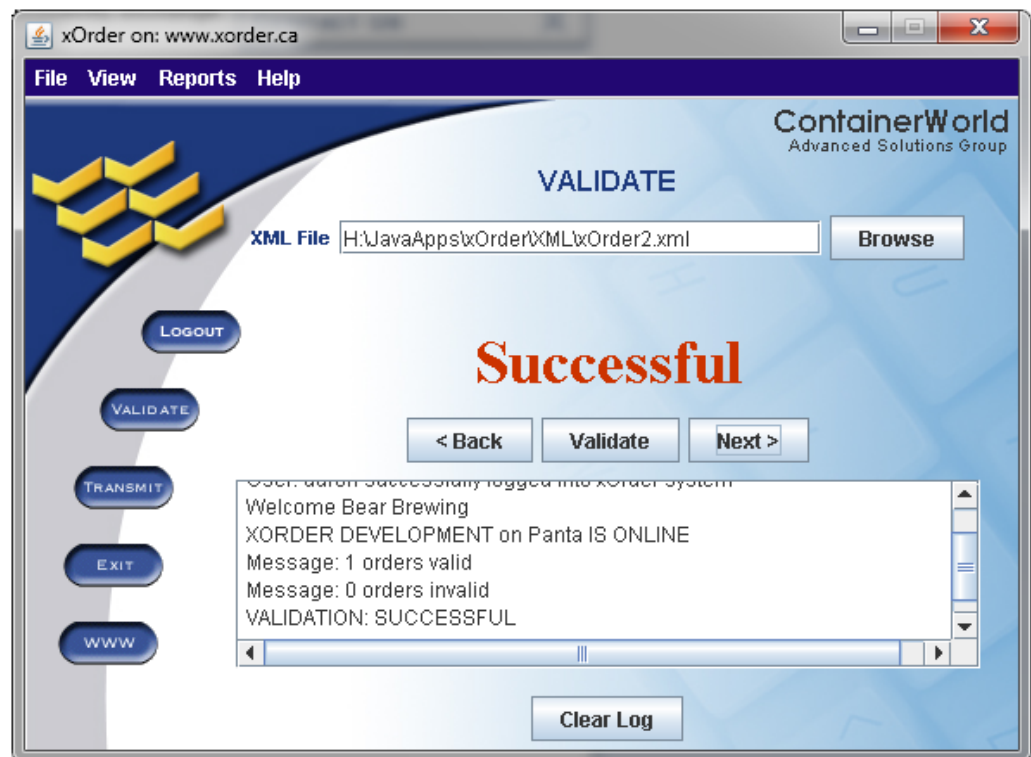


Figure 6: xOrder validate screen

Once a transfer is complete the user can logoff or shutdown the application. If there are any errors or situations arising from the transmission then the errors and/or warnings will be displayed in the log file, status output window, and with more detail in the xOrder Transaction Summary report (see the Reports and Confirmation section).

If there have been any orders rejected the user can investigate the problem, correct the data file and retransmit. Also, if any updates are necessary to the data file, xOrder can be used re-transmit and the changes will be applied. However, if ContainerWorld has already begun production of the order, the updates will be rejected.

- Orders are identified by the Supplier Reference Number in the data file. A change to this number will result in a new order. If a file is retransmitted with the same supplier reference number it will be treated as an update.
- xOrder does not handle order deletes. To delete an order you will have to contact the ContainerWorld Customer Service Department (see Support Section).

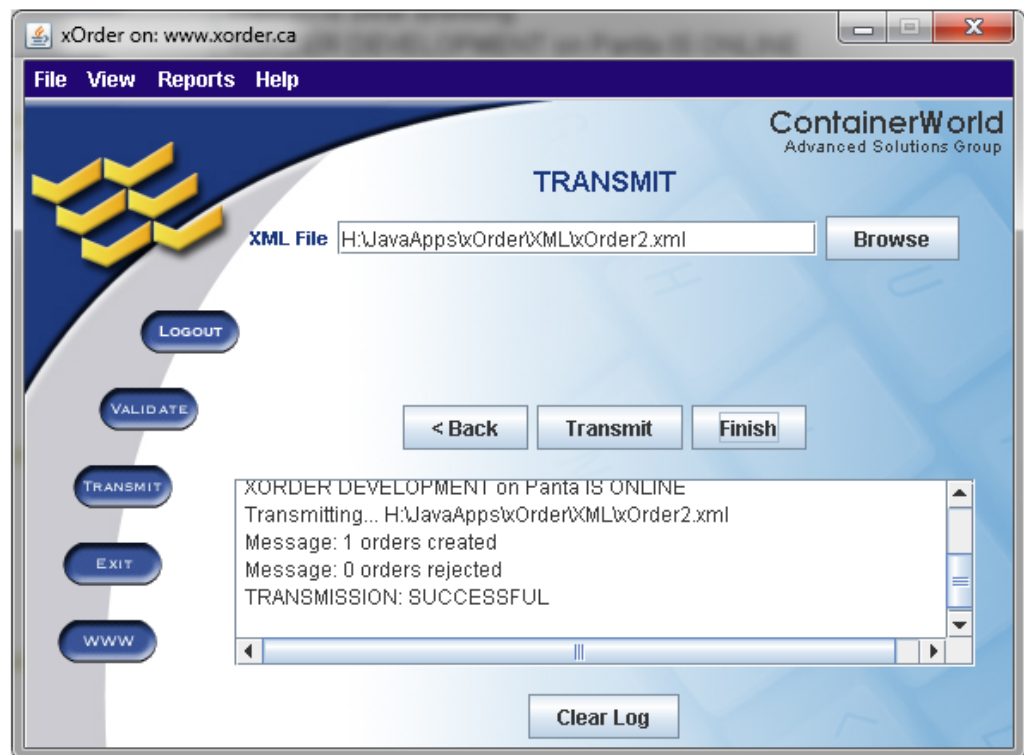
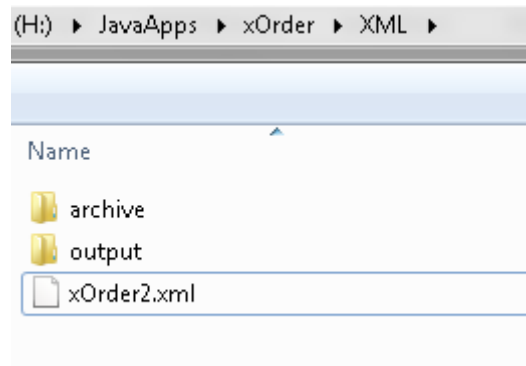


Figure 7: xOrder transmission screen

## XORDER ADMINISTRATION GUIDE

xOrder offers an option to archive the transmitted files. The settings to enable this option are very simple:

- 1) Let us know that you would like to archive your files and we will do necessary settings on the server.
- 2) Create the 'archive' folder in the input file directory.



## xOrder Direct Transmission API

*This option requires customization to enable ContainerWorld's customer's systems to connect directly to xOrder web services.*

To take advantage of the xOrder Direct Transmission option, ContainerWorld's customer systems will act as the "xOrder client" and will communicate directly with xOrder web services. These services are provided by two API Java Servlets: LogonServlet and xOrderServlet. The communication with these services is performed using xml documents transmission that includes a request from the client and a response from the web service.

### LogonServlet API

The LogonServlet performs user authentication with two services: Logon Request and Logout Request.

The "xOrder client" will construct an xml document including the username, password and bcldbNum; this xml document will be sent to the "Logon request" service for authentication, if successful, the service will create a new session in CW systems.

```
<xLogon>
  <userid>username</userid>
  <password>password</password>
  <bcldbNum>bcldbNum</bcldbNum>
</xLogon>
```

This session will be used by the LogonServlet to conform an xml document including the user status, username, sessionId, useDiscPriceFlag, archiveFileFlag and xOrderMessage which will be sent as response to the "xOrder Client".

```
<uservalidation>
  <userstatus>valid</userstatus>
  <username>userName</username>
  <sessionId>newSessionId</sessionId>
  <bcldbNum>BcldbNum</bcldbNum>
  <useDiscPriceFlag>UseDiscPriceFlag</useDiscPriceFlag>
  <archiveFileFlag>ArchiveFileFlag</archiveFileFlag>
  <xOrderMessage>xorderMessage</xOrderMessage>
</uservalidation>
```

The “xOrder client” (customer’s system) will store the session data and must include it when validating or transmitting purchase orders.

After transmission is completed, the “xOrder client” should perform a “*Logout Request*” to the Logon Service API to remove the session from CW systems. This request should reference the session data previously created by the “Logon Request”.

```
<xLogout>
  <userid>userName</userid>
  <sessionId>sessionId</sessionId>
  <bcldbNum>BcldbNum</bcldbNum>
</xLogout>
```

## xOrderServlet API

The xOrderServlet performs xml document validation and transmission. The validation will ensure that the XML document structure is compliant with the xOrder schema and that the data included in the document follows the ContainerWorld’s data definitions.

The transmission also performs the document validation and when successful, will process the data into ContainerWorld’s system.

The client application will select an XML document containing the actual orders; this document is sent as binary data in the HttpServletRequest inputStream.

Along with the xOrder XML document the client request includes the following headers in the HttpServletRequest:

Content-type	“validate/xml” for order validation, sets updateDBFlag=“N” “text/xml” for order creation, sets updateDBFlag=“Y”
userid	The user name
Password	The user password
useSuppDiscFlag:	The flag that indicates to use the supplier discount price, default value = “N”
sessionId	The sessionId obtained from logon request
bcldbNum	The supplier BCLDB number, which should match the bcldbNum included in the xOrder XML file
filename	The name of the xOrder XML file.

See Appendix E for a code sample of the API integration

## Transmission Results

After an XML document is sent to the xOrderServlet, the service will process the orders included and generate a response file stream in xml format which provides details for every order, a small summary and a transmission status. Response transmission is in real-time and can be used by the customer's system to integrate live results and exception handling within their system.

Possible order details:

- Not enough quantity for CSPC 610535: available - 1 order qty - 3.
- Error 14: PO 1184847 is no longer accepting updates.
- Error 25: Not enough available quantity for cspc +576959 vintage 2000.

Available response statuses:

- Success: when all the orders are processed correctly.
- Failed: when all the orders are rejected.
- Done: when some orders are processed correctly and some are rejected.

Sample response in xml format:

```
<!-- Validation messages -->
<servletMessages>
  <message>Order# 3091272 is no longer accepting updates.</message>
  <message>0 orders valid</message>
  <message>1 orders invalid</message>
  <status>failed</status>
  <warnings>true</warnings>
</servletMessages>

<!-- Failed validation messages -->
<servletMessages>
  <message>7 orders valid</message>
  <message>0 orders invalid</message>
  <status>success</status>
</servletMessages>

<!-- Successful transmission messages -->
<servletMessages>
  <message>7 orders created</message>
  <message>0 orders rejected</message>
  <status>success</status>
</servletMessages>
```


## Reports & Confirmation

*xOrder provides several detailed reports with real-time data to allow complete monitoring and analysis of transmissions and processing. There's no guess work in what's happening behind the software, you process it; you know it.*

Understanding what is happening when an order is processed and knowing the results is an important component of xOrder. This is why xOrder provides several real-time reports to ensure that the users are informed and can react quickly to any errors or warnings during transmission. All reports are accessible from within the xOrder software or through the ContainerWorld customer access web site. For the convenience of our customers, the system reports are generated into Adobe™ PDF format for easy archiving onto the customers data.

### Transaction Summary Report

For every transaction that is processed through xOrder a detailed transaction record is created and stored on a database. These records are used to generate the xOrder transaction summary report shown below in figure 8.



**Private Distribution System**  
 xOrder Transaction Summary

Page: 1 of 1  
 Report Code: XORDSND  
 Generated: 20-Jun-12 11:59:10

GREAT BEVERAGE CO.  
 WAREHOUSE  
 xorder\_great\_bev\_753.xml

Supplier Ref	CWFS PO	Licensee	Order Comments	No Stock	Rdcd Qty	Do Not Ship	Alt Addr	Off Sched	Priv Label	xOrder System Comments
753	2208997	195477 - Legacy Liquor Store								

Total batch order count: 1 of which 0 failed to be processed

Total report order count: 1 of which 0 failed to be processed

**Order Details**  
 follows for the 2nd half of report

xorder\_great\_bev\_753.xml

Supplier Ref	CWFS PO	Licensee	Deliv Date	Deliv Day	Delivery Schedule	Order Comments
753	2208997	195477 - Legacy Liquor Store	29-JUN-2012	Friday	Fr	
SKU Number	CSPC	Product Description	Order Qty	Filled Qty	N/Avail Qty	
123456	123456	Delicious Red Wine	1	1		

Total report order item count: 1 of which 0 differed from order request

Total report ordered case count: 1 of which 1 cases were filled. A difference of 0 cases

REPORT FILTER

Figure 8: xOrder transaction summary report



Within the summary report the results of each order transmission are shown; most importantly the user should know if any orders were rejected during the xOrder process. Any rejected orders are shown with a large red 'X' in the first column of the report beside the supplier order number and are accompanied by a system explanation in the column labeled 'xOrder System Comments'. For accepted orders there may still be system messages or warnings that the user should be aware of (also shown in the system comments). Lastly, there are specific flags that may be set by xOrder to indicate predefined special instructions. For example, if delivery is to be sent to a different address then the 'Alt Addr' (alternate address) flag is set and displayed to the user with an 'X'. Another example is when the requested order qty, in a xOrder data file, exceeds the available quantity in ContainerWorld's system; for this scenario the 'Rdcd Qty' (reduced quantity) flag is set. Currently, xOrder handles several predefined reporting flags, however the system allows for adding new reporting conditions to properly accommodate new customers and any unique business rules they may have.

## **Reverse Confirmation – Inventory Synchronization**

The main purpose for xOrder is to communicate data from a remote system to ContainerWorld's system, yet some systems require an acknowledgment or confirmation data. The xOrder reverse confirmation download is designed to bridge a two way communication requirement. The main benefit for using the reverse confirmation is to ensure inventory synchronization between two separate inventories. For example, a customer's order system generates an order for 10 cases of a product; this order is transmitted via xOrder, however the ContainerWorld system indicates an inventory level of 9 cases and therefore can only create an order for 9 cases, not 10 as originally requested. The reverse confirmation would report back to the customer's system allowing the original order to be adjusted.

To accommodate differing business rules for customers, the xOrder reverse confirmation can be configured to synchronize data at different positions along the life cycle of an order. As well, the content of the reverse data is customizable for individual installations. As an option, current inventory levels can be included in the synchronization data to capture any system discrepancies as they happen rather than trying to reconcile inventory through counts and monthly analysis.

Row Type	Supplier Order#	CW Order#	CSPC	Supplier Product#	Shipped Qty	Order Total Qty
I	100046	241278	000091	1000013	6	
I	100046	241278	078493	1000049	7	
T	100046	241278				13

Table 1: xOrder confirmation data definition

Currently, the confirmation data is available using two methods: web site download, or an embedded download in the xOrder software. The confirmation download was made available from the website and not just the xOrder software because of a common trend whereby IT and Order Desk staff had different roles with xOrder. Commonly IT wanted to handle the confirmation import to their system, meanwhile the order desk staff usually operated the xOrder transmission. Figure 9 shows the web interface and a caption of the resulting confirmation data. The existing confirmation data format is CSV (Comma Separated Value) although other formats are available upon request. A sample confirmation file is detailed above in table 1.

For the technical requirements of the xOrder Reverse Confirmation download data refer to Appendix D: Confirmation Structure. This appendix provides the allowed data types, column names, and maximum allowable sizes. Also note that xOrder supports customization to tailor a reverse data solution to specific business rules including automated or manual import via an ODBC SQL interface. Any questions regarding customization should be directed the xOrder Systems Integrator listed in the ‘Customer Service/Support’ section of this document.

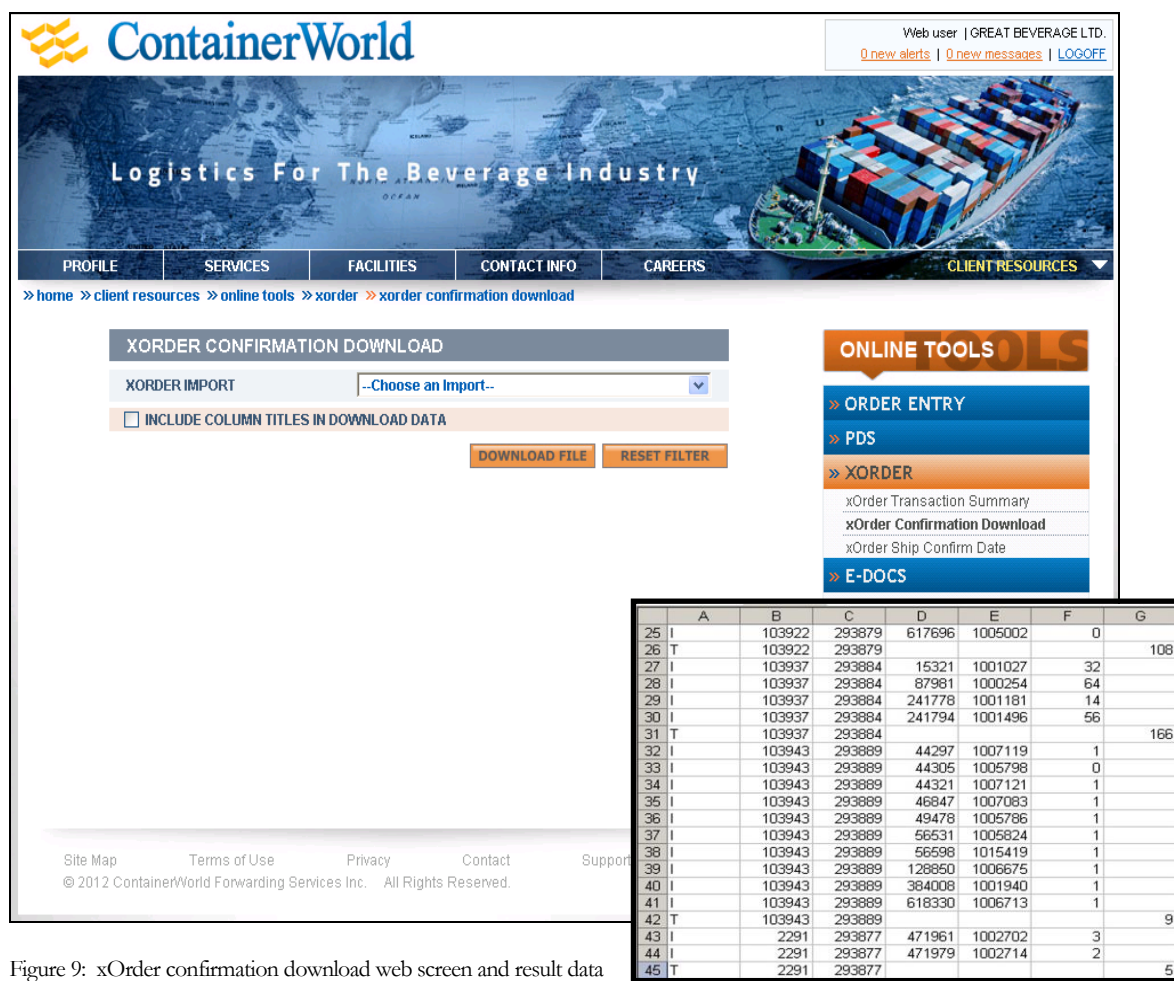


Figure 9: xOrder confirmation download web screen and result data

## Appendix A: XML Document Structure

Element	Valid Values	Max. Length	Tag Req / Opt
bclbdbNum	Numeric	11	Required
orderBolReqOverrideFlag	'Y' or 'N'	1	Optional
supplierPoRef	Alphanumeric	12	Required
orderDate	Format : YYYY-MM-DD (dashes included)	10	Required
deliveryDate	Format : YYYY-MM-DD (dashes included)	10	Required
paymentMethod	CC-AMEX (American Express) CC-MC (MasterCard) CC-VISA (Visa) CCO (Certified Cheque Only) CHQ (Cheque) CHQ-MAIL (Cheque by Mail) COD (Cash on Delivery) CSH (Cash only) DDB (Direct Debit) DDP (BCLDB Direct Debit) EF (Electronic Funds Transfer) NC (No charge) NET_30 (NET-30) PTF (Payment to Follow) ST (Stock Transfer)	7	Required
warehouse	CLIV (Commercial Logistics Vancouver) CLK2 (Commercial Logistics Kelowna 2)	6	Required
poComments	Alphanumeric	500	Optional
doNotShip	'Y' or 'N'	1	Optional
privateLabel	'Y' or 'N'	1	Optional
deliveryTo	'Y' or 'N'	1	Optional
customerNum	Numeric	6	Required

**XORDER ADMINISTRATION GUIDE**

shipToLocation	Numeric	6	Optional
customerOrderRef	Alphanumeric	12	Optional
custDiscountPercent	Decimal 4 decimal places (As of April 1st, 2015 this element is not valid and will be set to zero.)	11	Optional
taxGstPercent	Decimal 2 decimal places	N/A	Required
taxGstAmount	Decimal 2 decimal places	N/A	Required
taxPstPercent	Decimal 2 decimal places	N/A	Optional
taxPstAmount	Decimal 2 decimal places	N/A	Optional
orderType	Alphanumeric in list ('D', 'U', 'R', 'P') If orderType is not included in the xOrder XML file the value will be set to 'D'.	1	Optional
subTotal	Decimal 2 decimal places	N/A	Required
litterDeposit	Decimal 2 decimal places (total order deposit amount)	9	Required
totAmount	Decimal 2 decimal places	N/A	Required
totCases	Numeric	N/A	Required
cspc	Alphanumeric	6	Required
pidNum	Alphanumeric	12	Optional
actionType	'R' or 'P'	1	Optional
vintage	Numeric	4	Optional
quantity	Numeric	11	Required
custPrice	Decimal 2 decimal places	9	Required
custDiscountPrice	Decimal 2 decimal places (As of April 1st, 2015 this element is not valid and will be set to the same value as custPrice.)	11	Optional

## XORDER ADMINISTRATION GUIDE

custLitterDeposit	Decimal 2 decimal places	9	Required
kegQuantity	Numeric	5	Optional
kegAdjustmentAmount	Decimal 2 decimal places	9	Optional
refNumber	Numeric	11	Optional
speedyMailerAmount	Decimal 2 decimal places	9	Optional

Table 2: xOrder XML document structure – technical requirements

Please note the following:

- xOrder supports the BCLDB new wholesale pricing model effective April 1<sup>st</sup>, 2015.
- It's possible that not all optional elements apply to your business needs, before including those into the xml file please contact ContainerWorld's I.T. department.

## Appendix B: XML Document Definitions

Element	Description
bcldbNum	CWFS assigned customer number
orderBolReqOverrideFlag	Used to override Form60 and BOL settings 'Y' - BOL required 'N' - do not override existing settings
supplierPoRef	Supplier assigned/controlled unique order number. The supplier order number is printed on Form60.
orderDate	Date the order was placed at supplier's order desk
deliveryDate	Required delivery date to the customer (licensee)
paymentMethod	Method of payment by a licensee for the order
warehouse	Warehouse where to ship the order from
poComments	Order comments from the supplier's order desk
doNotShip	Flag to indicate that the order will not be delivered
privateLabel	Flag to indicate that the product is a private Label
deliveryTo	Flag to indicate there will be a different delivery address
customerNum	Number for the customer/licensee placing the order
shipToLocation	Supplier assigned/controlled number for an alternate location or site reference of a customer/licensee
customerOrderRef	Licensee assigned/controlled order number. The customer order number is printed on Form60.
custDiscountPercent	Discount Factor, either agency or licensee. As of April 1st, 2015 this element is not valid and will be set to zero.

**XORDER ADMINISTRATION GUIDE**

taxGstPercent	GST rate applied to the order
taxGstAmount	GST amount equals subtotal multiplied by GST rate
taxPstPercent	PST rate applied to the order
taxPstAmount	PST amount equals subtotal multiplied by PST rate
orderType	D = Delivery U = Customer Pickup R = Return Pickup and Delivery P = Return Pickup If orderType is not included in the xOrder XML file the value will be set to 'D'.
subTotal	Supplier assigned/order subtotal equal to the sum of custPrice multiplied by quantity
litterDeposit	Supplier assigned/controlled order total litter deposit equal to the sum of custLitterDeposit multiplied by quantity
totAmount	Supplier assigned/controlled order total amount equal to the sum of product subTotal + taxGstAmount + litterDeposit
totCases	Total number of cases in the order
cspc	BCLDB product number
pidNum	Supplier assigned/controlled product number
actionType	R = Replenish customer (normal delivery) P = Pickup from customer (returning product)
vintage	Product vintage year
quantity	Number of shipping units (cases) ordered
custPrice	Supplier assigned/controlled product price per shipping unit (case)
custDiscountPrice	Calculated Discounted price per shipping unit (As of April 1st, 2015 this element is not valid and will be set to the same value as custPrice.)
custLitterDeposit	Calculated litter deposit per shipping unit
kegQuantity	Number of keg credits for one keg adjustment
kegAdjustmentAmount	Total dollar amount for one keg adjustment

## XORDER ADMINISTRATION GUIDE

refNumber	Supplier assigned/controlled reference number for one adjustment
SpeedyMailerAmount	Total dollar amount for one speedy mailer adjustment

Table 3: xOrder XML document definitions



## Appendix C: xOrder Schema (a)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema" elementFormDefault = "qualified">
  <!--Defintion of simple types -->
  <xs:simpleType name = "StdDateType">
    <xs:annotation>
      <xs:documentation>Restricted xs:date that accepts only positive Gregorian calendar dates in the format YYYY-MM-DD
      (dashes included) without time zone indicator.</xs:documentation>
    </xs:annotation>
    <xs:restriction base = "xs:date">
      <xs:pattern value = "[1-9][0-9]{3}\-(0[1-9]|1[012])\-(0[1-9]|12)[0-9]{3}[01]"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name = "StdCurrencyType">
    <xs:annotation>
      <xs:documentation>Accepts dollar amounts with 2 decimal places.</xs:documentation>
    </xs:annotation>
    <xs:restriction base = "xs:decimal">
      <xs:fractionDigits value = "2"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name = "YesNoFlag">
    <xs:restriction base = "xs:string">
      <xs:enumeration value = "Y"/>
      <xs:enumeration value = "N"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name = "XOrder">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "Validation"/>
        <xs:element ref = "PdsPurchaseOrders"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name = "Validation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "bcldbNum"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name = "bcldbNum">
    <xs:simpleType>
      <xs:restriction base = "xs:integer">
        <xs:totalDigits value = "11"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name = "PdsPurchaseOrders">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs = "unbounded" ref = "PdsPurchaseOrder"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

## Appendix C: xOrder Schema (b)

```

<xs:element name = "PdsPurchaseOrder">
  <xs:complexType>
    <xs:all>
      <xs:element
        minOccurs = "0"
        name = "orderBolReqOverrideFlag"
        type = "YesNoFlag"/>
      <xs:element ref = "supplierPoRef"/>
      <xs:element ref = "orderDate"/>
      <xs:element ref = "deliveryDate"/>
      <xs:element ref = "paymentMethod"/>
      <xs:element ref = "warehouse"/>
      <xs:element minOccurs = "0" ref = "poComments"/>
      <xs:element minOccurs = "0" ref = "doNotShip"/>
      <xs:element minOccurs = "0" ref = "privateLabel"/>
      <xs:element minOccurs = "0" ref = "deliveryTo"/>
      <xs:element ref = "customerNum"/>
      <xs:element minOccurs = "0" ref = "shipToLocation"/>
      <xs:element minOccurs = "0" ref = "customerOrderRef"/>
      <xs:element minOccurs = "0" ref = "custDiscountPercent"/>
      <!-- New Elements to support Sales Taxes excluded from product price -->
      <xs:element ref = "taxGstPercent"/>
      <xs:element ref = "taxGstAmount"/>
      <xs:element minOccurs = "0" ref = "taxPstPercent"/>
      <xs:element minOccurs = "0" ref = "taxPstAmount"/>
      <xs:element minOccurs = "0" ref = "orderType"/>
      <xs:element ref = "Products"/>
      <xs:element ref = "subTotal"/>
      <xs:element ref = "litterDeposit"/>
      <xs:element ref = "totAmount"/>
      <xs:element ref = "totCases"/>
      <xs:element minOccurs = "0" ref = "Adjustments"/>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name = "supplierPoRef">
  <xs:simpleType>
    <xs:restriction base = "xs:string">
      <xs:maxLength value = "12"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name = "orderDate" type = "StdDateType"/>
<xs:element name = "deliveryDate" type = "StdDateType"/>
<xs:element name = "paymentMethod">
  <xs:simpleType>
    <xs:restriction base = "xs:string">
      <xs:enumeration value = "CC-AMEX"/>
      <xs:enumeration value = "CC-MC"/>
      <xs:enumeration value = "CC-VISA"/>
      <xs:enumeration value = "CCO"/>
      <xs:enumeration value = "CHQ"/>
      <xs:enumeration value = "CHQ-MAIL"/>
      <xs:enumeration value = "COD"/>
      <xs:enumeration value = "CSH"/>
      <xs:enumeration value = "DDB"/>
      <xs:enumeration value = "DDP"/>
      <xs:enumeration value = "EF"/>
      <xs:enumeration value = "NC"/>
      <xs:enumeration value = "NET_30"/>
      <xs:enumeration value = "PTF"/>
      <xs:enumeration value = "ST"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name = "warehouse">
  <xs:simpleType>
    <xs:restriction base = "xs:string">
      <xs:enumeration value = "CLIV"/>
      <xs:enumeration value = "CLK2"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

## Appendix C: xOrder Schema (c)

```

    <xs:element name = "poComments">
      <xs:simpleType>
        <xs:restriction base = "xs:string">
          <xs:minLength value = "0"/>
          <xs:maxLength value = "500"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name = "doNotShip" type = "YesNoFlag"/>
    <xs:element name = "privateLabel" type = "YesNoFlag"/>
    <xs:element name = "deliveryTo" type = "YesNoFlag"/>
    <xs:element name = "customerNum">
      <xs:simpleType>
        <xs:restriction base = "xs:integer">
          <xs:totalDigits value = "6"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name = "shipToLocation">
      <xs:simpleType>
        <xs:restriction base = "xs:integer">
          <xs:totalDigits value = "6"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name = "customerOrderRef">
      <xs:simpleType>
        <xs:restriction base = "xs:string">
          <xs:maxLength value = "12"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name = "custDiscountPercent">
      <xs:simpleType>
        <xs:restriction base = "xs:decimal">
          <xs:fractionDigits value = "4"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name = "taxGstPercent">
      <xs:simpleType>
        <xs:restriction base = "xs:decimal">
          <xs:fractionDigits value = "2"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name = "taxPstPercent">
      <xs:simpleType>
        <xs:restriction base = "xs:decimal">
          <xs:fractionDigits value = "2"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name = "orderType">
      <xs:annotation>
        <xs:documentation>Order Types: D=Delivery, U=Customer Pickup, R=Return Pickup and Delivery and P=Return
        Pickup.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base = "xs:string">
          <xs:enumeration value = "D"/>
          <xs:enumeration value = "U"/>
          <xs:enumeration value = "R"/>
          <xs:enumeration value = "P"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name = "Products">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs = "unbounded" ref = "Product"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

```

## Appendix C: xOrder Schema (d)

```

<xs:element name = "Product">
  <xs:complexType>
    <xs:all>
      <xs:element minOccurs = "0" ref = "cspc"/>
      <xs:element minOccurs = "0" ref = "pidNum"/>
      <xs:element minOccurs = "0" ref = "vintage"/>
      <xs:element minOccurs = "0" ref = "actionType"/>
      <xs:element ref = "quantity"/>
      <xs:element ref = "custPrice"/>
      <xs:element minOccurs = "0" ref = "custDiscountPrice"/>
      <xs:element minOccurs = "0" ref = "custLitterDeposit"/>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name = "cspc">
  <xs:simpleType>
    <xs:restriction base = "xs:string">
      <xs:maxLength value = "6"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name = "pidNum">
  <xs:simpleType>
    <xs:restriction base = "xs:string">
      <xs:maxLength value = "12"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name = "vintage">
  <xs:simpleType>
    <xs:restriction base = "xs:integer">
      <xs:totalDigits value = "4"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name = "actionType">
  <xs:simpleType>
    <xs:restriction base = "xs:string">
      <xs:enumeration value = "P"/>
      <xs:enumeration value = "R"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name = "quantity">
  <xs:simpleType>
    <xs:restriction base = "xs:integer">
      <xs:totalDigits value = "11"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name = "custPrice" type = "StdCurrencyType"/>
<xs:element name = "custDiscountPrice" type = "StdCurrencyType"/>
<xs:element name = "custLitterDeposit" type = "StdCurrencyType"/>
<xs:element name = "subTotal" type = "StdCurrencyType"/>
<!-- New Elements to support Sales Taxes excluded from product price-->
<xs:element name = "taxGstAmount" type = "StdCurrencyType"/>
<xs:element name = "taxPstAmount" type = "StdCurrencyType"/>
<xs:element name = "litterDeposit" type = "StdCurrencyType"/>
<xs:element name = "totAmount" type = "StdCurrencyType"/>
<xs:element name = "totCases">
  <xs:simpleType>
    <xs:restriction base = "xs:integer">
      <xs:totalDigits value = "11"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

## Appendix C: xOrder Schema (e)

```

<xs:element name = "Adjustments">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs = "0" ref = "KegAdjustment"/>
      <xs:element
        minOccurs = "0"
        maxOccurs = "2"
        ref = "SpeedyMailerAdjustment"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name = "KegAdjustment">
  <xs:complexType>
    <xs:all>
      <xs:element name = "kegQuantity" type = "xs:integer"/>
      <xs:element name = "kegAdjustmentAmount" type = "StdCurrencyType"/>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name = "SpeedyMailerAdjustment">
  <xs:complexType>
    <xs:all>
      <xs:element name = "refNumber" type = "xs:integer"/>
      <xs:element name = "speedyMailerAmount" type = "StdCurrencyType"/>
    </xs:all>
  </xs:complexType>
</xs:element>
</xs:schema>

```

The xOrder schema can be found in [http://www.xorder.ca/xOrder/schema/xorder\\_schema.xsd](http://www.xorder.ca/xOrder/schema/xorder_schema.xsd)

## Appendix D: XML Document, Sample a.

```

<?xml version="1.0" encoding="UTF-8"?>
<XOrder>
  <Validation><bcldbNum>201234</bcldbNum></Validation>
  <PdsPurchaseOrders>
    <PdsPurchaseOrder>
      <supplierPoRef>10072A</supplierPoRef>
      <orderDate>2015-04-03</orderDate>
      <deliveryDate>2015-04-20</deliveryDate>
      <paymentMethod>CC-VISA</paymentMethod>
      <warehouse>CLIV</warehouse>
      <poComments>Deliver to the back door</poComments>
      <customerNum>192907</customerNum>
      <customerOrderRef>12345</customerOrderRef>
      <orderType>D</orderType>
      <taxGstPercent>5</taxGstPercent>
      <taxGstAmount>12.50</taxGstAmount>
      <taxPstPercent>0</taxPstPercent>
      <taxPstAmount>0.00</taxPstAmount>
      <custDiscountPercent>0.0000</custDiscountPercent>
      <Products>
        <Product>
          <cspc>593087</cspc>
          <pidNum>1000227</pidNum>
          <actionType>R</actionType>
          <quantity>5</quantity>
          <custPrice>10.00</custPrice>
          <custDiscountPrice>10.00</custDiscountPrice>
          <custLitterDeposit>.10</custLitterDeposit>
        </Product>
        <Product>
          <cspc>188227</cspc>
          <pidNum>1080227</pidNum>
          <actionType>R</actionType>
          <quantity>10</quantity>
          <custPrice>20.00</custPrice>
          <custDiscountPrice>20.00</custDiscountPrice>
          <custLitterDeposit>.10</custLitterDeposit>
        </Product>
      </Products>
      <subTotal>250.00</subTotal>
      <litterDeposit>1.50</litterDeposit>
      <totAmount>264.00</totAmount>
      <totCases>15</totCases>
    </PdsPurchaseOrder>
    <PdsPurchaseOrder>
      <supplierPoRef>10072B</supplierPoRef>
      <orderDate>2015-04-05</orderDate>
      <deliveryDate>2015-04-11</deliveryDate>
      <paymentMethod>CC-VISA</paymentMethod>
      <warehouse>CLIV</warehouse>
      <poComments>Order has been taken by Rep. Order only for inventory balance</poComments>
      <customerNum>192634</customerNum>
      <customerOrderRef>12345</customerOrderRef>
      <doNotShip>Y</doNotShip>
      <privateLabel>Y</privateLabel>
      <taxGstPercent>5</taxGstPercent>
      <taxGstAmount>20.00</taxGstAmount>
      <Products>
        <Product>
          <cspc>593087</cspc>
          <vintage>2014</vintage>
          <quantity>40</quantity>
          <custPrice>10.00</custPrice>
          <custLitterDeposit>.20</custLitterDeposit>
        </Product>
      </Products>
      <subTotal>400.00</subTotal>
      <litterDeposit>8.00</litterDeposit>
      <Adjustments>
        <KegAdjustment>
          <kegQuantity>2</kegQuantity>
          <kegAdjustmentAmount>60.00</kegAdjustmentAmount>
        </KegAdjustment>
        <SpeedyMailerAdjustment>
          <refNumber>456</refNumber>
          <speedyMailerAmount>12.00</speedyMailerAmount>
        </SpeedyMailerAdjustment>
      </Adjustments>
      <totAmount>356.00</totAmount>
      <totCases>40</totCases>
    </PdsPurchaseOrder>
  </PdsPurchaseOrders>
</XOrder>

```

## Appendix D: XML Document, Sample b.

```

<?xml version="1.0" encoding="UTF-8"?>
<XOrder>
  <Validation><bcldbNum>987456</bcldbNum></Validation>
  <PdsPurchaseOrders>
    <PdsPurchaseOrder>
      <orderBolReqOverrideFlag>N</orderBolReqOverrideFlag>
      <supplierPoRef>1696421</supplierPoRef>
      <orderDate>2015-04-01</orderDate>
      <deliveryDate>2015-04-02</deliveryDate>
      <paymentMethod>CC-VISA</paymentMethod>
      <warehouse>CLIV</warehouse>
      <poComments>Deliver before noon</poComments>
      <customerNum>7252</customerNum>
      <custDiscountPercent>0</custDiscountPercent>
      <taxGstPercent>5</taxGstPercent>
      <taxPstPercent>0</taxPstPercent>
      <Products>
        <Product>
          <cspc>78493</cspc>
          <pidNum>1000049</pidNum>
          <actionType>R</actionType>
          <quantity>2</quantity>
          <custPrice>54.00</custPrice>
          <custDiscountPrice>54.00</custDiscountPrice>
          <custLitterDeposit>1.20</custLitterDeposit>
        </Product>
        <Product>
          <cspc>5934</cspc>
          <pidNum>1097258</pidNum>
          <actionType>R</actionType>
          <quantity>2</quantity>
          <custPrice>131.25</custPrice>
          <custDiscountPrice>131.25</custDiscountPrice>
          <custLitterDeposit>1.20</custLitterDeposit>
        </Product>
      </Products>
      <subTotal>370.50</subTotal>
      <litterDeposit>4.80</litterDeposit>
      <taxGstAmount>18.53</taxGstAmount>
      <taxPstAmount>0.00</taxPstAmount>
      <totAmount>393.83</totAmount>
      <totCases>4</totCases>
    </PdsPurchaseOrder>
    <PdsPurchaseOrder>
      <orderBolReqOverrideFlag>N</orderBolReqOverrideFlag>
      <supplierPoRef>1696422</supplierPoRef>
      <orderDate>2015-04-02</orderDate>
      <deliveryDate>2015-04-08</deliveryDate>
      <paymentMethod>CC-VISA</paymentMethod>
      <warehouse>CLIV</warehouse>
      <poComments>Deliver before noon</poComments>
      <customerNum>17349</customerNum>
      <shipToLocation>27381</shipToLocation>
      <custDiscountPercent>0</custDiscountPercent>
      <taxGstPercent>5</taxGstPercent>
      <taxPstPercent>10</taxPstPercent>
      <Products>
        <Product>
          <cspc>78493</cspc>
          <pidNum>1000049</pidNum>
          <actionType>R</actionType>
          <quantity>2</quantity>
          <custPrice>54.00</custPrice>
          <custDiscountPrice>54.00</custDiscountPrice>
          <custLitterDeposit>1.20</custLitterDeposit>
        </Product>
        <Product>
          <cspc>5934</cspc>
          <pidNum>1097258</pidNum>
          <actionType>R</actionType>
          <quantity>2</quantity>
          <custPrice>131.25</custPrice>
          <custDiscountPrice>131.25</custDiscountPrice>
          <custLitterDeposit>1.20</custLitterDeposit>
        </Product>
      </Products>
      <subTotal>370.50</subTotal>
      <litterDeposit>4.80</litterDeposit>
      <totAmount>430.88</totAmount>
      <totCases>4</totCases>
      <taxGstAmount>18.53</taxGstAmount>
      <taxPstAmount>37.05</taxPstAmount>
    </PdsPurchaseOrder>
  </PdsPurchaseOrders>
</XOrder>

```

## Appendix E: Direct Transmission Code Snippet

```

/**Receives the user info and the file to be transmitted to the xOrderServlet,
 * conforms and xml file and sends the client request to validate or transmit.
 * @param args request parameters
 * @param userid String
 * @param password String
 * @return String
 * @throws Exception
 * */

public String getPostOrderXml(String[] args, String userid, String password) throws Exception {
    String returnXmlString = "";
    String inputFileName = "";
    // the parameters in args should be as follows
    // args[0]= "http://www.xorder.ca/xOrder/xOrderServlet";
    // args[1]= "C:\xOrder_files\your_file_name.xml";
    // args[2]= "Update"; //alternative value "Validate"
    if (args.length < 3) {
        return("<message>Invalid call to xPostOrder. Argument length Invalid.</message>");
    }
    else {
        try {
            URL url = new URL(args[0]);
            FileReader fr = new FileReader(args[1]);
            char[] buffer = new char[1024*1024]; // 1 Megabyte buffer for reading xml files

            int bytes_read = 0;

            URLConnection urlc = url.openConnection();
            // identify the action type and the input file type info the request header
            if (args[2].equals("Validate")) {
                urlc.setRequestProperty("Content-Type","validate/xml");
            }
            else {
                urlc.setRequestProperty("Content-Type","text/xml");
            }
            urlc.setRequestProperty("userid",userid);
            urlc.setRequestProperty("password",password);
            urlc.setRequestProperty("useSuppDiscFlag", xog.getUseSuppDiscFlag() );
            // below the sessionId is concatenated with an empty string to force the int to string
            urlc.setRequestProperty("sessionId","" + xog.getSessionId() );
            urlc.setRequestProperty("bcldbNum","" + xog.getBcldbNum() );
            // input file name for xorder_summary report
            inputFileName = args[1].substring(args[1].lastIndexOf("\\") + 1);
            urlc.setRequestProperty("fileName", inputFileName );
            urlc.setDoOutput(true);
            urlc.setDoInput(true);
            PrintWriter pw = new PrintWriter(urlc.getOutputStream());

            // Read and transmit your file to the servlet until the end of the XML file
            while ((bytes_read = fr.read(buffer)) != -1) {
                pw.write(buffer, 0, bytes_read);
            }
            fr.close();
            pw.close();
        }
    }
}

```



```
        // receive xml from servlet
        BufferedReader in = new BufferedReader(new InputStreamReader(urlc.getInputStream()));
        String inputLine;
        while ((inputLine = in.readLine()) != null) {
            returnXmlString = returnXmlString + inputLine;
        }
        in.close();
    } catch ( ConnectException ce ) {
        ce.printStackTrace();
    }
    catch (IOException ioe) {
        ioe.printStackTrace();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}
if (returnXmlString == ""){
    return "<servletMessages><message>Error: no messages returned from
Servlet</message></servletMessages>";
}
else
{
    return returnXmlString;
}
}
```

xOrder Direct Transmission: code snippet used to read an xml document from local file system and transmit it to xOrder Servlet.

## **Appendix F: Testing your XML file**

Once you are able to produce a XML file with data extracted from your ordering system, you should be able to validate it and test it.

Validate the xml file with the xOrder schema, this will confirm that your xml file structure is correct and complies with the xOrder specification. There are a few online tools available to perform this validation, we use this one: <http://www.corefiling.com/opensource/schemaValidate.html>, but you can find a few others with a quick google search.

This tool will indicate whether your file is “Well Formed” and “Complies with the Schema requirements” or will provide you with detailed messages concerning any errors.

Please keep in mind that this validation ensures that your XML files follow the xOrder specification, but does not guarantee that the data provided will result in a new order in ContainerWorld’s system, your file needs to be processed by xOrder to obtain this result.

After the schema validation is successful, your XML file can be verified using our xOrder ‘TEST’ server, please contact us to provide with an account and the appropriate URL to perform your testing.

Finally, please provide us with the Form-60, BOL or appropriate document to verify your testing results.

## Appendix G: Confirmation Structure

Element	Valid Values	Maximum Length
Row Type	Alpha in ('I', 'T') where I = item and T = total	1
Supplier Order #	Alphanumeric	12
CW Order #	Numeric	10
CSPC	Alphanumeric	6
Supplier Product #	Alphanumeric	12
Shipped Qty	Numeric	11
Order Total Qty	Numeric	11

Table 4: xOrder reverse confirmation structure – technical requirements

## **Customer Service/Technical Support**

### **Technical Support**

*Mon – Fri, 8:30AM – 4:30 PM PST*

ph: 1 (877) 838-8880

fax: (604) 276-1301 attn: xOrder system

[helpdesk@containerworld.com](mailto:helpdesk@containerworld.com)

<http://www.xOrder.ca>

### **General Customer Service, Order Desk**

*Mon – Fri, 8:00AM – 4:00 PM PST*

ph: (604) 276-1360

fax: (604) 276-1361