

Stone Edge Order Manager Developer's Guide

Version 5.9

© Copyright 2000-2014
 Stone Edge Technologies, Inc.
 399 Arcola Road
 Collegeville, PA 19426
 www.StoneEdge.com
 support@stoneedge.com
 484-927-4804

DOCUMENT OVERVIEW:

This document is for developers working with shopping carts that are not officially integrated with the Order Manager. If you have the ability to access the data gathered by your shopping cart and can manipulate the output of the data, you can build your own custom script on the website that uses the Order Manager's XML Import Specification to retrieve Order, Customer, Product and Inventory data. The Order Manager also supports two-way communication of product quantity on hand data and can also support sending status information back to your shopping cart system. This document is intended to explain how Order Manager communicates with the custom script on your website and what it expects in return. The custom script can be developed by you, Stone Edge Technologies, Inc. or a third party, and can be written in any language supported by your web server. Services of this nature are not covered by an annual Stone Edge Technical Support Contract, but may be purchased separately. Contact the [Sales](#) department for a quote or more information.

The Order Manager has the capacity to make a Form Post to a script on your website. The post includes a number of variables that dictate what the script on the website should do and what data or response it should return. The script should contain some type of user validation based on a user name, password and, optionally, a store code that can uniquely identify the given store to act upon should multiple stores reside on the website. The script should also test to ensure that the connection is using the https protocol prior to outputting any sensitive data.

To define what user credentials are passed by the Order Manager to the script on the website, the user will need to set up a "Shopping Cart" in the Order Manager to identify the User Name, Password and, optionally, the Store Code. Here you must also enter the URL that points to the custom script on the website. Since the Order Manager only maintains a single URL, the single custom script must perform all functions. The Order Manager passes a variable called "setifunction" describing which function should be executed by the script (function names are detailed later in this document). Based on the function name and supporting variables, the script on the website should then execute the required function then return either XML formatted data or a specific text response to the Order Manager.

Since the user can set up multiple shopping carts in the Order Manager, all areas where interaction with the website takes place will first require that you select the specific shopping cart that you want to work with prior to initiating the action (like downloading orders). The cart selection process tells the program where to post the data for download (which URL to post to). The Order Manager will gather the necessary information in form variables then, using the form "post" method, send the data to the custom script using a secure SSL connection. The Order Manager then awaits the response from the script on the website. Once the response is received, the Order Manager will process the information.

IMPORTANT NOTE: This document provides both "query string" and Form Post examples of the data posted to the website. The "query string" URLs can be used from a browser window (using the "Get" method) to test your script's response. Stone Edge suggests that when you develop your script, design it to respond to both the "Get" and "Post" methods for troubleshooting purposes.

IMPORTANT NOTE: Since you will be transferring sensitive customer data between your shopping cart and the Order Manager, we highly recommend that you add a test to your script that looks at the calling URL and determines that a secure connection exists before exporting any data. Order Manager posts the data using port 443 and https (Secure Hypertext Transfer Protocol - 128 bit encryption). Your script should not allow data output if the connection is not secure (the script should return an error if connection is not secure – error responses are discussed later in this document). Order Manager also has the ability to send a user name and password both in

For questions regarding this document, contact Stone Edge Technologies, Inc.
 484-927-4804 or Support@StoneEdge.com

the form post (to validate that the calling party has administrative access to your shopping cart) and in the header (for server login). User and password are optional security features but are highly recommended!

SCRIPT ERRORS:

Should errors occur in your custom script file while it is processing a request, the Order Manager can receive and display the error to the user. Should an error be encountered, the error should be the only output returned to Order Manager (do not start sending orders then throw an error unless the error is not anticipated or trappable).

Errors that can be handled by your script should return the following XML data:

```
<?xml version="1.0"?>
<SETITypeOfDownload>
  <Response>
    <ResponseCode>3</ResponseCode>
    <ResponseDescription>error message</ResponseDescription>
  </Response>
</SETITypeOfDownload>
```

In the above example "TypeOfDownload" represents either "Orders", "Products" or "Customers", so if you encountered an error during order download then the custom script should return:

```
<?xml version="1.0"?>
<SETIOrders>
  <Response>
    <ResponseCode>3</ResponseCode>
    <ResponseDescription>error message</ResponseDescription>
  </Response>
</SETIOrders>
```

The ResponseCode element should contain the number 3 when returning errors. The ResponseDescription element should contain the actual error message and any actions that the user can take to resolve the error (like changing user name or password, etc.). If the Order Manager receives a response code 3 it will display the "Error Message" in the ResponseDescription element to the user then abort the import.

The Order Manager can also accept the following type of error response at the beginning of the data stream:

SETIError: *Error Message*

If the Order Manager finds "SETIError: in the response it will display the "Error Message" to the user and abort the import.

If an unexpected error occurs then typically the server processing the script will output the run-time error message. In cases like this the error may not be picked up by Order Manager. If a complete order is output prior to the error, the program will not "see" the error message. If an order record is interrupted (partially output) then Order Manager will "see" the error and abort the import. This is crucial to validating the communications between Order Manager and the website. If the server throws an error but it is not seen by Order Manager during the import, the user may only get a fraction of the data available to download without knowing that the error occurred. This is why it is important to anticipate as many errors as possible and hold the output until all processing is completed by the script.

DOWNLOADING ORDER DATA:

When downloading orders, the Order Manager makes several form posts to a script on your website. The Order Manager makes the requests using the HTTPS protocol (port 443, 128 bit encryption) to ensure that the data transferred is secure. The script called should return the requested data using the XML Order format. The concept is similar to requesting a web page. System parameters guide how Order Manager responds to the downloaded data. All downloads typically entail three distinct communication steps:

1. Order Manager requests the script's version number from the website. This is done for general testing of communications between Order Manager and the script file and also to ensure that if structural changes are made to Order Manager to support a new script that you can test to see if the latest script is installed before running the download. This action also validates that the URL for the script is valid prior to initiating additional actions.
2. Order Manager then requests the total number of orders that need to be downloaded from the site. Typically a download of 100 orders or less causes no problems, however, if there are a large number of orders to retrieve, you may run in to timeout issues on the server as the script executes. The Order Manager has a system parameter that allows you to set the maximum number of orders to download at a time. If the total number of orders to download exceeds the parameter value, the Order Manager will call for several downloads, retrieving the maximum number of orders in each call until all orders are received.

Note: Effective with the v5.505 release of the Order Manager, two new parameters have been added to the program.

InternetConnectionTimeout - By default, the Order Manager will wait up to 60 seconds for a server connection. If you receive connection timeout errors from the site, you can set the InternetConnectionTimeout parameter to a higher value. The value is entered in the number of milliseconds to wait for the connection (60 seconds = 60000 milliseconds)

InternetReceiveTimeout – By default, the Order Manager will wait up to 30 seconds for a server response after the post has been sent. If you are receiving server timeout errors after 30 seconds, you can increase the InternetReceiveTimeout value as needed. The value is entered in the number of milliseconds to wait for the connection (30 seconds = 30000 milliseconds). While this allows the Order Manager to wait longer for a response, you may want to review your script instead. Typically a script should be able to output 100 orders in about 5-15 seconds. If it is taking much longer, review the code in the script and/or indexing on your database at the website. Be aware that these timeouts are on the "client" side of the exchange and do not impact timeouts explicitly defined on your web server! Your web server will likely have a 60-90 second timeout for script operations.

3. Once Order Manager determines how many times it will execute the download, it begins the download process. The data received by Order Manager is then fed into one or more XML files that are stored in the NewOrders directory (identified in the system parameter NewOrderLocation). Once the download(s) is complete, Order Manager will test and import the order file(s). Upon successful completion, the XML file(s) in the NewOrders directory can be moved to the DataArchives directory (identified in the system parameter ArchiveLocation) for recovery purposes. Keep in mind that these imported order files are not encrypted and may contain sensitive customer data. They should only be maintained for short periods as needed for possible troubleshooting or data recovery. If you do not want the Order Manager to record these files in the DataArchives directory, you can set the DeleteDownloadTextFiles system parameter to True to have the program delete the XML files after import.

1) Requesting Script Version Data:

This function should **not** require user name and password validation to receive the response and should also be allowed to run if the connection is not secure (should respond to both http and https requests). This enables quick testing by Stone Edge Technologies, Inc. when troubleshooting web communications.

Form Variables:

Variable Name	Value	From	Required
setifunction	sendversion	Hard coded into visual basic, cannot be modified	YES
omversion	The version number of the Order Manager being used	Value of the Version field in the last record in the Version table – Used for version control	NO

For questions regarding this document, contact Stone Edge Technologies, Inc.
484-927-4804 or Support@StoneEdge.com

For the examples below, Replace *www.mystore.com/script.ext* with the appropriate URL pointing to the script on your web server

Example URL for testing from browser – uses “GET” method:

After making the revisions detailed above, copy the entire line and paste it into the address bar of your browser.

<https://www.mystore.com/script.ext?setifunction=sendversion&omversion=6.000>

Example Form Post – uses the “POST” method:

Save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the “Send” button to send the request.

```
<FORM ACTION="https://www.mystore.com/script.ext" METHOD=POST>  
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE ="sendversion">  
  <INPUT TYPE=TEXT NAME = "omversion" VALUE ="5.000">  
  <INPUT TYPE=SUBMIT VALUE=Send>  
</FORM>
```

Expected Response:

SETIResponse: version=#

The # symbol above represents the version of the script file on the website in ##### or #.### format. The version number must be 4 numbers plus decimal. For example SETIResponse: version=1.000

Details:

Order Manager will test the returned value against the GenericScriptVersionNumber system parameter. If the returned value is less than the parameter value, Order Manager will notify the user that the script version is too low and will abort the order import. This provides a method of version control for new features or bug fixes.

2) Determining the number of orders to download

This function should require user name and password validation prior to a response and should also test that the connection is secure. For added security, a store “code” can be added to identify which store’s data you’ll need, should multiple stores reside on the website. If user validation or the test for a secure connection fails, an error response should be generated.

Order Manager will post a request for the total number of orders available for download. To determine the number of orders that need to be downloaded, the Order Manager will send two variables called “lastorder” and “lastdate”. If orders already exist in the Order Manager then the “lastorder” variable will contain the last web order number that exists in the Order Manager. Your script should only send orders that were received after that one. The “lastdate” variable will contain the last date that orders were downloaded from the website, usually in the “dd-mmm-yyyy” format; however, the format is determined by the source of the date, and may vary. You can use either variable as your starting point to determine how many additional orders need to be sent. If you use the “lastdate” variable, you should include all orders from that date in the download because some orders from that day may not have been received. The Order Manager will not import duplicate orders, so if some of the orders from that date have already been imported, the new download will ignore them.

IMPORTANT NOTE: *If there are NO orders in the Order Manager, both of these variables (lastorder & lastdate) will contain the value “All” to signal that ALL orders should be downloaded (normally only the first download into the program). Make sure that your code can handle this! There may not always be a number or date in the lastorder or lastdate variable!*

Form Variables:

Variable Name	Value	From	Required
setifunction	ordercount	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	Optional: A unique identifier for the store being contacted	Shopping Carts Form StoreCode field	NO
lastorder	The last order number in Order Manager received from this store. In the absence of an order number Order Manager will pass the text “all”	Retrieved programmatically at the time of download. Use this OR lastdate to identify which new orders to download.	Yes - if the lastdate variable is empty
lastdate	The date of the last order downloaded from this store. In the absence of an order date Order Manager will pass the text “all”	Retrieved programmatically at the time of download. Use this OR lastorder to identify which new orders to download. If using this as the determining factor, always include all orders from this date.	Yes – if the lastorder variable is empty
omversion	The version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

In the examples below revise the following data points prior to testing:

Replace “www.mystore.com/script.ext: with the appropriate URL pointing to the script on your web server.

Replace “auser” with the proper user name for login on the website.

Replace “pwd” with the proper password for login to the website.

Replace “mystore” with the store code identifier (optional).

Replace “1001” with an order number or the word “All”.

Replace “10-Jun-2003” with an appropriate date to start order download.

Example URL for testing from browser – uses “GET” method:

After making revisions detailed above, copy the entire line and paste it into the address bar of your browser.

`https://www.mystore.com/script.ext?setifunction=ordercount&setiuser=auser&password=pwd&code=mystore&lastorder=1001&lastdate=10-Jun-2003&omversion=5.000`

Example Form Post – uses “POST” method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the “Send” button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="ordercount">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "lastorder" VALUE = "1001">
  <INPUT TYPE=TEXT NAME = "lastdate" VALUE = "10-Jun-2003">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Note: If you have your web server set up to validate the request, you may need to enter your user name and password prior to receiving a response. If the user name and password are sent in the form variables, this assumes that the script, rather than the server, will perform the user validation.

Expected Response:

SetiResponse: ordercount=#

The symbol # above represents the number of orders available for download. This must be an integer value.

Details:

Order Manager will look at the value returned from the order count request. If the value is 0 (zero) then the user is prompted that there are no new orders in the system. If orders are available, Order Manager will check the GenericMaxOrderDownload system parameter to see if it needs to perform one or more calls for download. If more than one is required then the Order Manager will pass additional variables in the request identifying the record start value and the maximum number of orders to send (detailed in the next section).

3) Downloading Orders

This function should require user name and password validation prior to a response and should also test that the connection is secure. For added security, a store “code” can be added to identify which store’s data you’ll need. If user validation or the secure connection test fails, an error response should be generated.

Order Manager can read two text based file formats or XML formatted data. If you are integrating the Order Manager with your website using this guide, the system parameter GenericImportMethod should be set to “XML” to tell Order Manager to use the XML Import Specification.

NOTE: As of 4/1/05 we no longer include details regarding the “text” file import specification for the Order Manager in this document. We are limiting support for the text import as it has some inherent bug issues that are eliminated by the use of XML. The Order Manager will continue to support the Text File Import Specification for the foreseeable future for those merchants who currently use the text based system. Contact Stone Edge Technologies, Inc. if you require information on the “Text Import Specifications” for the Order Manager.

If the Order Manager determines that the number of orders to download exceeds the value of the GenericMaxOrderDownload parameter, the program will make multiple calls to the script, each time expecting to download the number of orders defined in the GenericMaxOrderDownload parameter. To identify which orders need to be sent in each call, the Order Manager will include two variables called “startnum” and “batchsize”. The variable “startnum” will contain the record number (NOT THE ORDER NUMBER) to start with and “batchsize” will contain the total number of orders to output for the request.

For example: If there are 500 orders to download from the website and the GenericMaxOrderDownload parameter is set to 100, the first call from the Order Manager to the website will set the startnum variable to 1 and the batchsize variable to 100. This means that your script should start at the first order record then download 100 orders. The second call will set the startnum variable to 101 and the batchsize variable to 100. This says start the export at the 101st order record and export 100 orders. The third call will set the startnum variable to 201 and the batchsize variable to 100. This says start the export at the 201st order record and export 100 orders. This continues through the fourth and fifth post until all orders are imported.

If the number of orders to download is less than the value of the GenericMaxOrderDownload then the Order Manager WILL NOT send the “startnum” and “batchsize” variables in the post. Your script must be able to deal with this by anticipating that the variables may not be present.

The Order Manager also has the ability to send a “decryption key” to the website should data on the site be encrypted. Several of the carts we support use public key/private key encryption of credit card information. If your site does support this, check the option “Website uses payment encryption” in the Create & Edit Shopping Carts form. Enter your private key in the Decryption Key field and again in the Reenter Key field. Entering your key in the Order Manager is optional. If you do not enter the key, but the encryption option is selected, the program will prompt for the key at the time of order download.

By selecting the payment encryption option, the Order Manager will send an additional variable in the post called “dkey” that will contain the private key entered by the user or stored in the shopping cart data. Should the script fail to decrypt the payment data, do not have the script throw an error. Instead, pass the text **** SECURED **** in one or more of the payment fields. The Order Manager will prompt the user that the data could not be decrypted but will give the user the option of importing the order data anyway.

IMPORTANT NOTE: Effective with the v5.000 release of the Order Manager, all passwords and encryption keys pertaining to shopping carts are encrypted in the Order Manager’s data file for security purposes.

Form Variables for order download:

Variable Name	Value	From	Required
setifunction	downloadorders	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
lastorder	The last order number in Order Manager received from this store. In the absence of an order number Order Manager will pass the text "All"	Retrieved programmatically at the time of download. Use this OR lastdate to identify which new orders to download.	Yes - if the lastdate variable is empty
lastdate	The date of the last order downloaded from this store. In the absence of an order date Order Manager will pass the text "All"	Retrieved programmatically at the time of download. Use this OR lastorder to identify which new orders to download. If using this as the determining factor, always include all orders from this date.	Yes - if the lastorder variable is empty
startnum	If the number of orders exceeds the maximum allowed for download, this parameter will contain the record number to start with. This will not be present if all orders can be downloaded in a single call.	For Example: if you have 100 orders and allow a maximum of 35 per download then in the first call startnum will be - 1, the second call - 36, the third call - 71. Calculated programmatically.	Yes – if there are more orders to download than the MaxOrderDownload parameter specifies
batchsize	The total number of orders to download in one export. May exceed the available number of orders. This will not be present if all orders can be downloaded in a single call.	Value of the GenericMaxOrderDownload system parameter (default 100).	Yes – if there are more orders to download than the MaxOrderDownload parameter specifies
dkey	Key used to decrypt payment data (if supported by your shopping cart). This parameter will be sent if the "Cart Uses Data Encryption" option is selected in the Shopping Cart setup	Contained in the Shopping Cart setup data OR entered at the time of download.	NO
omversion	Version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

In the examples below, revise the following data points prior to testing:

Replace "www.mystore.com/script.ext:" with the appropriate URL pointing to the script on your web server.

Replace "auser" with the proper user name for login on the website.

Replace "pwd" with the proper password for login to the website.

Replace "mystore" with the store code identifier (optional).

Replace "1001" with an order number or the word "All".

Replace "10-Jun-2003" with an appropriate date to start order download.

If using data encryption, include the "dkey" field and replace "decryptionkey" with the appropriate private key

It is recommended to include, exclude and change the startnum and batchsize values to verify your script properly adheres to the download cycling principles.

It is recommended to change the lastorder and lastdate values to the text "All" to verify your script does not fail when a date or number is not passed. Also use different dates or order numbers during testing.

Example URL for testing from browser – uses "GET" method:

After making revisions detailed above, copy the entire line and paste it into the address bar of your browser.

```
https://www.mystore.com/script.ext?setifunction=downloadorders&setiuser=auser&password=pwd
&code=mystore&lastorder=1001&lastdate=10-Jun-2003&startnum=1&batchsize=100&dkey=decryptionkey
&omversion=5.000
```

Example Form Post – uses "POST" method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the "Send" button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="downloadorders">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "lastorder" VALUE = "1001">
  <INPUT TYPE=TEXT NAME = "lastdate" VALUE = "10-Jun-2003">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "dkey" VALUE = "decryptionkey">
  <INPUT TYPE=TEXT NAME = "startnum" VALUE = "1">
  <INPUT TYPE=TEXT NAME = "batchsize" VALUE = "100">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Expected Response:

The script should respond with an XML document defined in the OrderImport.xsd specification.

Example:

```
<?xml version="1.0"?>
<SETIOrders>
  <Response>
    <ResponseCode>1</ResponseCode>
    <ResponseDescription>Success</ResponseDescription>
  </Response>
  ...ONE OR MORE ORDER ELEMENTS IN THIS SECTION IF EXPORT SUCCESSFUL...
  <Order>
    ...
  </Order>
</SETIOrders>
```

The ResponseCode element should be set to 1 and the ResponseDescription to "Success" if there are orders available to process. If no orders are present in the file, the merchant will receive an error response from the Order Manager. Set the

ResponseCode to 2 and the ResponseDescription to "Success" if there are NO orders available to process. Set the ResponseCode to 3 should an error occur during script execution. Set the ResponseDescription to the error message to deliver to the merchant.

Details:

Order Manager will test the incoming order data to see if it is legitimate. If so, the three phase import will begin:

Phase 1 – Data Download: The downloaded data will be written to an XML file(.xml) in the NewOrders directory specified in the NewOrderLocation system parameter. This action will continue until all order data is retrieved from the website.

Phase 2 – Order Import: The data is read from the .xml file(s) on hard drive and added to the store data file. If the import is unsuccessful for any reason (error) during this phase, the imported data will be removed from the system (rolled back) and the file will be left in the NewOrders directory for the user to address. The file can be opened in any text editor (Microsoft's WordPad is recommended) and modified to eliminate error conditions (if possible). Contact support at Stone Edge Technologies, Inc. for questions regarding import errors. We attempt to anticipate a variety of error possibilities, however, since the general public is entering the data in the website, we cannot guarantee that it will be without problems. If using SQL Express or SQL Server for data storage with The Order Manager Enterprise Level, data rollback does not occur if an error is triggered. The imported orders and order details must be physically deleted from the SQL tables to recover from the error.

Phase 3 – Order Processing: The program will begin to "process" the orders just imported. Should an error occur at this point in the process, the data cannot be rolled back (all Order Manager versions). The program attempts to recover from the error condition and continue to process the remaining orders, however, this may not always be possible. If the error is severe, processing will stop at the point of the error. This will leave a number of orders "unprocessed". The easiest way to determine if an order is unprocessed is 1) order will be marked as Approved, 2) all line items will be backordered (marked as needed), 3) there will be no transactions written and 4) the customer ID recorded on the order is zero (0). These orders should be deleted from the system and re-imported, you should not attempt to work with them! If the import is successful then the .xml file will be moved from the NewOrders directory to the DataArchives directory specified in the ArchiveLocation system parameter. This way you will have a backup of each download available for data restoration purposes. Since these files may contain sensitive customer data, you may want to remove or eliminate the files in the data archive once you determine that the data is no longer needed or set the system DeleteDownloadTextFiles parameter to delete the files from the NewOrders directory after import.

XML Order Import Specification

The Order Manager XML Import Specification defines the structure of the order data that should be returned from the website. This section details the elements and their various data points.

For specific detail on the XML Specification, see the OrderImport.xsd schema file included with the Developer's Guide.

The Order Element is the container element for an order. This element can appear multiple times in the download.			
Element Name	Description	Restrictions	Required
Order	Container element for a single order		YES if Response Code is set to 1
Order/OrderNumber	Contains the website's order number	If GenericOrderNumberType parameter set to "Text" then 50 character limit. If set to "Numeric" then values between 1 and 2147483642	YES
Order/OrderDate	The date the order was placed. Must be in an unambiguous date/time format such as mmm/dd/yyyy – can also include a time stamp in hh:mm:ss format	Date Time value	YES
Order/MarketName	If using a cart system that collects orders from multiple sources such as Channel Advisor or Monsoon, this element should contain the name of the market that sourced the order – for example: AmazonMarketplaceUS, Ebay	50 Character Limit	NO
Order/MarketOrderID	This element should contain the Market's Order Identifier	100 Character Limit	NO
Order/MarketCustomerID	This element should contain the Market's Customer Identifier	200 Character Limit	NO
Order/OrderStatus	Status of the order at the website at time of export	50 Character Limit	NO

The Billing element is the container element for the billing party name and address information.
This element appears only once within the Order element.

Element Name	Description	Restrictions	Required
Order/Billing	Container Element for billing party information		YES
Order/Billing/ FullName	The Billing Party Name	255 characters	YES
Order/Billing/ Company	The billing party company name	255 characters	NO
Order/Billing/ Phone	The billing party phone number	255 characters	NO
Order/Billing/ Email	The Billing Party Email Address	255 characters	NO
Order/Billing/ Address	Container Element for address information		YES
Order/Billing/ Address/Street1	Street address of the billing party	255 characters	YES
Order/Billing/ Address/Street2	Street address 2 of the billing party	255 characters	NO
Order/Billing/ Address/City	Name of city for the billing party	255 characters	YES
Order/Billing/ Address/State	State/Province code for the billing party	2 characters	YES
Order/Billing/ Address/Code	The Postal Code for the billing party	255 characters	YES
Order/Billing/ Address/Country	The country code for the billing party	2 characters	NO

The Shipping Element is the container element for the shipping party name and address information as well as the list of products purchased on the order.
This element must appear only once in the Order element.

Element Name	Description	Restrictions	Required
Order/Shipping	Container Element for shipping party and order detail information		YES
Order/Shipping/ FullName	The shipping party name	255 characters	YES
Order/Shipping/ Company	The shipping party Company name	255 characters	NO
Order/Shipping/ Phone	The phone number of the shipping party	255 characters	NO
Order/Shipping/ Email	The email address of the shipping party	255 characters	NO
Order/Shipping/ Address	Container element for the shipping party address		YES
Order/Shipping/ Address/Street1	The Street Address of the shipping party	255 characters	YES
Order/Shipping/ Address/Street2	The street address 2 of the shipping party	255 characters	NO
Order/Shipping/ Address/City	The city name for the shipping party	255 characters	YES
Order/Shipping/ Address/State	The state/province code of the shipping party	2 characters	YES
Order/Shipping/ Address/Code	The postal code of the shipping party	255 characters	YES
Order/Shipping/ Address/Country	The country code of the shipping party	2 characters	NO

The Product Element is the container element for an item purchased. This element can appear multiple times within the Shipping element.			
Element Name	Description	Restrictions	Required
Order/Shipping/Product	Container element for the product purchased		YES
Order/Shipping/Product/SKU	Product Identification Code	255 characters	YES
Order/Shipping/Product/Name	The short description for the product	255 characters	YES
Order/Shipping/Product/Quantity	The quantity of the item purchased	Integer value	YES
Order/Shipping/Product/ItemPrice	The price per unit of the item purchased. If you sell items with options that have price modifiers, the inclusion/exclusion of the option price modifiers in the reported price per unit must be defined by the OptionPriceInItemPrice parameter	Decimal value No currency symbol	YES
Order/Shipping/Product/Weight	The weight per unit	Decimal value No weight Unit Symbol	NO
Order/Shipping/Product/ProdType	The type of product	Limited to "Tangible" or "Download"	NO
Order/Shipping/Product/Taxable	Indicates whether the product is or is not taxable (overrides the Order Manager's internal settings if present)	Limited to "Yes" or "No"	NO
Order/Shipping/Product/ CustomerText	Any comments or instructions from the customer for this item		NO
Order/Shipping/Product/LineID	Line item ID from shopping cart Note: Used for Status Updates back to website	50 characters	NO
Order/Shipping/Product/MarketLineID	If using a cart system that collects orders from multiple markets such as Channel Advisor or Monsoon then this element should contain the source market's line item identifier.	200 Characters	NO
Order/Shipping/Product/ItemStatus	The status of the line item at time of order export	50 Characters	NO
Order/Shipping/Product/ FulfillmentCenter	If the item is being shipped from a fulfillment center and the fulfillment center was notified by the shopping cart, enter the center's Name or ID	20 Characters	NO
Order/Shipping/Product/Total	The extended price (Quantity * (Price Per Unit + Option Price Modifiers))	Decimal value No currency symbol	NO

Order/Shipping/ Product/Dimensions	Container element for the shipping dimensions of the item		NO
Order/Shipping/ Product/Dimensions/ Length	Length of the item (longest dimension)	Decimal value No Measure Unit Symbol	YES if Dimensions Element present
Order/Shipping/ Product/Dimensions/ Width	Width of the item	Decimal value No Measure Unit Symbol	YES if Dimensions Element present
Order/Shipping/ Product/Dimensions/ Height	Height of the item	Decimal value No Measure Unit Symbol	YES if Dimensions Element present
Order/Shipping/ Product/OrderOption	This is the container element for options pertaining to the product		NO
Order/Shipping/ Product/OrderOption/ OptionName	The name of the attribute such as "Size", "Color" or "Style"	50 characters	NO
Order/Shipping/ Product/OrderOption/ SelectedOption	The option choice made by the customer ("Large", "Blue", "Sleeveless", etc.)		YES
Order/Shipping/ Product/OrderOption/ OptionPrice	The amount, per unit, that this option impacts the price of the item	Decimal value No currency symbol	NO
Order/Shipping/ Product/OrderOption/ OptionCode	Any text to add to the parent product SKU to identify this option	50 characters	NO
Order/Shipping/ Product/OrderOption/ OptionType	Type of option – is the attribute a selectable type for impacting variant products	Limited to = "select", "radio", "text", "memo", "checkbox"	NO
Order/Shipping/ Product/OrderOption/ OptionWeight	The amount, per unit, that this option impacts the weight of the item	Decimal value No Currency Symbol	NO
Order/Shipping/ Product/OrderOption/ OptionCost	The amount, per unit, that this option impacts the cost of the item	Decimal value No Currency Symbol	NO

The Payment Element is the container element for any payments recorded on the order (except Gift Certificates). This element is optional and can appear more than once within the Order element			
Element Name	Description	Restrictions	Required
Order/Payment	<p>Container Element for Payment information</p> <p>Various Payment Methods' XML structure is detailed in the following sections</p>	<p>If present, must only contain one of the following payment types – other types are ignored.</p> <p>CreditCard eCheck PayPal COD Check PurchaseOrder Generic1 Generic2 GiftCard StoreCredit</p>	NO

The Credit Card Element is the container element for any credit card payments recorded on the order.
This element is optional and can appear only once in a Payment Element

Element Name	Description	Restrictions	Required
Order/Payment/ CreditCard	This is the container element for credit card payments		NO
Order/Payment/ CreditCard/Issuer	This is the card type used for the payment (e.g. Visa, Mastercard, Amex, Discover, etc.)	Value must match a "credit card" payment method name in the Order Manager	YES if full card number not present
Order/Payment/ CreditCard/Number	Credit Card Account number.	4-20 characters	YES if card issuer not present
Order/Payment/ CreditCard/ ExpirationDate	Card expiration date	Limited to MMY, MMY, MM/YY, MM-YY, MM-YYYY or MM/YYYY formats	NO
Order/Payment/ CreditCard/ VerificationValue	The results of the Card Security Code test	Limit to a single character of one of the following: M, N, P, S, U	NO
Order/Payment/ CreditCard/ FullName	Cardholder Name	255 characters	NO
Order/Payment/ CreditCard/ Company	Cardholder Company Name	255 characters	NO
Order/Payment/ CreditCard/ BankName	Name of card issuing bank	50 Characters	NO
Order/Payment/ CreditCard/ OrderProcessingInfo	Payment information received from processing bank	255 characters	NO
Order/Payment/ CreditCard/ AVS	The results of the Address Verification test	Limit to 1 - 2 characters from one of the following: A, B, C, D, E, G, I, M, N, P, R, S, U, W, X, Y, Z	NO
Order/Payment/ CreditCard/ TransID	The Transaction ID if the card was processed at the website (authorization or sale)	100 characters	NO*
Order/Payment/ CreditCard/ AuthCode	The Authorization Code if the card was processed at the website (authorization or sale)	100 Characters	NO

Order/Payment/ CreditCard/ ProcessLevel	NOT YET SUPPORTED: The level of transaction processed against the card	Limit to "None", "Auth Only", "Capture"	NO
Order/Payment/ CreditCard/ Amount	Dollar amount of transaction	Decimal value	NO
Order/Payment/ CreditCard/ StartDate	The Start Date of the card (UK Credit Cards Only)	Limit to MMYT or MMYYYY, MM/YY, MM-YY, MM/YYYY, MM-YYYY formats	NO
Order/Payment/ CreditCard/ IssueNumber	The Issue Number of the card (UK Credit Cards Only)	2 Digits	NO
Order/Payment/ CreditCard/ SecurityKey	The security key if the transaction was run at the website (Protx gateway only) OR The transaction's "Order Number" (Linkpoint gateway only)	50 characters	NO
Order/Payment/ CreditCard/ CAVV	The Verified By Visa result code	Limit to 1 character from the following: 0-9, A, B	NO
Order/Payment/ CreditCard/ ECI	The Verified By Visa Authentication Code	Limit to 1 number from the following: 0-9	NO
Order/Payment/ CreditCard/ XID	The Verified by Visa transaction code	50 characters	NO

*Although this field is not required, if no TransID is provided a record is not written to the Order Manager's Transactions table and there is no data to display at the Payment tab of the Process Orders form.

eCheck Element is the container element signaling an eCheck payment.
This element is optional and can appear only once in a Payment Element

Element Name	Description	Restrictions	Required
Order/Payment/ eCheck	Container element for eCheck payments		NO
Order/Payment/ eCheck/ RoutingNumber	The ABA (American Banking Assn) number or Bank Routing Number	50 characters	YES
Order/Payment/ eCheck/ AccountNumber	The Checking Account Number	50 characters	YES
Order/Payment/ eCheck/ AccountType	The type of account used	Limit to "Checking", "Savings", "BusinessChecking" or "BusinessSavings"	NO gateway dependent
Order/Payment/ eCheck/ BankName	The name of the bank the account is drawn against	50 characters	No gateway dependent
Order/Payment/ eCheck/ BankState	The state code where the bank the account is drawn against resides	2 characters	No gateway dependent
Order/Payment/ eCheck/ NameOnAccount	Account holder's name	50 characters	No gateway dependent
Order/Payment/ eCheck/ CheckNumber	The document number of the physical check	50 characters	No gateway dependent
Order/Payment/ eCheck/ IDState	State Code the account holder's ID is provided by	2 characters	No gateway dependent
Order/Payment/ eCheck/ IDNumber	Account Holder's ID number	50 characters	No gateway dependent

The PayPal Element is the container element signaling a PayPal DirectPayment or ExpressCheckout payment through PayPal Website Payments Pro or other payment process. This element is optional and can appear only once in a Payment Element			
Element Name	Description	Restrictions	Required
Order/Payment/ PayPal/	Container element for PayPal payment information		NO
Order/Payment/ PayPal/Payer	Container element for payer information	50 characters	NO
Order/Payment/ PayPal/Payer/ID	Payer's PayPal ID	50 characters	NO
Order/Payment/ PayPal/Payer/ PayerStatus	The PayPal status for the Payer	Limit to "verified" or "unverified"	NO
Order/Payment/ PayPal/Payer/ Prefix	The name prefix of the payer (Mr., Mrs., etc.)	255 characters	NO
Order/Payment/ PayPal/Payer/ FirstName	Payer's first name	255 characters	NO
Order/Payment/ PayPal/Payer/ MiddleName	Payer's Middle Name	255 characters	NO
Order/Payment/ PayPal/Payer/ LastName	Payers Last Name	255 characters	NO
Order/Payment/ PayPal/Payer/ Suffix	Name suffix of Payer (Jr., Sr., etc.)	255 characters	NO
Order/Payment/ PayPal/Payer/ Email	Payer's Email address	255 characters	NO
Order/Payment/ PayPal/Payer/ Company	Payer's Company Name	255 characters	NO
Order/Payment/ PayPal/Payer/ AddressStatus	PayPal's Address Status for Payer	Limit to "none", "confirmed", "unconfirmed"	NO
Order/Payment/ PayPal/ Transaction	Container element for the transactional data		NO

Order/Payment/ PayPal/ Transaction/ TransID	PayPal transaction ID number	50 characters	NO
Order/Payment/ PayPal/ Transaction/ Status	Status of the PayPal payment	Limit to: "none", "cancelled-reversal", "completed", "denied", "expired", "failed", "pending", "refunded", "reversed", "processed", "voided"	NO
Order/Payment/ PayPal/ Transaction/ Amount	Dollar amount of PayPal transaction	Decimal value No currency symbol	NO
Order/Payment/ PayPal/ Transaction/ TransDate	Date of the PayPal Transaction	Date/Time in yyyy-mm-ddThh:mm:ssZ format	NO
Order/Payment/ PayPal/ Transaction/ ProcessingFee	Amount of fees charged by PayPal for the transactions	Decimal value No currency symbol	NO
Order/Payment/ PayPal/ Transaction/ TaxAmount	Amount of taxes paid on the transaction	Decimal value No currency symbol	NO
Order/Payment/ PayPal/ Transaction/ ReasonCode	If the Pending Reason is "reversal", provide a reason code defining the reversal	Limit to: "none", "chargeback", "guarantee", "buyer-complaint", "refund", "other"	NO
Order/Payment/ PayPal/ Transaction/ PendingReason	If the Status of the transaction is "pending" provide the reason	Limit to: "none", "authorization", "address", "echeck", "intl", "multi-currency", "verify", "other"	NO
Order/Payment/ PayPal/ Transaction/ Issuer	If payment processed through PayPal Website Payments Pro, provide the card type used for the transaction ("Visa", Mastercard", etc.)	Must match a "credit card" payment method in the Order Manager	NO
Order/Payment/ PayPal/ Transaction/ Number	Card number if processed by Website Payments Pro.	4-20 characters	NO
Order/Payment/ PayPal/ Transaction/ ExpirationDate	If payment processed through PayPal Website Payments Pro, provide the card Expiration date	Limit to MMY, MMYYYY, MM/YY, MM-YY, MM/YYYY or MM-YYYY formats	NO

Order/Payment/ PayPal/ Transaction/ AVS	If payment processed through PayPal Website Payments Pro, provide the AVS results for the card transaction	Limit to 1 - 2 characters from one of the following: A, B, C, D, E, G, I, M, N, P, R, S, U, W, X, Y, Z	NO
Order/Payment/ PayPal/ Transaction/ CVV2	If payment processed through PayPal Website Payments Pro, provide the Card Security test results for the card transaction	Limit to a single character of one of the following: M, N, P, S, U	NO
Order/Payment/ PayPal/ Transaction/ StartDate	If processed via Website Payments Pro, Card Start Date for UK cards	Limit to MMY, MMY, MM/YY, MM-YY, MM/YYYY or MM-YYYY formats	NO
Order/Payment/ PayPal/ Transaction/ IssueNumber	If processed via Website Payments Pro, Card Issue Number for UK cards	2 Digits	No

COD Element is the container element signaling a Cash on Delivery payment. This element has no children elements
This element is optional and can appear only once in a Payment Element

Element Name	Description	Restrictions	Required
Order/Payment/COD	Signals a COD payment	No children elements	NO

The PurchaseOrder Element is the container element signaling that the payment will be made on a PO.
This element is optional and can appear only once in a Payment Element

Element Name	Description	Restrictions	Required
Order/Payment/ PurchaseOrder	Container element signaling payment will be made by PO		NO
Order/Payment/ PurchaseOrder/ PurchaseNumber	PO Number or Customer Purchase Number	50 characters	NO

The Check Element is the container element signaling that a check is to be mailed for payment or can provide the necessary data for recording an eCheck
This element is optional and can appear only once in a Payment Element

Element Name	Description	Restrictions	Required
Order/Payment/ Check	Container element that signals a Check payment		NO
Order/Payment/ Check/ RoutingNumber	ABA or Routing number on the check	50 characters	NO
Order/Payment/ Check/ AccountNumber	Account Number on the check	50 characters	NO
Order/Payment/ Check/ CheckNumber	The document number of the physical check	50 characters	NO

The GiftCard Element is the container element signaling that the payment was made by a GiftCard (non credit card type gift card).
This element is optional and can appear only once in a Payment Element

Element Name	Description	Restrictions	Required
Order/Payment/ GiftCard	Container element signaling payment was made by Gift Card		NO
Order/Payment/ GiftCard/PayType	Name of the Gift Card that should match payment method in the Order Manager		NO
Order/Payment/ GiftCard/Number	Card Number	4-20 characters	NO
Order/Payment/ GiftCard/PIN	Card PIN (Personal ID Number)	4 digits	NO
Order/Payment/ GiftCard/FaceValue	Value of the Gift Card	Decimal value No currency symbol	NO
Order/Payment/ GiftCard/ ExpirationDate	Card expiration date	Limit to MMY, MMYYYY, MM/YY, MM-YY, MM/YYYY, MM-YYYY formats	NO
Order/Payment/ GiftCard/ TransID	Transaction ID if card was processed at the website	100 characters	NO
Order/Payment/ GiftCard/AuthCode	Authorization Code if transaction was run at the website	100 characters	NO

The StoreCredit Element is the container element signaling that the payment was made from a Store Credit or Member Rewards program.
This element is optional and can appear only once in a Payment Element

Element Name	Description	Restrictions	Required
Order/Payment/ StoreCredit	Container element signaling payment was made via store credit		NO
Order/Payment/ StoreCredit/Total	Amount of the credit	Decimal value No currency symbol	NO
Order/Payment/ StoreCredit/ Description	Brief description of the credit type	255 characters	NO

The Generic1 and Generic2 Elements are the container element used for any other type of payment not already supported. These elements allow the delivery of a payment method name, description and up to 4 data points regarding the payment transaction. The example here is for Generic1 and uses the same structure as Generic2.
This element is optional and can appear only once in a Payment Element

Element Name	Description	Restrictions	Required
Order/Payment/ Generic1	Container element for generic payment types		NO
Order/Payment/ Generic1/Name	The Name of the Payment Method as it appears in the Order Manager's Payment Methods list	Must match a payment method in the Order Manager	NO
Order/Payment/ Generic1/ Description	Brief Description of the method	255 characters	NO
Order/Payment/ Generic1/Field1	First data point – placed into Pay1 field in the Orders table	255 characters	NO
Order/Payment/ Generic1/Field2	Second Data Point – placed into the Pay2 field of the Orders table	255 characters	NO
Order/Payment/ Generic1/Field3	Third data point – placed into the Pay3 field of the Orders table	255 characters	NO
Order/Payment/ Generic1/Field4	Fourth data point – placed into the Pay4 field of the Orders table	255 characters	NO

The Totals Element is the container element for various totals on the order			
Element Name	Description	Restrictions	Required
Order/Totals	Container Element for order totals (see: Total Element)		YES
Order/Totals/ProductTotal	Extended amount for all products without additional discounts or charges applied	Decimal value No currency symbol	YES
Order/Totals/Discount	Container element for discount information. One or more may be present in the Totals element		NO
Order/Totals/Discount/Type	Specifies whether the discount is a percentage or dollar amount	Limit to "Flat" or "Percent"	NO – flat assumed
Order/Totals/Discount/Description	Brief description of discount	255 characters	NO
Order/Totals/Discount/Percent	If Type=Percent then the actual or integer value of discount percent (e.g. for 5% then return "5" or ".05")	Decimal value No currency symbol	NO
Order/Totals/Discount/Amount	Dollar amount of discount	Decimal value No currency symbol	YES, if a Discount is provided
Order/Totals/Discount/ApplyDiscount	Signals whether discount is applied before of after tax calculation	Limit to "Pre" or "Post"	NO – pre assumed
Order/Totals/SubTotal	Dollar Amount of the Line Items after discount	Decimal value No currency symbol	NO
Order/Totals/Tax	Container element for tax information		NO
Order/Totals/Tax/TaxAmount	Dollar amount of taxes charged on the order	Decimal value No currency symbol	YES of Tax element present
Order/Totals/Tax/TaxRate	Tax percentage applied in actual or integer value (e.g. 6.5% can be 6.5 or .065)	Decimal value No currency symbol	NO
Order/Totals/Tax/TaxShipping	Signals whether shipping was included in tax calculations	Limit to "Yes" or "No"	NO – Yes assumed
Order/Totals/Tax/TaxExempt	Flag specifying whether billing party is tax exempt	Limit to "Yes", "No"	NO – No assumed
Order/Totals/Tax/TaxID	If TaxExempt is Yes, then this element should have the Tax Payer ID number	50 characters	NO
Order/Totals/GrandTotal	Dollar amount for the entire order	Decimal value No currency symbol	YES
Order/Totals/Surcharge	Container element for surcharge information. More than one can be present in the Totals element		NO

Order/Totals/ Surcharge/Total	Dollar amount of the surcharge	Decimal value No currency symbol	YES if Surcharge element present
Order/Totals/ Surcharge/ Description	Brief description of the surcharge	255 characters	NO
Order/Totals/ ShippingTotal	Container element for shipping method and cost		NO
Order/Totals/ ShippingTotal/Total	Dollar amount for shipping charges	Decimal value No currency symbol	Yes if Shipping Total element present
Order/Totals/ ShippingTotal/ Description	Brief description of shipping method	255 characters	NO

The Coupon Element is the container element for Coupons applied to the order
This element may appear multiple times.

Element Name	Description	Restrictions	Required
Order/Coupon	Container Element for coupon details		NO
Order/Coupon/Name	Name of the Coupon	50 characters	NO
Order/Coupon/ Status	Brief information regarding coupon	255 characters	NO
Order/Coupon/Total	Dollar amount of coupon	Decimal value No currency symbol	YES if Coupon element present
Order/Coupon/ ApplyCoupon	Flag specifying if coupon is applied before or after tax calculation	Limit to "Pre" or "Post"	NO – pre is assumed

The GiftCertificate Element is the container element for details regarding the redemption of gift certificate
This element can appear multiple times.

Order/ GiftCertificate	Container Element for certificate redemption. One or more certificates can be present		NO
Order/ GiftCertificate/ Name	Name of the certificate	255 characters	NO
Order/ GiftCertificate/ Status	Brief description of the certificate	255 characters	NO
Order/ GiftCertificate/ Total	Dollar amount redeemed	Decimal value No currency symbol	YES of GiftCert element present

The Other Element is the container element for additional details regarding the order This element must appear only once for an order			
Order/Other	Container Element for additional order information		YES
Order/Other/ Associate	Reference Number or Name of an affiliate or sales person	50 characters	NO
Order/Other/ OrderInstructions	Order instructions from customer		NO
Order/Other/ Comments	Comments from customer		NO
Order/Other/ IPHostName	IP address of customer	255 characters	NO
Order/Other/ TotalOrderWeight	Total weight of items purchased	Decimal value	NO
Order/Other/ WebCustomerID	Website customer ID number	integer	NO
Order/Other/ EmailList	Any text in this field triggers the “add to email list” flag in new customer record		NO
Order/Other/ GiftMessage	Customer’s gift message		NO
Order/Other/ NoteToCustomer	Text to display on invoice		NO
Order/Other/ CustomCheckout Field	The CustomCheckoutField element is a container element used to pass additional data points from the website to the Order Manager		NO
Order/Other/ CustomCheckout Field/FieldName	Name of the custom field – Used for matching labels applied to Custom Order Fields in the Order Manager	255 characters	YES if parent field present
Order/Other/ CustomCheckout Field/FieldValue	Data contained in the custom field	255 characters for custom text fields	YES if parent field present

Example XML Response for Order Download

NOTE: The line indentations and breaks need not be in the real data but are used here for clarity purposes to show parent, child and sibling relationships in the XML.

```
<?xml version="1.0"?>
<SETIOrders>
  <Response>
    <ResponseCode>1</ResponseCode>
    <ResponseDescription>success</ResponseDescription>
  </Response>
  <Order>
    <OrderNumber>1261</OrderNumber>
    <OrderDate>2003-03-05 21:04:13</OrderDate>
    <Billing>
      <FullName>Kevin Smith</FullName>
      <Company>Stone Edge Technologies Inc.</Company>
      <Phone>215-641-1837</Phone>
      <Email>kevin@stoneedge.com</Email>
      <Address>
        <Street1>One Valley Square</Street1>
        <Street2>Suite 130</Street2>
        <City>Blue Bell</City>
        <State>PA</State>
        <Code>19422</Code>
        <Country>US</Country>
      </Address>
    </Billing>
    <Shipping>
      <FullName>Kevin Smith</FullName>
      <Company>Stone Edge Technologies, Inc.</Company>
      <Phone>215-641-1837</Phone>
      <Email>Kevin@stoneedge.com</Email>
      <Address>
        <Street1>One Valley Square</Street1>
        <Street2>Suite 130</Street2>
        <City>Blue Bell</City>
        <State>PA</State>
        <Code>19422</Code>
        <Country>US</Country>
      </Address>
    <Product>
      <SKU>SHRT</SKU>
      <Name>MyShirt</Name>
      <Quantity>1</Quantity>
      <ItemPrice>5.00</ItemPrice>
      <Weight>2</Weight>
      <ProdType>Tangible</ProdType>
      <Taxable>Yes</Taxable>
      <CustomerText>Gift wrap this item please</CustomerText>
      <LineID>125487</LineID>
      <Total>5.00</Total>
      <Dimension>
        <Length>10</Length>
        <Width>2</Width>
        <Height>6</Height>
      </Dimension>
      <OrderOption>
        <OptionName>Size</OptionName>
        <SelectedOption>DS-DivSmall</SelectedOption>
        <OptionPrice>0</OptionPrice>
        <OptionCode>DS</OptionCode>
        <OptionType>select</OptionType>
        <OptionWeight>0</OptionWeight>
      </OrderOption>
    </Product>
  </Order>
</SETIOrders>
```

```

    <OrderOption>
      <OptionLabel>Color</OptionLabel>
      <SelectedOption>RD-Red</SelectedOption>
      <OptionPrice>0</OptionPrice>
      <OptionCode>RD</OptionCode>
      <OptionType>select</OptionType>
      <OptionWeight>0</OptionWeight>
    </OrderOption>
  </Product>
</Shipping>
<Payment>
  <CreditCard>
    <Issuer>Visa</Issuer>
    <Number>4111111111111111</Number>
    <ExpirationDate>05/2007</ExpirationDate>
    <VerificationValue>Y</VerificationValue>
    <FullName>Kevin Smith</FullName>
    <Company>Stone Edge Technologies, Inc.</Company>
    <OrderProcessingInfo>*ANET*:1:123456,P,This transaction has been approved.:</OrderProcessingInfo>
    <AVS>YYY</AVS>
    <TransID>4729238728739452876</TransID>
    <AuthCode>123456</AuthCode>
    <ProcessLevel>Auth Only</ProcessLevel>
  </CreditCard>
</Payment>
<Totals>
  <ProductTotal>25.00</ProductTotal>
  <Discount>
    <Type>Flat</Type>
    <Amount>5.00</Amount>
    <ApplyDiscount>Pre</ApplyDiscount>
  </Discount>
  <Subtotal>20.00</Subtotal>
  <Tax>
    <TaxShipping>No</TaxShipping>
    <TaxAmount>2.68</TaxAmount>
    <TaxRate>5.00</TaxRate>
  </Tax>
  <GrandTotal>64.43</GrandTotal>
  <ShippingTotal>
    <Total>8.25</Total>
    <Description>Ground</Description>
  </ShippingTotal>
</Totals>
<Other>
  <Associate/>
  <OrderInstructions>Please deliver between 2:00 and 5:00 PM</OrderInstructions>
  <Comments>Additional Comments Section at Checkout</Comments>
  <IpHostname>IP Address:151.197.12.62, Host Name:pool-151-197-12-62.phil.east.verizon.net</IpHostname>
  <TotalOrderWeight>4</TotalOrderWeight>
  <EmailList>Add me to your E-mailing list</EmailList>
  <CustomCheckoutField>
    <FieldName>BuyFor</FieldName>
    <FieldValue>Own Store</FieldValue>
  </CustomCheckoutField>
  <CustomCheckoutField>
    <FieldName>HeardAbout</FieldName>
    <FieldValue>Through The Grapevine</FieldValue>
  </CustomCheckoutField>
  <CustomCheckoutField>
    <FieldName>PublicationsRead</FieldName>
    <FieldValue>Ecom Extras</FieldValue>
  </CustomCheckoutField>
</Other>
</Order>
</SETIOrders>

```

Downloading Customer Information:

Order Manager can collect customer data from your shopping cart application by going to the Cart Functions section on the Maintenance Menu. The returned data can populate or update the internal customer information. Record matching can be done against the Customer's ID number in the Shopping cart along with email, phone number or name, address and zip. If a match is found, the record can be updated with the incoming information. If the record is not found, a new customer is added to the system. To configure the customer matching system, see the CustomerSearchField# parameters in the Program group. The default of the program is to match first on email then on phone.

By default the Cart Functions option will not be displayed on the Maintenance Menu for Generic Cart Users. To allow this option to be displayed, set the system parameter GenericUseAdvancedFunctions to True. From the Cart Functions screen, you can download your customer data directly into Order Manager's Customers table. The download is a three step process:

1. Order Manager requests the script's version number from the website. This is done for general testing of communications between Order Manager and the script file and also to ensure that if structural changes are made to Order Manager to support a new script that you can test to see if the latest script is installed before running the download.
2. Order Manager then requests the total number of customers that need to be downloaded from the site. Typically a download of 500 customers or less causes no problems, however, if there is a large number to retrieve, you may run in to timeout issues on the server as the script executes. There is a field on the Cart Functions screen labeled "Maximum number of products or customer to download at one time" that allows you to set the maximum number of customers per call. If the total number of customers to download exceeds the max download value, the Order Manager will call for several downloads, retrieving the maximum number of customers in each call until all customers are received.
3. Once Order Manager determines how many times it will execute the download, it begins the download process. The data received by Order Manager is fed into one or more XML files that are stored in the DataArchives directory (identified in the system parameter ArchiveLocation). Once the download(s) is complete, Order Manager will test and import the customer file(s).

1) Requesting Script Version Data: This function should **not** require user name and password validation to receive the response and should also be allowed to run if the connection is not secure (should respond to both http and https requests). This enables quick testing by Stone Edge Technologies, Inc. when troubleshooting web communications.

Form Variables:

Variable Name	Value	From	Required
setifunction	sendversion	Hard coded into visual basic, cannot be modified	YES
omversion	The version number of the Order Manager being used	Value of the Version field in the last record in the Version table – Used for version control	NO

For the examples below, Replace www.mystore.com/script.ext with the appropriate URL pointing to the script on your web server

Example URL for testing from browser – uses "GET" method:

After making the revisions detailed above, copy the entire line and paste it into the address bar of your browser.

<https://www.mystore.com/script.ext?setifunction=sendversion&omversion=5.000>

Example Form Post – uses the “POST” method:

Save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the “Send” button to send the request.

```
<FORM ACTION="https://www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE ="sendversion">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE ="5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Expected Response:

SETIResponse: version=#

The # symbol above represents the version of the script file on the website in ##### or #.### format. The version number must be 4 numbers with a decimal. For example SETIResponse: version=1.000

Details:

Order Manager will test the returned value against the GenericScriptVersionNumber system parameter. If the returned value is less than the parameter value, Order Manager will notify the user that the script version is too low and will abort the order import. This provides a method of version control for new features or bug fixes.

2) Determining the number of customers to download

This function should require user name and password validation prior to a response and should also test that the connection is secure. For added security, a store “code” can be added to identify which store’s data you’ll need. If user validation or the test for a secure connection fail, an error response should be generated

Order Manager will post a request for the total number of customers available for download. The total count of customer records must be returned as an integer value.

Form Variables:

Variable Name	Value	From	Required
setifunction	getcustomerscount	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	Optional: A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
omversion	The version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

In the examples below revise the following data points prior to testing:

Replace “www.mystore.com/script.ext” with the appropriate URL pointing to the script on your web server.

Replace “auser” with the proper user name for login on the website.

Replace “pwd” with the proper password for login to the website.

Replace “mystore” with the store code identifier (optional).

Example URL for testing from browser – uses “GET” method:

https://www.mystore.com/script.ext?setifunction=getcustomerscount&setiuser=*auser*&password=*pwd*&code=*mystore*&omversion=5.000

Example Form Post – uses “POST” method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the “Send” button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="getcustomerscount">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Note: If you have your web server set up to validate the request, you may need to enter your user name and password prior to receiving a response. If the user name and password are sent in the form variables, this assumes that the script, rather than the server, will perform the user validation.

Expected Response:

SETIResponse: itemcount=#

The symbol # above represents the total number of customers to download.

Details:

Order Manager will look at the value returned from the customer count request. If the value is 0 (zero) then the user is prompted that there are no customers to download. If customers are available, Order Manager will check the maximum download value to see if it needs to perform one or more calls. If more than one is required then the Order Manager will pass additional variables in the request identifying the record start value and the maximum number of customers to send.

3) Downloading Customer Data

This function should require user name and password validation prior to a response and should also test that the connection is secure. For added security, a store “code” can be added to identify which store’s data you’ll need. If user validation or the test for a secure connection fail, an error response should be generated.

If the Order Manager determines that the number of customers to download exceeds the value of the maximum download value, the program will make multiple calls to the script, each time expecting to download the number of customers defined in the download value. To identify which customer records need to be sent in each call, the Order Manager will include two variables called “startnum” and “batchsize”. The variable “startnum” will contain the record number to start with and “batchsize” will contain the total number of customer records to output in this request.

For example: If there are 500 customers to download from the website and the maximum download value is set to 100, the first call from the Order Manager to the website will set the startnum variable to 1 and the batchsize variable to 100. This means that your script should start at the first customer record then output 100 customers. The second call will set the startnum variable to 101 and the batchsize variable to 100. This says start the export at the 101st customer record and export 100 customers. The third call will set the startnum variable to 201 and the batchsize variable to 100. This says start the export at the 201st customer record and export 100 customers. This continues through the fourth and fifth post until all customers are imported.

If the number of customers to download is less than the value of the maximum download value then the Order Manager WILL NOT send the “startnum” and “batchsize” variables in the post. Your script must be able to deal with this by anticipating that the variables may not be present.

Form Variables:

Variable Name	Value	From	Required
setifunction	downloadcustomers	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
startnum	If the number of customers exceeds the maximum allowed for download, this parameter will contain the record number to start with. This will not be present if all customers can be downloaded in a single call.	For Example: if you have 100 customers and allow a maximum of 35 per download then in the first call startnum will be 1, the second call, 36, the third call, 71. Calculated programmatically.	Yes – if there are more customers to download than the MaxOrderDownload parameter specifies
batchsize	The total number of customers to download in one export. May exceed the available number of customers. This will not be present if all customers can be downloaded in a single call.	Value of the “maximum number to download” field (default 100).	Yes – if there are more customers to download than the max download specifies
omversion	Version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

In the examples below, revise the following data points prior to testing:

Replace "www.mystore.com/script.ext" with the appropriate URL pointing to the script on your web server.

Replace "auser" with the proper user name for login on the website.

Replace "pwd" with the proper password for login to the website.

Replace "mystore" with the store code identifier (optional).

It is recommended to include, exclude and change the startnum and batchsize values to verify your script properly adheres to the download cycling principles.

Example URL for testing from browser – uses "GET" method:

<https://www.mystore.com/script.ext?setifunction=downloadcustomers&setiuser=auser&password=pwd&code=mystore&startnum=1&batchsize=100&omversion=5.000>

Example Form Post – uses "POST" method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the "Send" button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="downloadcustomers">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "startnum" VALUE = "1">
  <INPUT TYPE=TEXT NAME = "batchsize" VALUE = "100">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Expected Response:

Script should respond with an XML structure defined in the CustomerImport.xsd specification.

```
<?xml version="1.0"?>
<SETICustomers>
  <Response>
    <ResponseCode>1</ResponseCode>
    <ResponseDescription>success</ResponseDescription>
  </Response>

  ...ONE OR MORE CUSTOMER ELEMENTS IN THIS SECTION IF EXPORT SUCCESSFUL...
  <Customer>
    ...
  </Customer>
</SETICustomers>
```

The ResponseCode element should be set to 1 and the ResponseDescription to "Success" if there are customers available to process. If no customers are present in the file, the merchant will receive an error response. Set the ResponseCode to 2 and the ResponseDescription to "Success" if there are NO customers available to process. Set the ResponseCode to 3 should an error occur during script execution. Set the ResponseDescription to the error message to deliver to the merchant.

Details:

Order Manager will test the incoming Customer data to see if it is legitimate. If so, the data will be written to one or more XML files (.xml) in the DataArchives directory specified in the ArchiveLocation system parameter. Once the internet download is complete, the Order Manager will attempt to import the file(s). If the file(s) is successfully imported, the original text file(s) will be deleted from the Archive Directory. If an error occurs then the file(s) will remain for review.

IMPORTANT NOTE: Although the Customer Import Specification provides for a shipping address for a customer, the data is not currently processed by the Order Manager. If the data is included in the output, the program will disregard it.

XML Customer Import Specification

The Order Manager XML Import Specification defines the structure of the customer data that should be returned from the website. This section details the elements and their various data points.

For specific detail on the XML Specification, see the CustomerImport.xsd schema file included with the Developer's Guide.

The Customer Element is the container element for a customer. This element can appear multiple times within the SETICustomers element.			
Element Name	Description	Restrictions	Required
Customer	Container element for a single Customer		YES if Response Code is set to 1
Customer/ WebID	Customer ID from website	Integer	NO
Customer/ UserName	Customer's User Name on the website	25 characters	NO
Customer/ Password	Customer's password on the website	50 characters	NO
Customer/ AffiliateID	Customer's affiliate identifier	50 characters	NO

The BillAddr Element is the container element for a customer's Billing Name and Address.
This element can appear only once within the Customer element.

Element Name	Description	Restrictions	Required
Customer/ BillAddr	Container element for a customer's Billing name and address		YES
Customer/ BillAddr/ NamePrefix	Customer's Name Prefix (Mr., Mrs. etc.)	50 characters	NO
Customer/ BillAddr/ FirstName	Customer's First Name	50 Characters	YES
Customer/ BillAddr/ MiddleName	Customer's Middle Initial or Name	50 characters	NO
Customer/ BillAddr/ LastName	Customer's Surname/Last Name	50 Characters	YES
Customer/ BillAddr/ NameSuffix	Customer's Name Suffix (Jr., Sr. III, etc.)	50 characters	NO
Customer/ BillAddr/ Company	Customer's Company Name	50 characters	NO
Customer/ BillAddr/ Phone	Customer's Phone Number	50 characters	NO
Customer/ BillAddr/ Fax	Customer's Fax Number	50 characters	NO
Customer/ BillAddr/ Email	Customer's email address	255 characters	NO
Customer/ BillAddr/ TaxID	Customer's tax ID number	20 characters	NO

Customer/ BillAddr/ Address	Container element for the customer's address information		YES
Customer/ BillAddr/Address Addr1	Customer's Street address	255 characters	YES
Customer/ BillAddr/Address Addr2	Street address 2	255 characters	NO
Customer/ BillAddr/Address City	City Name	255 characters	YES
Customer/ BillAddr/Address State	2 letter state code	2 characters	YES
Customer/ BillAddr/Address Zip	Postal code	255 characters	YES
Customer/ BillAddr/Address Country	2 letter Country code	2 characters	YES

The ShipAddr Element is the container element for a customer's Shipping Name and Address.
 This element can appear only once within the Customer element.
CURRENTLY NOT SUPPORT – CAN BE PRESENT IN DOWNLOAD

Element Name	Description	Restrictions	Required
Customer/ ShipAddr	Container element for a customer's Shipping name and address		YES
Customer/ ShipAddr/ NamePrefix	Customer's Name Prefix (Mr., Mrs. etc.)	50 characters	NO
Customer/ ShipAddr/ FirstName	Customer's First Name	50 Characters	YES
Customer/ ShipAddr/ MiddleName	Customer's Middle Initial or Name	50 characters	NO
Customer/ ShipAddr/ LastName	Customer's Surname/Last Name	50 Characters	YES
Customer/ ShipAddr/ NameSuffix	Customer's Name Suffix (Jr., Sr. III, etc.)	50 characters	NO
Customer/ ShipAddr/ Company	Customer's Company Name	50 characters	NO
Customer/ ShipAddr/ Phone	Customer's Phone Number	50 characters	NO
Customer/ ShipAddr/ Email	Customer's email address	255 characters	NO

Customer/ ShipAddr/ Address	Container element for the customer's address information		YES
Customer/ ShipAddr/Address Addr1	Customer's Street address	255 characters	YES
Customer/ ShipAddr/Address Addr2	Street address 2	255 characters	NO
Customer/ ShipAddr/Address City	City Name	255 characters	YES
Customer/ ShipAddr/Address State	2 letter state code	2 characters	YES
Customer/ ShipAddr/Address Zip	Postal code	255 characters	YES
Customer/ ShipAddr/Address Country	2 letter Country code	2 characters	YES

The CustomFields Element is the container element for custom data regarding the customer
This element can appear only once within the Customer element.

Element Name	Description	Restrictions	Required
Customer/ CustomFields	Container element for custom customer data		NO
Customer/ CustomFields/ CustomField	Container element for single custom field. Can appear more than once in CustomFields element		YES if parent element present
Customer/ CustomFields/ CustomField/ FieldName	Name of the custom field	Must match a custom customer field label to have the data point imported	YES if parent element present
Customer/ CustomFields/ CustomField/ FieldValue	Value of the custom field data point	255 characters Custom fields come in 4 data types (text, integer, date, currency). Data contained in this element must match the data type of the field receiving the data to be successfully imported.	YES if parent element present

Example XML response for Customer Download

NOTE: The line indentations and breaks will not be in the real data but is used here for clarity purposes to show parent, child and sibling relationships in the XML.

```

<?xml version='1.0'?>
<SETICustomers>
  <Response>
    <ResponseCode>1</ResponseCode>
    <ResponseDescription>Success</ResponseDescription>
  </Response>
  <Customer>
    <WebID>12548</WebID>
    <UserName>kevin</UserName>
    <Password>xyz123</Password>
    <BillAddr>
      <NamePrefix>Mr.</NamePrefix>
      <FirstName>Kevin</FirstName>
      <LastName>Smith</LastName>
      <Company>Stone Edge Technologies, Inc.</Company>
      <Phone>215-641-1837</Phone>
      <Fax>215-641-1837</Fax>
      <Email>Kevin@stoneedge.com</Email>
      <TaxIDNumber>1234567890</TaxIDNumber>
      <Address>
        <Addr1>One Valley Square</Addr1>
        <Addr2>Suite 130</Addr2>
        <City>Blue Bell</City>
        <State>PA</State>
        <Zip>19422</Zip>
        <Country>US</Country>
      </Address>
    </BillAddr>
    <ShipAddr>
      <NamePrefix>Mr.</NamePrefix>
      <FirstName>Kevin</FirstName>
      <LastName>Smith</LastName>
      <Email>Kevin@stoneedge.com</Email>
      <Company>Stone Edge Technologies, Inc.</Company>
      <Phone>215-641-1837</Phone>
      <Address>
        <Addr1>One Valley Square</Addr1>
        <Addr2>Suite 130</Addr2>
        <City>Blue Bell</City>
        <State>PA</State>
        <Zip>19422</Zip>
        <Country>US</Country>
      </Address>
    </ShipAddr>
    <CustomFields>
      <CustomField>
        <FieldName>Nickname</FieldName>
        <FieldValue>Smitty</FieldValue>
      </CustomField>
    </CustomFields>
  </Customer>
</SETICustomers>

```

Downloading Product & Quantity on Hand Information:

Order Manager can collect product and quantity on hand data from your shopping cart application by accessing the Cart Functions section of the Maintenance Menu. The product import can be used to populate, replace or update your product data in the Order Manager.

By default the Cart Functions option will not be displayed for Generic Cart Users. To allow this tab to be displayed, set the Generic Group System Parameter GenericUseAdvancedFunctions to True. From the Cart Functions screen, you can download your product data directly into Order Manager's Inventory table. The download is a three step process:

1. Order Manager requests the script's version number from the website. This is done for general testing of communications between Order Manager and the script file and also to ensure that if structural changes are made to Order Manager to support a new script that you can test to see if the latest script is installed before running the download.
2. Order Manager then requests the total number of products that need to be downloaded from the site. Typically a download of 100 products or less causes no problems, however, if there is a large number to retrieve or you have products that contain a large number of options, you may run into timeout issues on the server as the script executes. There is a field on the Cart Functions screen labeled "Maximum number of products or customer to download at one time" that allows you to set the maximum number per download. If the total number of products to download exceeds the max download value, the Order Manager will call for several downloads, retrieving the maximum number of products in each call until all products are received.
3. Once Order Manager determines how many times it will execute the download, it begins the download process. The data received by Order Manager is then fed into one or more XML files that are stored in the DataArchives directory (identified in the system parameter ArchiveLocation). Once the download(s) is complete, Order Manager will test and import the product file(s). Both the Product Download and Quantity on Hand download perform the same for step 1 and step 2, so these steps are documented once below, however, the XML output of the script in response to the two functions will be different. The Product download and Quantity on Hand download XML responses are defined separately below.

1) Requesting Script Version Data: This function should **not** require user name and password validation to receive the response and should also be allowed to run if the connection is not secure (should respond to both http and https requests).. This enables easy testing by Stone Edge Technologies, Inc. when troubleshooting web communications.

Form Variables:

Variable Name	Value	From	Required
setifunction	sendversion	Hard coded into visual basic, cannot be modified	YES
omversion	The version number of the Order Manager being used	Value of the Version field in the last record in the Version table – Used for version control	NO

For the examples below, Replace www.mystore.com/script.ext with the appropriate URL pointing to the script on your web server

Example URL for testing from browser – uses "GET" method:

After making the revisions detailed above, copy the entire line and paste it into the address bar of your browser.

<https://www.mystore.com/script.ext?setifunction=sendversion&omversion=5.000>

Example Form Post – uses the “POST” method:

Save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the “Send” button to send the request.

```
<FORM ACTION="https://www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE ="sendversion">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE ="5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Expected Response:

SETIResponse: version=#

The # symbol above represents the version of the script file on the website in ##### or #.### format. The version number must be 4 numbers plus decimal. For example SETIResponse: version=1.000

Details:

Order Manager will test the returned value against the GenericScriptVersionNumber system parameter. If the returned value is less than the parameter value, Order Manager will notify the user that the script version is too low and will abort the order import. This provides a method of version control for new features or bug fixes.

2) Determining the number of products to download

This function should require user name and password validation prior to a response and should also test that the connection is secure. For added security, a store “code” can be added to identify which store’s data you’ll need. If user validation or the test for a secure connection fail, an error response should be generated

Order Manager will post a request for the total number of products available for download. The total count of product records should be returned as an integer value.

Form Variables:

Variable Name	Value	From	Required
setifunction	getproductscount	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	Optional: A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
omversion	The version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

In the examples below revise the following data points prior to testing:

Replace “www.mystore.com/script.ext” with the appropriate URL pointing to the script on your web server.

Replace “auser” with the proper user name for login on the website.

Replace “pwd” with the proper password for login to the website.

Replace “mystore” with the store code identifier (optional).

Example URL for testing from browser – uses “GET” method:

<https://www.mystore.com/script.ext?setifunction=getproductscount&setiuser=auser&password=pwd&code=mystore&omversion=5.000>

Example Form Post – uses “POST” method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the “Send” button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="getproductscount">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Note: If you have your web server set up to validate the request, you may need to enter your user name and password prior to receiving a response. If the user name and password are sent in the form variables, this assumes that the script, rather than the server, will perform the user validation.

Expected Response:

SetiResponse: itemcount=#

The symbol # above represents the total number of products to download. This should be an integer value only.

Details:

Order Manager will look at the value returned from the product count request. If the value is 0 (zero) then the user is prompted that there are no products to download. If products are available, Order Manager will check the maximum download value to see if it needs to perform one or more downloads. If more than one is required then the Order Manager will pass additional variables in the request identifying the record start value and the maximum number of products to send.

3) Downloading Product Data

This function should require user name and password validation prior to a response and should also test that the connection is secure. For added security, a store “code” can be added to identify which store’s data you’ll need. If user validation or the test for a secure connection fail, an error response should be generated.

If the Order Manager determines that the number of products to download exceeds the value of the maximum download value, the program will make multiple calls to the script, each time expecting to download the number of products defined in the download value. To identify which product records need to be sent in each call, the Order Manager will include two variables called “startnum” and “batchsize”. The variable “startnum” will contain the product record number to start with and “batchsize” will contain the total number of product records to output in this request.

For example: If there are 500 products to download from the website and the maximum download value is set to 100, the first call from the Order Manager to the website will set the startnum variable to 1 and the batchsize variable to 100. This means that your script should start at the first product record then output 100 products. The second call will set the startnum variable to 101 and the batchsize variable to 100. This says start the export at the 101st product record and export 100 products. The third call will set the startnum variable to 201 and the batchsize variable to 100. This says start the export at the 201st product record and export 100 products. This continues through the fourth and fifth post until all products are imported.

If the number of products to download is less than the value of the maximum download value then the Order Manager WILL NOT send the “startnum” and “batchsize” variables in the post. Your script must be able to deal with this by anticipating that the variables may not be present.

Form Variables:

Variable Name	Value	From	Required
setifunction	downloadprods	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
startnum	If the number of products exceeds the maximum allowed for download, this parameter will contain the record number to start with. This will not be present if all products can be downloaded in a single call.	For Example: if you have 100 products and allow a maximum of 35 per download then in the first call startnum will be 1, the second call, 36, the third call, 71. Calculated programmatically.	Yes – if there are more products to download than the MaxOrderDownload parameter specifies
batchsize	The total number of products to download in one export. May exceed the available number of orders. This will not be present if all products can be downloaded in a single call.	Value of the “maximum number to download” field (default 100).	Yes – if there are more products to download than the max download specifies
omversion	Version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

In the examples below, revise the following data points prior to testing:

Replace "www.mystore.com/script.ext" with the appropriate URL pointing to the script on your web server.

Replace "auser" with the proper user name for login on the website.

Replace "pwd" with the proper password for login to the website.

Replace "mystore" with the store code identifier (optional).

It is recommended to include, exclude and change the startnum and batchsize values to verify your script properly adheres to the download cycling principles.

Example URL for testing from browser – uses "GET" method:

https://www.mystore.com/script.ext?setifunction=downloadprods&setiuser=**auser**&password=**pwd**&code=**mystore**&startnum=**1**&batchsize=**100**&omversion=5.000

Example Form Post – uses "POST" method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the "Send" button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="downloadprods">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "startnum" VALUE = "1">
  <INPUT TYPE=TEXT NAME = "batchsize" VALUE = "100">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Expected Response:

```
<?xml version="1.0"?>
<SETIPProducts>
  <Response>
    <ResponseCode>1</ResponseCode>
    <ResponseDescription>success</ResponseDescription>
  </Response>

  ...ONE OR MORE PRODUCT ELEMENTS IN THIS SECTION IF EXPORT SUCCESSFUL...
  <Product>
    ...
  </Product>
</SETIPProducts>
```

The ResponseCode element should be set to 1 and the ResponseDescription to "Success" if there are products available to process. If no products are present in the file, the merchant will receive an error response. Set the ResponseCode to 2 and the ResponseDescription to "Success" if there are NO products available to process. Set the ResponseCode to 3 should an error occur during script execution. Set the ResponseDescription to the error message to deliver to the merchant.

Details:

Order Manager will test the incoming Product data to see if it is legitimate. If so, the data will be written to one or more XML files (.xml) in the DataArchives directory specified in the ArchiveLocation system parameter. Once the internet download is complete, the Order Manager will attempt to import the file(s). If the file(s) is successfully imported, the original text file(s) will be deleted from the Archive Directory. If an error occurs then the file(s) will remain for review.

XML Product Import Specification

The Order Manager XML Import Specification defines the structure of the product data that should be returned from the website. This section details the elements and their various data points.

For specific detail on the XML Specification, see the ProductImport.xsd schema file included with the Developer's Guide.

The Product Element is the container element for a product. This element can appear multiple times within the SETIProducts element.			
Element Name	Description	Restrictions	Required
Product	Container element for a single Product		YES if Response Code is set to 1
Product/Code	Product SKU	255 characters	YES
Product/WebID	Product's ID number from website	50 characters	NO
Product/Name	Name for the item – typically the short description	200 characters	YES
Product/Price	The Product's Unit Price. This may or may not include price modifiers from product options. The OptionPriceInPrice parameter (Order Options group) defines whether the option price is included in this value	Decimal value No currency symbol	YES
Product/Cost	NOT YET SUPPORTED The Product's Cost Per Unit	Decimal value No currency symbol	NO
Product/Description	The long description for the product		NO
Product/Weight	The Unit Weight of the product	Decimal value No weight measurement symbol	YES
Product/Thumb	Relative or Absolute URL to thumbnail image for product	250 characters	NO
Product/Image	Relative or Absolute URL to image for product	250 characters	NO
Product/Taxable	Taxable flag for product	Limit to "Yes" or "No"	NO Yes is assumed
Product/Discontinued	Flag noting whether product is inactive	Limit to "Yes" or "No"	NO No is assumed
Product/QOH	Quantity on hand for the product	Integer value	NO

The OptionLists Element is the container element for all attributes and options pertaining to a product. This element can appear once within the Product element.			
Element Name	Description	Restrictions	Required
Product/ OptionLists	Container element for all Product Options		NO
Product/ OptionLists/ ProductOption	Container Element for a single Attribute and its selectable options		YES
Product/ OptionLists/ ProductOption/ WebID	The website's ID number for the attribute	Integer value	NO
Product/ OptionLists/ ProductOption/ TemplateID	The Website's ID Number for the Attribute Template the attribute belongs to	Integer Value	NO
Product/ OptionLists/ ProductOption/ Name	The Name of the attribute like Size, Color or Style	50 characters	YES
Product/ OptionLists/ ProductOption/ Prompt	The text seen by the customer for this attribute	50 characters	NO
Product/ OptionLists/ ProductOption/ Type	The type of control displayed to the customer for this attribute – this can impact how SubSKUs (product variants) are built in the Order Manager.	Limit to: "select", "radio", "text", "memo", "checkbox"	NO
Product/ OptionLists/ ProductOption/ Price	Price Modifier of the attribute per unit. This value is only recognized for attributes that do not have selectable options such a text, memo or checkbox type options	Decimal value No currency symbol	NO
Product/ OptionLists/ ProductOption/ Cost	Cost Modifier of the attribute per unit. This value is only recognized for attributes that do not have selectable options such a text, memo or checkbox type options	Decimal value No currency symbol	NO
Product/ OptionLists/ ProductOption/ Image	Relative or Absolute path to Image file relating to this attribute	250 characters	NO

Product/ OptionLists/ ProductOption/ SortOrder	Numeric value identifying the sort order of this attribute across all attributes for this product. – this can impact how SubSKUs (product variants) are built in the Order Manager.	Integer value	NO
Product/ OptionLists/ ProductOption/ DefaultID	The WebID number of the option that will appear as the default for this attribute	Integer Value	NO
Product/ OptionLists/ ProductOption/ Required	Flag indicating whether a selection from this attribute is required by the customer	Limit to “Yes” or “No”	NO
Product/ OptionLists/ ProductOption/ UseForBuild	Flag indicating whether this attribute should be used for building SubSKUs	Limit to “Yes” or “No”	NO

The OptionValue Element is the container element for all options pertaining to this attribute.
This element can appear multiple times within the ProductOption element.

Element Name	Description	Restrictions	Required
Product/ OptionLists/ ProductOption/ OptionValue	Container element for a selectable option for the given attribute		NO
Product/ OptionLists/ ProductOption/ OptionValue/ WebID	The unique ID Number of the option from the website	Integer	NO
Product/ OptionLists/ ProductOption/ OptionValue/ Name	The name or value of the option. For example if the current attribute is Size, this would be Small or Medium or Large	250 characters	YES
Product/ OptionLists/ ProductOption/ OptionValue/ Code	Value to add to the parent SKU for the creation of a variant SKU (SubSKU) The value reported in this element can be reported in the Name element instead	50 characters	NO
Product/ OptionLists/ ProductOption/ OptionValue/ Prompt	Text that would appear for the customer for this option	50 characters	NO

Product/ OptionLists/ ProductOption/ OptionValue/ Price	Price Modifier of the option, per unit.	Decimal value No currency symbol	NO
Product/ OptionLists/ ProductOption/ OptionValue/ Cost	Cost modifier of the option, per unit	Decimal Value No currency symbol	NO
Product/ OptionLists/ ProductOption/ OptionValue/ Weight	Weight modifier of the option, per unit	Decimal value No weight measurement symbol	NO
Product/ OptionLists/ ProductOption/ OptionValue/ Image	Relative or absolute path to image file pertaining to this selectable option	250 characters	NO
Product/ OptionLists/ ProductOption/ OptionValue/ SortOrder	Unique number identifying the sort order of this item across all of the options for this attribute	Integer	NO

<p>The CustomFields Element is the container element for custom data regarding the Product This element can appear only once within the Product element.</p>			
Element Name	Description	Restrictions	Required
Product/ CustomFields	Container element for custom Product data		NO
Product/ CustomFields/ CustomField	Container element for single custom field. Can appear more than once in CustomFields element		YES if parent element present
Product/ CustomFields/ CustomField/ FieldName	Name of the custom field	Must match a custom Product field label to have the data point imported	YES if parent element present
Product/ CustomFields/ CustomField/ FieldValue	Value of the custom field data point	255 characters Custom fields come in 4 data types (text, integer, date, currency). Data contained in this element must match the data type of the field receiving the data to be successfully imported.	YES if parent element present

EXAMPLE XML Response to Product Download

NOTE: The line indentations and breaks will not be in the real data but is used here for clarity purposes to show parent, child and sibling relationships in the XML.

```

<?xml version='1.0'?>
<SETIProducts>
  <Response>
    <ResponseCode>1</ResponseCode>
    <ResponseDescription>Success</ResponseDescription>
  </Response>
  <Product>
    <Code>ABC</Code>
    <WebID>5</WebID>
    <Name>Letters</Name>
    <Price>1.00</Price>
    <Cost>1.00</Cost>
    <Description></Description>
    <Weight>0.00</Weight>
    <Thumb></Thumb>
    <Image></Image>
    <Taxable>No</Taxable>
    <Discontinued>No</Discontinued>
    <QOH>12</QOH>
    <CustomFields>
      <CustomField>
        <FieldName>Condition</FieldName>
        <FieldValue>Used</FieldValue>
      </CustomField>
    </CustomFields>
  </Product>
  <Product>
    <Code>BOX</Code>
    <WebID>7</WebID>
    <Name>Colorful Toy Box</Name>
    <Price>100.00</Price>
    <Cost>25.00</Cost>
    <Description>
      This is a colorful toy box. Select the colors for the top, bottom and sides...
    </Description>
    <Weight>10.00</Weight>
    <Thumb></Thumb>
    <Image></Image>
    <Taxable>Yes</Taxable>
    <Discontinued>No</Discontinued>
    <QOH>NA</QOH>
    <OptionLists>
      <ProductOption>
        <WebID>11</WebID>
        <TemplateID>0</TemplateID>
        <Name>Wood</Name>
        <Prompt>Select the type of wood</Prompt>
        <Type>select</Type>
        <Price>0.00</Price>
        <Cost>0.00</Cost>
        <Image></Image>
        <SortOrder>11</SortOrder>
      </ProductOption>
    </OptionLists>
  </Product>
</SETIProducts>

```

```

<DefaultID>31</DefaultID>
<Required>No</Required>
<OptionValue>
  <WebID>31</WebID>
  <Name>Pine</Name>
  <Prompt>Pine</Prompt>
  <Price>0.00</Price>
  <Cost>0.00</Cost>
  <Image></Image>
  <SortOrder>31</SortOrder>
</OptionValue>
<OptionValue>
  <WebID>32</WebID>
  <Name>Maple</Name>
  <Prompt>Maple</Prompt>
  <Price>0.00</Price>
  <Cost>0.00</Cost>
  <Image></Image>
  <SortOrder>32</SortOrder>
</OptionValue>
<OptionValue>
  <WebID>33</WebID>
  <Name>Oak</Name>
  <Prompt>Oak</Prompt>
  <Price>10.00</Price>
  <Cost>0.00</Cost>
  <Image></Image>
  <SortOrder>33</SortOrder>
</OptionValue>
</ProductOption>
<ProductOption>
  <WebID>2</WebID>
  <TemplateID>2</TemplateID>
  <Name>Top</Name>
  <Prompt>Select a Color for the Product Top</Prompt>
  <Type>select</Type>
  <Price>0.00</Price>
  <Cost>0.00</Cost>
  <Image></Image>
  <SortOrder>2</SortOrder>
  <DefaultID>0</DefaultID>
  <Required>No</Required>
  <OptionValue>
    <WebID>7</WebID>
    <Name>Blue</Name>
    <Prompt>Blue</Prompt>
    <Price>0.00</Price>
    <Cost>0.00</Cost>
    <Image></Image>
    <SortOrder>7</SortOrder>
  </OptionValue>
  <OptionValue>
    <WebID>6</WebID>
    <Name>Green</Name>
    <Prompt>Green</Prompt>
    <Price>0.00</Price>
    <Cost>0.00</Cost>

```

```
    <Image></Image>
    <SortOrder>6</SortOrder>
  </OptionValue>
  <OptionValue>
    <WebID>5</WebID>
    <Name>Red</Name>
    <Prompt>Red</Prompt>
    <Price>0.00</Price>
    <Cost>0.00</Cost>
    <Image></Image>
    <SortOrder>5</SortOrder>
  </OptionValue>
</ProductOption>
</OptionLists>
<CustomFields>
  <CustomField>
    <FieldName>Condition</FieldName>
    <FieldValue>New</FieldValue>
  </CustomField>
</CustomFields>
</Product>
</SETIProducts>
```

3) Downloading Quantity on Hand (Inventory) Data

This function, activated from the Get QOH from Website button on the Advanced Cart Functions screen, is used to return the product SKU (Code) and its current quantity on hand (QOH) from the website. This can be used to synchronize the inventory levels in the Order Manager with those on the website. This function makes the same three calls as the product download. The only differences are the value of the setifunction variable and the amount of data returned.

If the Order Manager determines that the number of products to download exceeds the value of the maximum download value, the program will make multiple calls to the script, each time expecting to download the number of products defined in the download value. To identify which product records need to be sent in each call, the Order Manager will include two variables called "startnum" and "batchsize". The variable "startnum" will contain the product record number to start with and "batchsize" will contain the total number of product records to output in this request.

For example: If there are 500 products to download from the website and the maximum download value is set to 100, the first call from the Order Manager to the website will set the startnum variable to 1 and the batchsize variable to 100. This means that your script should start at the first product record then download 100 products. The second call will set the startnum variable to 101 and the batchsize variable to 100. This says start the export at the 101st product record and export 100 products. The third call will set the startnum variable to 201 and the batchsize variable to 100. This says start the export at the 201st product record and export 100 products. This continues through the fourth and fifth post until all products are imported.

If the number of products to download is less than the value of the maximum download value then the Order Manager WILL NOT send the "startnum" and "batchsize" variables in the post. Your script must be able to deal with this by anticipating that the variables may not be present.

Form Variables:

Variable Name	Value	From	Required
setifunction	downloadqoh	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
startnum	If the number of products exceeds the maximum allowed for download, this parameter will contain the record number to start with. This will not be present if all products can be downloaded in a single call.	For Example: if you have 100 products and allow a maximum of 35 per download then in the first call startnum will be 1, the second call, 36, the third call, 71. Calculated programmatically.	Yes – if there are more products to download than the MaxOrderDownload parameter specifies
batchsize	The total number of products to download in one export. May exceed the available number of orders. This will not be present if all products can be downloaded in a single call.	Value of the "maximum number to download" field (default 100).	Yes – if there are more products to download than the max download specifies
omversion	Version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

In the examples below, revise the following data points prior to testing:

Replace "www.mystore.com/script.ext" with the appropriate URL pointing to the script on your web server.

Replace "auser" with the proper user name for login on the website.

Replace "pwd" with the proper password for login to the website.

Replace "mystore" with the store code identifier (optional).

It is recommended to include, exclude and change the startnum and batchsize values to verify your script properly adheres to the download cycling principles.

Example URL for testing from browser – uses "GET" method:

<https://www.mystore.com/script.ext?setifunction=downloadqoh&setiuser=auser&password=pwd&code=mystore&startnum=1&batchsize=100&omversion=5.000>

Example Form Post – uses "POST" method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the "Send" button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="downloadqoh">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "startnum" VALUE = "1">
  <INPUT TYPE=TEXT NAME = "batchsize" VALUE = "100">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Expected Response:

```
<?xml version="1.0"?>
<SETIProducts>
  <Response>
    <ResponseCode>1</ResponseCode>
    <ResponseDescription>success</ResponseDescription>
  </Response>

  ...ONE OR MORE PRODUCT ELEMENTS IN THIS SECTION IF EXPORT SUCCESSFUL...
  <Product>
    ...
  </Product>
</SETIProducts>
```

The ResponseCode element should be set to 1 and the ResponseDescription to "Success" if there are products available to process. If no products are present in the file, the merchant will receive an error response. Set the ResponseCode to 2 and the ResponseDescription to "Success" if there are NO products available to process. Set the ResponseCode to 3 should an error occur during script execution. Set the ResponseDescription to the error message to deliver to the merchant.

Details:

Order Manager will test the incoming Product data to see if it is legitimate. If so, the data will be written to one or more XML files (.xml) in the DataArchives directory specified in the ArchiveLocation system parameter. Once the internet download is complete, the Order Manager will attempt to import the file(s). If the file(s) is successfully imported, the original text file(s) will be deleted from the Archive Directory. If an error occurs then the file(s) will remain for review.

XML QOH Import Specification

The Order Manager XML Import Specification defines the structure of the product data that should be returned from the website. This section details the elements and their various data points.

For specific detail on the XML Specification, see the QOHImport.xsd schema file included with the Developer's Guide.

The Product Element is the container element for a product. This element can appear multiple times within the SETIPProducts element.			
Element Name	Description	Restrictions	Required
Product	Container element for a single Product		YES if Response Code is set to 1
Product/Code	Product SKU	255 characters	YES
Product/WebID	Product ID number from website	50 characters	NO
Product/QOH	Product's quantity on hand from website	Integer Value	YES

Example XML Response to Inventory Download

NOTE: The line indentations and breaks will not be in the real data but is used here for clarity purposes to show parent, child and sibling relationships in the XML.

```
<?xml version='1.0'?>
<SETIPProducts>
  <Response>
    <ResponseCode>1</ResponseCode>
    <ResponseDescription>Success</ResponseDescription>
  </Response>
  <Product>
    <Code>ABC</Code>
    <WebID>120</WebID>
    <QOH>10</QOH>
  </Product>
  <Product>
    <Code>BOX</Code>
    <WebID>121</WebID>
    <QOH>27</QOH>
  </Product>
  <Product>
    <Code>Kit</Code>
    <WebID>122</WebID>
    <QOH>15</QOH>
  </Product>
  <Product>
    <Code>SEOM</Code>
    <WebID>123</WebID>
    <QOH>75</QOH>
  </Product>
</SETIPProducts>
```

Inventory Synchronization

Uploading Bulk Inventory Data

Order Manager can send SKU and quantity on hand (QOH) data up to your website so you can revise the store's QOH to correspond to the Order Manager's reported QOH. This is run from the Send QOH to Website option on the Advanced Cart Options form.

The Order Manager can generate a delimited string or an XML structure containing SKU and QOH data. The data format sent is dependant on the setting of the GenericQOHChangeSendAsXML system parameter (Generic group). If this parameter is True, the "update" field will contain an XML based data structure defined by the QOHExport.xsd. If this parameter is False (default) or the version of the Order Manager is less than 5.500, the "update" variable will contain delimited data.

The number of products uploaded per send is based on the Maximum Number of Items To Send field (default = 50). This number can be modified to affect the speed at which the Order Manager operates. Typically a higher number per send can run faster in the Order Manager, however your script may "time-out" if fed too much data in a single post. Many factors determine the script's speed from server operating systems, script efficiency, upload bandwidth, etc. so play with this value to see what will give you the best performance for your site.

Form Variables:

Variable Name	Value	From	Required
setifunction	qohreplace	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
count	number of items sent to be updated	calculated at time of send	NO
update	delimited string or XML structure containing SKU and QOH data	product data gathered at the time of send. Format of data depends on GenericQOHChangeSendAsXML parameter	YES
omversion	Version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

In the examples below, revise the following data points prior to testing:

Replace "www.mystore.com/script.ext:" with the appropriate URL pointing to the script on your web server.

Replace "auser" with the proper user name for login on the website.

Replace "pwd" with the proper password for login to the website.

Replace "mystore" with the store code identifier (optional).

If the GenericQOHChangeSendAsXML Parameter is False:

The "update" variable contains a string built using the product SKU, then a tilde "~", the replacement quantity on hand value then a pipe character "|". This is repeated for each SKU/QOH combination. Last entry in the delimited string will not have a pipe character at the end. Order Manager does not test the product SKU for the presence of the delimiter characters (~, |), so you'll want to ensure your script checks for more than one tilde per product string and use the last tilde as the parse point. The "count" variable will give you a total number of SKU/QOH pairs in the "update" variable.

Example URL for testing from browser – uses "GET" method:

<https://www.mystore.com/script.ext?setifunction=qohreplace&setiuser=auser&password=pwd&code=mystore&count=5&update=sku1~10|sku2~8|sku3~15|sku4~32|sku5~18&omversion=5.000>

Example Form Post – uses "POST" method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the "Send" button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="qohreplace">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "count" VALUE = "5">
  <INPUT TYPE=TEXT NAME = "update" VALUE = "sku1~10|sku2~8|sku3~15|sku4~32|sku5~18">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

If the GenericQOHChangeSendAsXML Parameter is True:

The “update” variable will contain the following XML structure as defined in the QOHExport.xsd schema file.

QOH Export XML Structure contained in the “update” form variable.			
Element Name	Description	Restrictions	Required
SETIProducts	Container Element for one or more products		YES
SETIProducts/ Product	Container element for a single Product		YES
SETIProducts/ Product/SKU	Product SKU	255 characters	YES
SETIProducts/ Product/QOH	Product’s quantity on hand from website	Positive Integer Value	YES

Example XML:

NOTE: The line indentations and breaks will not be in the real data but is used here for clarity purposes to show parent, child and sibling relationships in the XML.

```

<SETIProducts>
  <Product>
    <SKU>SKU1</SKU>
    <QOH>10</QOH>
  </Product>
  <Product>
    <SKU>SKU2</SKU>
    <QOH>8</QOH>
  </Product>
  <Product>
    <SKU>SKU3</SKU>
    <QOH>15</QOH>
  </Product>
  <Product>
    <SKU>SKU4</SKU>
    <QOH>32</QOH>
  </Product>
  <Product>
    <SKU>SKU5</SKU>
    <QOH>18</QOH>
  </Product>
</SETIProducts>

```

Example URL for testing from browser – uses “GET” method:

```
https://www.mystore.com/script.ext?setifunction=qohreplace&setiuser=ouser&password=pwd&code=mystore
&update=<SetiProducts><Product><SKU>SKU1</SKU><QOH>10</QOH>
</Product><Product><SKU>SKU2</SKU><QOH>8</QOH></Product><Product><SKU>SKU3</SKU><QOH>15</Q
OH></Product><Product><SKU>SKU4</SKU><QOH>32</QOH></Product><Product><SKU>SKU5</SKU><QOH>1
8</QOH></Product></SETIProducts>&omversion=5.000
```

Example Form Post – uses “POST” method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the “Send” button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="qohreplace">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "ouser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=TEXT NAME = "update" VALUE = "<SetiProducts><Product>
  <SKU>SKU1</SKU><QOH>10</QOH></Product><Product><SKU>SKU2</SKU><QOH>8</QOH></Product
  ><Product><SKU>SKU3</SKU><QOH>15</QOH></Product><Product><SKU>SKU4</SKU><QOH>32</QOH></
  Product><Product><SKU>SKU5</SKU><QOH>18</QOH></Product></SETIProducts">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Expected Response:

A text based output with header of “SETIResponse”, then an individual record of each SKU’s result (SKU=OK, SKU=NA or SKU=NF), finishing with a “SETIEndOfData” record. Results can be “OK” for correctly updated items, “NA” if the item’s inventory numbers are not being tracked or “NF” if the item’s SKU was not found in the website’s inventory. The Order Manager does not process this result other than to look for the SETIEndOfData marker to ensure the update was completed. The system will write the response to a text file in the DataArchives directory (specified in the ArchiveLocation system parameter) so the user can review the results of the inventory update.

EXAMPLE OUTPUT:

```
SETIResponse
SKU1=OK
SKU2=NA
SKU3=OK
SKU4=OK
SKU5=NF
SETIEndOfData
```

Uploading Individual Inventory Data

As changes are made to the quantity on hand (QOH) levels of your inventory in the Order Manager, the program can notify all of your shopping carts of the change (provided the cart is capable and has been set up to synchronize QOH). The Order Manager can send the product SKU and amount of change to the QOH up to your website to revise the store's QOH. This feature runs automatically in the background as you sell, adjust or receive inventory in the Order Manager. Should you have several shopping carts set up in the program, an import of orders from one cart can signal the change in QOH to all the other carts.

This feature is called Synchronize QOH and can be turned on or off by using the Program group system parameter "SynchQOH". Once this parameter is set to True a new checkbox called "Synchronize Inventory" will be available in the Shopping Carts setup form. You will need to edit the shopping cart setup to check this option. The Order Manager can synchronize with multiple stores, so make sure you edit all stores that will be part of the inventory synchronization.

As changes are made to internal inventory levels, the Order Manager will assemble the data to post to one or more websites in a table called the Post Stack. One computer will act as the "Stack Controller" and check the stack every 20 seconds to see if there is data to send to the website. If data exists and an internet connection exists, the data is transmitted to the website. To establish the location of the Stack Controller, look in the Program group of the system parameters for WebStackLocation. If you run Order Manager on one computer, set this to "Local". If running on a network, set this to "Shared". If you have set this to shared then you will also need to establish which computer will control the communications with the website, look for the WebStackController parameter. Enter the network name of the computer on your network that will trigger the sending of the data. The computer that acts as the Stack Controller must have Order Manager running for the data to be sent. Should the internet connection be down or unavailable, the Order Manager can leave the data in the Post Stack table until a connection becomes available. The amount of time the data stays in the Post Stack is determined by the WebStackRetentionDays system parameter. If this is set to zero (default) the data will remain there indefinitely.

Should a server error be returned, the Order Manager will write the error response into the Post Stack table, will increment the number of attempts made for the post and will increase the timer on the Stack Controller so that Order Manager waits longer between posts. The timer can be advanced up to a total of 20 minutes between sends. Posts that receive server errors will remain on the Post Stack based on the value of the WebStackRetentionDays parameter.

Should a script error or negative result be returned by the script at the time a post is made, the Order Manager will write the error response into the Post Stack table, will increment the number of attempts made for the post and will attempt to send the data again up to a total of 11 attempts. If the script registers a failure or negative result at the 11th send, the Order Manager will remove the record from the Post Stack and record an error in the ErrorLog table.

The Order Manager can generate a delimited string (SKU~QOH) or an XML structure containing SKU and QOH data. The data format sent is dependant on the setting of the GenericQOHChangeSendAsXML system parameter (Generic group). If this parameter is True, the "update" field will contain an XML based data structure defined by the QOHExport.xsd. If this parameter is False (default) or the version of the Order Manager in less than 5.500, the "update" variable will contain delimited data.

Form Variables:

Variable Name	Value	From	Required
setifunction	invupdate	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
update	delimited string or XML structure containing SKU and QOH data	product data gathered at the time of send. Format of data depends on GenericQOHChangeSendAsXML parameter	YES
omversion	Version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

In the examples below, revise the following data points prior to testing:

Replace “www.mystore.com/script.ext: with the appropriate URL pointing to the script on your web server.

Replace “auser” with the proper user name for login on the website.

Replace “pwd” with the proper password for login to the website.

Replace “mystore” with the store code identifier (optional).

If the GenericQOHChangeSendAsXML Parameter is False:

The “update” variable contains a string built using the product SKU, then a tilde “~”, then amount to increase or decrease the quantity on hand value. Order Manager does not test the product SKU for this delimiter, so you’ll want to ensure your script checks for more than one tilde per product string and use the last tilde as the parse point.

Example URL for testing from browser – uses “GET” method:

Increasing QOH by 10 units:

`https://www.mystore.com/script.ext?setifunction=invupdate&setiuser=auser&password=pwd&code=mystore&update=sku1~10&omversion=5.000`

Decreasing QOH by 100 units:

`https://www.mystore.com/script.ext?setifunction=invupdate&setiuser=auser&password=pwd&code=mystore&update=sku1~-100&omversion=5.000`

Example Form Post – uses “POST” method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the “Send” button to send the request.

Increasing QOH by 10 units:

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="invupdate">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "update" VALUE = "sku1~10">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Decreasing QOH by 100 units:

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="invupdate">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "update" VALUE = "sku1~-100">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

If the GenericQOHChangeSendAsXML Parameter is True:

The “update” variable will contain the following XML structure as defined in the QOHExport.xsd schema file.

QOH Export XML Structure contained in the “update” form variable.			
Element Name	Description	Restrictions	Required
SETIProducts	Container Element for one or more products		YES
SETIProducts/ Product	Container element for a single Product		YES
SETIProducts/ Product/SKU	Product SKU	255 characters	YES
SETIProducts/ Product/QOH	Product’s quantity on hand from website	Signed Integer Value	YES

Example XML:

Increase QOH by 10 units:

```
<SetiProducts>
  <Product>
    <SKU>SKU1</SKU>
    <QOH>10</QOH>
  </Product>
</SETIProducts>
```

Decrease QOH by 100 units:

```
<SetiProducts>
  <Product>
    <SKU>SKU1</SKU>
    <QOH>-100</QOH>
  </Product>
</SETIProducts>
```

Example URL for testing from browser – uses “GET” method:

Increase QOH by 10 units:

```
https://www.mystore.com/script.ext?setifunction=invupdate&setiuser=auser&password=pwd&code=mystore
&update=<SetiProducts><Product><SKU>SKU1</SKU><QOH>10</QOH></Product></SETIProducts>
&omversion=5.000
```

Decrease QOH by 100 units:

```
https://www.mystore.com/script.ext?setifunction=invupdate&setiuser=auser&password=pwd&code=mystore
&update=<SetiProducts><Product><SKU>SKU1</SKU><QOH>-100</QOH></Product></SETIProducts>
&omversion=5.000
```

Example Form Post – uses “POST” method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the “Send” button to send the request.

Increase QOH by 10 units:

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="invupdate">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=TEXT NAME = "update" VALUE = "<SetiProducts><Product><SKU>SKU1</SKU>
  <QOH>10</QOH></Product></SETIProducts">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Decrease QOH by 100 units:

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="invupdate">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=TEXT NAME = "update" VALUE = "<SetiProducts><Product><SKU>SKU1</SKU>
  <QOH>-100</QOH></Product></SETIProducts">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Expected Response:

Response is text based and includes the following semicolon delimited name/value pairs of information:

SETIRESPONSE – value should be “OK” for a proper update to the website’s QOH or “False” if an update did not occur

SKU – show’s the product SKU being updated

QOH – shows the new QOH on the website

NOTE – if the response is false, this should contain one of the following:

“NotTracking” – means that inventory tracking for this SKU is turned off at the website

“NotFound” – SKU was not found on the website

“Error” – Error when attempting the QOH update (if not returned through a SETIError structure)

Good response (item updated):

SETIRESPONSE=OK;SKU=sku1;QOH=20;NOTE=

Bad Response (item not updated):

SETIRESPONSE=False;SKU=sku1;QOH=NA;NOTE=NotTracking

Bad Response (item not found):

SETIRESPONSE=False;SKU=sku1;QOH=NF;NOTE=NotFound

Details:

If the response is OK then the record in the Post Stack will be deleted. If the response is False then the Order Manager will attempt to send the item again for a total of 11 attempts. If the item continues to respond as false, the record is deleted and an entry is made in the ErrorLog table. Should the script not respond (server error or internet error), the record will remain on the stack until the error condition is remedied or the user deletes the records from the Stack Viewer.

Order Status Data

Uploading Order Status information

The Order Manager can generate name/value pairs or an XML structure containing information pertaining to an order's current status and any tracking data associated with the order. The XML structure can also include individual line item details in the status notification. To use this system with your shopping cart you must set the UseGenericStatusUpdates parameter (Program group) to True.

The data format sent is dependant on the setting of the GenericStatusSendXML system parameter (Generic group). If this parameter is True, an "update" variable will contain an XML based data structure defined by the StatusExport.xsd. If this parameter is False (default) or the version of the Order Manager is less than 5.500, the program will post name/value pair data.

If the GenericStatusSendXML parameter is False:

Form Variables for Name/Value Pair post:

Variable Name	Value	From	Required
setifunction	updatestatus	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
ordernumber	The order number from the website	SourceOrderNumber or SourceOrderID field in the Orders table	YES
orderstatus	Current status label applied to the order	OrderStatus field in the Orders table	YES
refnumber	The Order Manager's Order Number	OrderNumber field in the Orders table	NO
orderdetail	Additional notes pertaining to the order status	OrdStatDetail field in the Orders table	NO
tracknum	Tracking Number associated with an order SEE NOTE BELOW!	TrackingID field in the TrackingT table	NO
trackcarrier	Shipper associated with the tracking number SEE NOTE BELOW!	Carrier field in the Tracking table	NO
trackpickupdate	Date the package was shipped SEE NOTE BELOW!	PickupDate field in the Tracking table	NO
trackcount	Quantity of tracking numbers included in the post SEE NOTE BELOW!	Programmatically calculated	NO
omversion	Version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

IMPORTANT NOTE: The trackcount variable will identify the total number of tracking numbers included in the post. If this variable is zero, then the tracknum, trackcarrier and trackpickupdate variables will not be present in the post. If set to one then the variable names will appear as listed above. If there are multiple tracking numbers associated with an order, the tracknum, trackcarrier and trackpickupdate variables will have a number appended to their name such as tracknum1, tracknum2, etc. for each tracking number included in the post.

In the examples below, revise the following data points prior to testing:

Replace "www.mystore.com/script.ext:" with the appropriate URL pointing to the script on your web server.

Replace "auser" with the proper user name for login on the website.

Replace "pwd" with the proper password for login to the website.

Replace "mystore" with the store code identifier (optional).

Replace "webordernumber" with an order number from the website

Example URL for testing from browser – uses "GET" method:

<https://www.mystore.com/script.ext?setifunction=updatestatus&setiuser=auser&password=pwd&code=mystore&ordernumber=webordernumber&orderstatus=Shipped&refnumber=1021&orderdetail=Date%20Shipped%58%206/1/2003&trackcount=1&tracknum=1Z9876543218754187&trackcarrier=UPS&trackpickupdate=6/1/2003&omversion=5.000>

Example Form Post – uses "POST" method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the "Send" button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="qohreplace">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "ordernumber" VALUE = "webordernumber">
  <INPUT TYPE=TEXT NAME = "orderstatus" VALUE = "Shipped">
  <INPUT TYPE=TEXT NAME = "refnumber" VALUE = "1021">
  <INPUT TYPE=TEXT NAME = "orderdetail" VALUE = "Date%20Shipped%58%206/1/2003">
  <INPUT TYPE=TEXT NAME = "trackcount" VALUE = "1">
  <INPUT TYPE=TEXT NAME = "tracknum" VALUE = "1Z9876543218754187">
  <INPUT TYPE=TEXT NAME = "trackcarrier" VALUE = "UPS">
  <INPUT TYPE=TEXT NAME = "trackpickupdate" VALUE = "6/1/2003">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.000">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

If the GenericSendStatusXML parameter is set to True:**Form Variables for XML post:**

Variable Name	Value	From	Required
setifunction	updatestatus	Hard coded into visual basic, cannot be modified	YES
setiuser	A username passed for access validation	Shopping Carts form CartID field	NO
password	A password passed for access validation	Shopping Carts form Password field	NO
code	A unique identifier for the store you are downloading from	Shopping Carts Form StoreCode field	NO
update	XML structure defined in StatusExport.xsd	Programmatically generated	YES
omversion	Version number of the Order Manager	Value of the Version field in the last record in the Version table – Used for version control	NO

The “update” variable will contain the following XML structure as defined in the StatusExport.xsd schema file.

Order Status Export XML Structure contained in the “update” form variable.			
Element Name	Description	Restrictions	Required
Orders	Container Element for one or more orders		YES
Orders/Order	Container element for a single order		YES
Orders/Order/ OrderNumber	Web Order Number		YES
Orders/Order/ ReferenceNumber	The Order Manager's Order Number	Integer value	YES
Orders/Order/Status	The current status label applied to the order	50 characters	YES
Orders/Order/Notes	Additional status information from OrdStatDetail field	255 characters	NO
Orders/Order/ Comments	Text from the Comments field in the Orders table		NO
Orders/Order/ ChangeDateTime	Date and time the status event triggered the notice	dd-mmm-yyyy hh:mm:ss format	NO

Packages Element is the container element for one or more tracking numbers. Packages Element is optional and if present is found only once per order element			
Orders/Order/ Packages	Container element for one or more tracking numbers		NO
Orders/Order/ Packages/Package	Container element for each tracking number – One or more Package elements may be present		YES - If parent is present
Orders/Order/ Packages/Package/ PackageID	Unique ID across all tracking numbers in an Order Manager data file		YES
Orders/Order/ Packages/Package/ TrackingID	Tracking Number		YES
Orders/Order/ Packages/Package PickupDate	Date package was shipped		YES
Orders/Order/ Packages/Package/ Shipper	Carrier for the shipment		YES
Orders/Order/ Packages/Package/ Method	Shipping Method used		YES

Items Element is the container element for one or more line items in the order. Items Element is optional and if present is found only once per order element			
Orders/Order/ Items	Container element for one or more line items on the order		NO
Orders/Order/ Items/Item	Container element for a single line item		YES if parent is present
Orders/Order/ Items/Item/ ItemNumber	The line Item ID Number from the website		YES
Orders/Order/ Items/Item/ RefNumber	The Order Manager's Line Item ID number		YES
Orders/Order/ Items/Item/ Status	The status label for the line item		YES
Orders/Order/ Items/Item/ Ordered	The quantity of the item purchased by the customer		YES
Orders/Order/ Items/Item/ Shipped	The quantity available to ship to the customer		YES
Orders/Order/ Items/Item/ Needed	The quantity remaining in backorder		YES
Orders/Order/ Items/Item/ Notes	Additional status data from ItemStatDetail field in the Order Details table		YES

Packages Element is the container element for one or more tracking number cross references. Packages Element is optional and if present is found only once per item element			
Orders/Order/ Items/Item/ Packages	Container element for one or more packages associated with the line item. NOTE: If packing data is available in the Order Manager, the XML will show which package the item is in. If packing data is not available then the program assigns one tracking number per line item		NO
Orders/Order/ Items/Item/ Packages/ Package	Container element for a single tracking number cross reference		YES if parent is present
Orders/Order/ Items/Item/ Packages/ Package/ PackageID	Unique Package ID number (same value found in Orders/Order/Packages/PackageID)		YES
Orders/Order/ Items/Item/ Packages/ Package/ Quantity	Quantity of the item placed in package		YES

Example XML:

```

<Orders>
  <Order>
    <OrderNumber>12548</OrderNumber>
    <ReferenceNumber>1001</ReferenceNumber>
    <Status>Shipped</Status>
    <Notes>Date Shipped: 6/1/2003</Notes>
    <Comments>Entry at rear of building</Comments>
    <ChangeDateTime>01-Jun-2003 13:11:51</ChangeDateTime>
    <Packages>
      <Package>
        <PackageID>6012</PackageID>
        <TrackingID>1Z9876543218754187</TrackingID>
        <PickupDate>6/1/2003</PickupDate>
        <Shipper>UPS</Shipper>
        <Method>Ground</Method>
      </Package>
    </Packages>
    <Items>
      <Item>
        <ItemNumber>54871</ItemNumber>
        <RefNumber>1</RefNumber>
        <Status>Shipped</Status>
        <Ordered>5</Ordered>
        <Shipped>5</Shipped>
        <Needed>0</Needed>
        <Notes>Date Shipped: 6/1/2003</Notes>
        <Packages>
          <Package>
            <PackageID>6012</PackageID>
            <Quantity>5</Quantity>
          </Package>
        </Packages>
      </Item>
    </Items>
  </Order>
</Orders>

```

In the examples below, revise the following data points prior to testing:

Replace "www.mystore.com/script.ext:" with the appropriate URL pointing to the script on your web server.

Replace "auser" with the proper user name for login on the website.

Replace "pwd" with the proper password for login to the website.

Replace "mystore" with the store code identifier (optional).

Replace the appropriate data points in the XML document

Example URL for testing from browser – uses "GET" method:

```
https://www.mystore.com/script.ext?setifunction=updatestatus&setiuser=auser&password=pwd&code=mystore
&omversion=5.000&update=<Orders><Order><OrderNumber>12548</OrderNumber>
<ReferenceNumber>1001</ReferenceNumber><Status>Shipped</Status>
<Notes>Date Shipped: 6/1/2003</Notes><Comments>Entry at rear of building</Comments>
<ChangeDateTime>01-Jun-2003 13:11:51</ChangeDateTime>
<Packages><Package><PackageID>6012</PackageID><TrackingID>1Z9876543218754187</TrackingID>
<PickupDate>6/1/2003</PickupDate><Shipper>UPS</Shipper><Method>Ground</Method></Package>
</Packages><Items><Item><ItemNumber>54871</ItemNumber><RefNumber>1</RefNumber>
<Status>Shipped</Status><Ordered>5</Ordered><Shipped>5</Shipped><Needed>0</Needed>
<Notes>Date Shipped: 6/1/2003</Notes><Packages><Package><PackageID>6012</PackageID>
<Quantity>5</Quantity></Package></Packages></Item></Items></Order></Orders>
```

Example Form Post – uses "POST" method:

After making revisions specified above, save the text below to a file on your hard drive (e.g. c:\temp\formpost.htm) then open the file from your browser. Click on the "Send" button to send the request.

```
<FORM ACTION="www.mystore.com/script.ext" METHOD=POST>
  <INPUT TYPE=TEXT NAME = "setifunction" VALUE="qohreplace">
  <INPUT TYPE=TEXT NAME = "setiuser" VALUE = "auser">
  <INPUT TYPE=TEXT NAME = "password" VALUE = "pwd">
  <INPUT TYPE=TEXT NAME = "code" VALUE = "mystore">
  <INPUT TYPE=TEXT NAME = "update" VALUE = "<Orders><Order><OrderNumber>12548</OrderNumber>
  <ReferenceNumber>1001</ReferenceNumber><Status>Shipped</Status>
  <Notes>Date Shipped: 6/1/2003</Notes><Comments>Entry at rear of building</Comments>
  <ChangeDateTime>01-Jun-2003 13:11:51</ChangeDateTime>
  <Packages><Package><PackageID>6012</PackageID><TrackingID>1Z9876543218754187</TrackingID>
  <PickupDate>6/1/2003</PickupDate><Shipper>UPS</Shipper><Method>Ground</Method>
  </Package></Packages><Items><Item><ItemNumber>54871</ItemNumber>
  <RefNumber>1</RefNumber><Status>Shipped</Status><Ordered>5</Ordered><Shipped>5</Shipped>
  <Needed>0</Needed><Notes>Date Shipped: 6/1/2003</Notes><Packages><Package>
  <PackageID>6012</PackageID><Quantity>5</Quantity></Package></Packages>
  </Item></Items></Order></Orders">
  <INPUT TYPE=TEXT NAME = "omversion" VALUE = "5.500">
  <INPUT TYPE=SUBMIT VALUE=Send>
</FORM>
```

Expected Response:

If the update was processed successfully then return the following string:

SETIResponse: update=OK;Notes=

If the update was unsuccessful, then send the following string. Replace “ErrorMessage” with details regarding the error condition causing the failure.

SETIResponse: update=False;Notes=*ErrorMessage*