ShipWorks 3.0

Store Integration Guide

version 1.0

Contents

Introduction 3
ShipWorks 3
Generic Store Type 4
Generic Store Protocol 4
Response Formats 4
Integration Lifecycle 5
Call Reference 10
GetModule 10
GetStore 10
GetStatusCodes 10
GetCount 11
GetOrders 11
UpdateStatus 12
UpdateShipment 12
ShipWorks Schema v1.0.0 13

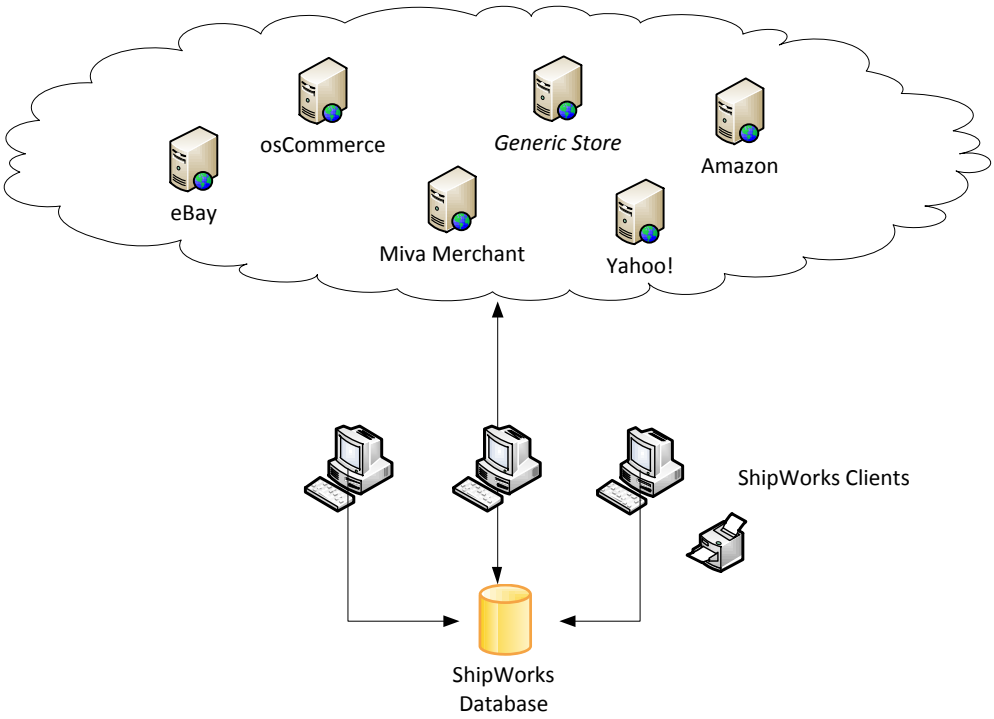
Introduction

The purpose of this guide is to introduce you to the ShipWorks concepts and technical details necessary to build an integration layer ShipWorks can use to interact with your system. Questions can be directed to support@interapptive.com.

ShipWorks

ShipWorks is Windows-based desktop software which ecommerce professionals use to download and process orders from their online stores, print shipping labels, generate invoices, and manage customer information. A user can have any number of supported shopping cart systems registered in ShipWorks, and the software will download orders from each in turn.

Orders pulled into ShipWorks consist of customer demographic information like shipping/billing address, items purchased, payments made on those items, and other charges or discounts relating to the online purchase. As shipments are processed and labels created in ShipWorks, the details are recorded as Shipments in the software and optionally uploaded back to the online store from where the orders originated.



There are over two dozen ecommerce systems which ShipWorks supports. These are made available to the user in the Manage Stores window when the Add Store button is selected. ShipWorks uses native APIs to communicate with many of these systems (eBay, Amazon.com, etc.), using standards like SOAP Web Services. The integration you are building will be used by the extension point in ShipWorks known as the Generic Store Type.

Generic Store Type

ShipWorks will communicate with your store using the *Generic Store Protocol*, using HTTP GETs and POSTs to exchange XML documents between the two systems. Once your integration is complete, users will be able to access it in ShipWorks via the Add Store Wizard by specifying the store type as Generic and providing the URL to your integration.

If the protocol is followed, ShipWorks will be able to download orders, order items, charges, and payments from your system as well as provide you with tracking numbers and status updates as they are generated by the ShipWorks user.

Generic Store Protocol

ShipWorks communicates with Generic Store Integrations using HTTP POSTs to a single integration endpoint URL. It is highly recommended that this endpoint be secured with SSL, although this is not a requirement.

Each POST will contain a collection of variables: username; password; *action*; and action-specific parameters. The *action* parameter can be thought of much like a function name, it is the activity to be executed. Throughout this document, when "calling XXXX" is stated it simply means making a request with action = *XXXX*.

Response Formats

Each response from your integration is an XML document that must conform to the published ShipWorks XML Schema. ShipWorks will not process responses that are not valid per the schema, and you will be notified by an error message in ShipWorks.

As indicated in the ShipWorks Schema, the root element for each response is the ShipWorks node. This node contains a pair of required attributes.

- *moduleVersion* is a versioning mechanism for you to track various versions of your integration. In addition, ShipWorks 3 requires that the module version be at least 3.0.0 before it will communicate with the integration.
- *schemaVersion* is where you specify which ShipWorks Schema version your response will validate against. As the Generic Store extension point evolves additions will be made in new versions of the ShipWorks Schema, and this is for backward compatibility.

In the descriptions that follow, not every option available will be discussed at length. For the full XML Schema with annotations, see the section ShipWorks Schema v1.0.0 at the end of this document.

Integration Lifecycle

The protocol is best understood by following its lifecycle - from setting up a Store in ShipWorks, to downloading orders, to updating order status, to sending tracking numbers back to your ecommerce system.

Store Setup

The user walks through the Add Store Wizard after selecting Generic Store as the store type. When they provide connection information to your integration endpoint, ShipWorks will make a sequence of calls to your integration to obtain details about your solution's capabilities, the demographic details about the user's online store, and a collection of the store's status codes if necessary.

Call 1: GetModule (action=getmodule)

This call is made to determine the capabilities of your integration and allows ShipWorks to correctly configure some of its behaviors with respect to your store.

In the response XML you can specify a Download Strategy, whether or not your store supports Order Statuses, if your customers can be identified by a Customer ID, and whether or not shipment details can be posted back when they are available.

Sample GetModule Response:

```
<?xml version="1.0" standalone="yes" ?>
<ShipWorks moduleVersion="3.0.1" schemaVersion="1.0.0">
  <Module>
    <Platform>Your Platform Name</Platform
    <Developer>Interapptive, Inc. (support@interapptive.com)</Developer>
    <Capabilities>
        <DownloadStrategy>ByModifiedTime</DownloadStrategy>
        <OnlineCustomerID supported="true" dataType="numeric" />
      <OnlineStatus supported="true" dataType="numeric" supportsComments="true" />
        <OnlineShipmentUpdate supported="false" />
    </Capabilities>
  </Module>
</ShipWorks>
```

Platform

This identifies your ecommerce system to Interapptive and should be the name of your product.

Developer

This is your company name and contact information.

DownloadStrategy

ShipWorks supports two methodologies for downloading orders from Generic Stores – ByModifiedTime and ByOrderNumber.

ByModifiedTime

With a ByModifiedTime download strategy, ShipWorks will look at all of the past orders it has downloaded from the store and calculate the most recent Modified Time. It will

then ask your integration for all orders that are new OR have been modified since that time.

This mechanism will allow things like order status updates and address changes made in your online store to be re-downloaded and reflected in the local ShipWorks copy of the order.

ByOrderNumber
Since not all ecommerce systems keep track of when an order was last modified, ShipWorks can also retrieve based on Order Number. This is only suitable when order numbers are incremental, numeric values, as ShipWorks will simply find the highest order number previously downloaded and ask for all orders with a larger order number.

One downside to ByOrderNumber is that downloading order updates becomes much more complicated and inefficient, requiring more state to be managed by your system. As such, it is highly recommended to use the ByModifiedTime strategy whenever possible.

OnlineCustomerID

If your store tracks customers by an identifier, specify supported="true" to enable ShipWorks to perform customer matching in its database based on online customer ID instead of email address. Specifying a data type will indicate the format of the OnlineCustomerID node you will be providing in the GetOrders response discussed later.

OnlineStatus

ShipWorks can retrieve a mapping of order status codes and their display names. Orders downloaded during the GetOrders call will then have their StatusCode translated to the user-friendly display name.

supported
Specifies if order status codes should be retrieved and handled by ShipWorks.

dataType
numeric or text to indicate the format of the StatusCode provided in the GetOrders response.

downloadOnly
Specifies if the order statuses are effectively read-only, meaning ShipWorks will not allow the user to update the online order statuses from within ShipWorks.

supportsComments
If downloadOnly is false or missing, meaning order statuses are updatable, this will control whether or not the user can specify a comment to accompany the status update. The user-entered comment will be sent along with the status update request.

OnlineShipmentUpdate

This will control whether ShipWorks can send shipment tracking information back to the online store. This includes allowing the user to manually send the update via right-clicking on an order as well as the Action Task for automatically sending the details as soon as a shipment is processed in ShipWorks.

Call 2: GetStore

The next call is made to retrieve the demographic and contact information for the user's online store. This information gets stored in ShipWorks and is used for shipping and printing invoices, among other things.

See the XML Schema for the response fields and format.

Call 3: GetStatusCodes

If OnlineStatus was specified as supported in *Call 1: GetModule,* ShipWorks will request the collection of available status codes and their associated display text. This data is stored in ShipWorks and will be refreshed periodically in future download operations.

See the XML Schema for the response fields and format.

Order Download

The following operations occur on each download cycle – meaning each time the user clicks the Download button in ShipWorks or the scheduled automatic download time passes.

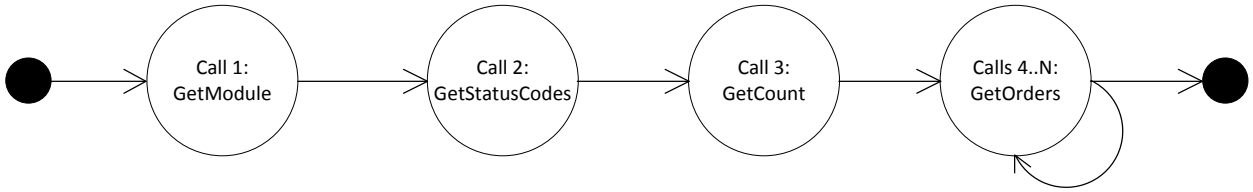


Figure 1 Download Cycle

Call 1: GetModule

ShipWorks makes this call to ensure the module capabilities are kept up-to-date.

Call 2: GetStatusCodes

This call will be made only when OnlineStatus was marked as supported in the GetModule response. This happens on every download cycle to make sure all status codes are up-to-date before download the orders which await.

Call 3: GetCount

ShipWorks makes the request to determine how many orders exist to be downloaded. The count returned is used to calculate and display download progress to the user. **If the returned count is 0, the download operation is considered complete and no further calls are made.**

Call 4…N: GetOrders

ShipWorks will continue making GetOrders calls, repeatedly, until no orders are returned in the response. If orders are continually returned to ShipWorks, the download will proceed infinitely.

The *maxcount* parameter is simply a requested batch size, or number of orders desired to be in this call's response. Please beware this is simply a recommended response size – when the download strategy is ByModifiedTime care must be taken to ensure that orders with matching Modified Time do not span batches. **This would result in orders being skipped since ShipWorks would re-request orders with a new *start* value on the next GetOrders call which would not include those skipped orders.** If this scenario arises, simply return as many orders as necessary without regard to the *maxcount* value.

Once again, downloading will be considered complete when no orders are returned.

Status and Shipment Updates

If the integration has indicated that it supports status, via its response to GetModule, ShipWorks will allow the user to change an order's online status. Similarly, the posting of shipment tracking information is enabled if the GetModule response indicates OnlineShipmentUpdate is supported.

Call: UpdateStatus

This call is made once per order being updated. The module should return a success or error response to ShipWorks once the operation is complete.

Call: UpdateShipment

This call is made once per shipment processed for an order. Like the UpdateStatus call, the module should return a success or error response to ShipWorks once the operation is complete.

Success Response

If an update operation is successful, the response should be as follows.

```
<?xml version="1.0" standalone="yes" ?>
<ShipWorks moduleVersion="3.0.0" schemaVersion="1.0.0">
  <UpdateSuccess/>
</ShipWorks>
```

Error Response

If an error occurs during any module operations, including the updates being discussed here, an Error should be returned as follows.

```
<?xml version="1.0" standalone="yes" ?>
<ShipWorks moduleVersion="3.0.0" schemaVersion="1.0.0">
  <Error>
    <Code>FOO100</Code>
    <Description>Something Failed. Internal Error.</Description>
  </Error>
</ShipWorks>
```

Call Reference

GetModule

Used by ShipWorks to determine the capabilities and behavior specifications of your integration.

This gets called once during store setup and again at the very start of each download cycle. This is done so that changes can be made to your integration and ShipWorks will be aware of changes as soon as possible, without user interaction.

POST Variables

Variable	Value
username	<user-entered password>
password	<user-entered password>
action	getmodule

GetStore

ShipWorks uses this to prepopulate store demographic information for the user during store setup.

POST Variables

Variable	Value
username	<user-entered password>
password	<user-entered password>
action	getstore

Response

Your integration should respond with a ShipWorks XML document containing a Module element.

GetStatusCodes

If your GetModule response indicates that status codes are supported (see Call 1: GetModule), ShipWorks needs to cache status codes → status display text and uses this request to do so.

This gets called once during store setup and at the start of each download cycle to ensure the local status code cache is not stale.

POST Variables

Variable	Value
username	<user-entered password>
password	<user-entered password>
action	getstatuscodes

Response

Your integration should respond with a ShipWorks XML document containing a StatusCodes element.

GetCount

To determine if orders exist to be downloaded, ShipWorks makes this call. If 0 is returned, ShipWorks will not proceed further in the download cycle.

POST Variables

Variable	Value
username	<user-entered password>
password	<user-entered password>
action	getcount
start	<last order number previously downloaded, or the highest LastModified date ShipWorks has downloaded>

The start parameter will either be an integer or a date/time value. When the Download Strategy in the GetModule response you provide is ByOrderNumber, ShipWorks will request those orders with an Order Number higher than *start*. A Download Strategy of ByModifiedTime will result in ShipWorks sending a date/time value that is the most recent Order LastModified value it has previously downloaded. You will use this *start* parameter to how many orders exist to be downloaded.

Response

Your integration should respond with a ShipWorks XML document containing a OrderCount element.

GetOrders

ShipWorks will make repeated calls to GetOders to download order data, and will cease when the response contains no orders.

POST Variables

Variable	Value
username	<user-entered password>
password	<user-entered password>
action	getorders
start	<last order number previously downloaded, or the highest LastModified date ShipWorks has downloaded>
maxcount	Desired results-page size, default is 50.

The start parameter will either be an integer or a date/time value. When the Download Strategy in the GetModule response you provide is ByOrderNumber, ShipWorks will request those orders with an Order Number higher than *start*. A Download Strategy of ByModifiedTime will result in ShipWorks sending a date/time value that is the most recent Order LastModified value it has previously downloaded. You will use this *start* parameter to determine which orders to provide to ShipWorks.

Response

Your integration should respond with a ShipWorks XML document containing Order elements.

UpdateStatus

If your integration has indicated that it supports order status updates, ShipWorks will make this request when the user decides to update the status. This can be done manually on the Order grid in ShipWorks or automatically via an Action Task.

POST Variables

Variable	Value
username	<user-entered password>
password	<user-entered password>
action	updatestatus
order	OrderNumber to update the status for.
status	StatusCode to assign to the order.
comments	Comments to apply with the update.

The UpdateStatus call will only be sent if the GetModule response indicates that your integration supports it. Comments will be sent only if supportsComments is true in the GetModule response.

Response

Your integration should respond with a ShipWorks XML document containing an UpdateSuccess element or an Error element.

UpdateShipment

If your integration has indicated that it supports shipment updates, ShipWorks will make this request when the user chooses to upload shipment details. This can be done manually on the Order grid in ShipWorks or automatically via an Action Task.

POST Variables

Variable	Value
username	<user-entered password>
password	<user-entered password>
action	updateshipment
order	OrderNumber the tracking number applies to.
tracking	Tracking number for the processed shipment.

ShipWorks Schema v1.0.0

The ShipWorks Schema can be downloaded at http://www.interapptive.com/shemas/ShipWorks1_0_0.xsd and is included here.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="ShipWorksModule"
    elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema”>

  <xs:element type="ShipWorks" name="ShipWorks"/>

  <xs:complexType name="ShipWorks">
  <xs:annotation>
    <xs:documentation>Root of all integration responses.</xs:documentation>
  </xs:annotation>

    <xs:sequence>
      <xs:element name="Parameters" minOccurs="0" >
        <xs:annotation>
          <xs:documentation>Container for echoing received parameters back to ShipWorks.  Useful for support staff.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:any minOccurs="0" maxOccurs="unbounded" processContents="skip"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:element name="Module" type="Module">
          <xs:annotation>
            <xs:documentation>Response element for GetModule call.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Store" type="Store">
          <xs:annotation>
            <xs:documentation>Response element for GetStore call.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="StatusCodes">
          <xs:annotation>
            <xs:documentation>Response element for GetStatusCodes call.</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="StatusCode" minOccurs="0" maxOccurs="unbounded" type="StatusCode"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="OrderCount" type="xs:int">
```

```
          <xs:annotation>
            <xs:documentation>Response element for GetCount call.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Orders">
          <xs:annotation>
            <xs:documentation>Response element for GetOrders call.</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Order" minOccurs="0" maxOccurs="unbounded" type="Order" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="UpdateSuccess">
          <xs:annotation>
            <xs:documentation>Response element for a successful UpdateStatus or UpdateShipment call.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Error">
          <xs:annotation>
            <xs:documentation>Response element for any fatal errors that occured in the processing of any requests.</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Code" type="xs:string" >
                <xs:annotation>
                  <xs:documentation>The error code will appear in logs and has no meaning to ShipWorks.</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="Description" type="xs:string" >
                <xs:annotation>
                  <xs:documentation>Error message that will be presented to the user when encountered by ShipWorks.</xs:documentation>
                </xs:annotation>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Debug" type="Debug" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Container for passing any debugging or support information back to ShipWorks.  Nothing contained here will be used by ShipWorks.
Responses will be logged by ShipWorks, so this should not contain sensitive information.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:choice>
    </xs:sequence>

    <xs:attribute name="moduleVersion" type="Version" use="required">
      <xs:annotation>
        <xs:documentation>The version of this integration module.  Must be at least 3.0.0 for ShipWorks 3, or an error will be raised.</xs:documentation>
      </xs:annotation>
    </xs:attribute>

    <xs:attribute name="schemaVersion" type="Version" use="required">
```

```
      <xs:annotation>
        <xs:documentation>ShipWorks Schema version to validate the response to.   Currently 1.0.0.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="Module">
    <xs:annotation>
      <xs:documentation>Descriptive information about the module and its developer, including the module's capabilities.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Platform" type="RequiredString">
          <xs:annotation>
            <xs:documentation>The ecommerce system this module allows ShipWorks to integrate with.</xs:documentation>
          </xs:annotation>
      </xs:element>
      <xs:element name="Developer" type="RequiredString">
          <xs:annotation>
            <xs:documentation>The company or individual name of the integration developer.</xs:documentation>
          </xs:annotation>
      </xs:element>
      <xs:element name="Capabilities">
          <xs:annotation>
            <xs:documentation>Declares the functionality available in the integration.  Options specified here setup expectations in ShipWorks such as the
availability of certain data elements, or features as a whole.</xs:documentation>
          </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="DownloadStrategy">
              <xs:annotation>
                <xs:documentation>Dictates how ShipWorks will retrieve orders from your system.

                ByModifiedTime: ShipWorks will request orders placed or modified since the most recent LastModified order it has downloaded in the past.

                ByOrderNumber: ShipWorks will request orders having order numbers larger than the highest order number it has downloaded in the past.

                Consult the Store Integration Guide for more information on the implications of this setting.
                </xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="ByModifiedTime">
                    <xs:annotation>
                      <xs:documentation>ShipWorks will request orders placed or modified since the most recent LastModified order it has downloaded in the
past.</xs:documentation>
                    </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="ByOrderNumber">
                    <xs:annotation>
                      <xs:documentation>ShipWorks will request orders having order numbers larger than the highest order number it has downloaded in the
past.</xs:documentation>
                    </xs:annotation>
                  </xs:enumeration>
                </xs:restriction>
```

```
            </xs:simpleType>
          </xs:element>
          <xs:element name="OnlineCustomerID">
            <xs:annotation>
              <xs:documentation>If your system assigns unique identifier to customers, ShipWorks can use this identifier to match up orders to customers in the
ShipWorks database.  Customer linking is done by email address otherwise.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:attribute name="supported" type="xs:boolean" use="required">
                <xs:annotation>
                  <xs:documentation>Specifies whether or not CustomerID will be provided in GetOrder respones.</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="dataType" type="VariantDataType" use="optional">
                <xs:annotation>
                  <xs:documentation>Tells ShipWorks what type of data to expect in the CustomerID element of the GetOrder respones.</xs:documentation>
                </xs:annotation>
              </xs:attribute>
            </xs:complexType>
          </xs:element>
          <xs:element name="OnlineStatus">
            <xs:annotation>
              <xs:documentation>ShipWorks can do more than just display order statuses.  Depending on the configuration specified here, ShipWorks can allow users
to change order status with or without comments.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:attribute name="supported" type="xs:boolean" use="required">
                <xs:annotation>
                  <xs:documentation>Specifies whether or not order statuses should be downloaded and displayed.</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="supportsComments" type="xs:boolean" use="optional">
                <xs:annotation>
                  <xs:documentation>Specifies if ShipWorks users can provide comments when doing order status updates.  Ignored if downloadOnly is
true.</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="downloadOnly" type="xs:boolean" use="optional">
                <xs:annotation>
                  <xs:documentation>Specifies if order statuses are read-only, not allowing the user to make changes to the online status.  If this is true, no
UpdateStatus calls will ever be made by ShipWorks.</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="dataType" type="VariantDataType" use="optional">
                <xs:annotation>
                  <xs:documentation>Tells ShipWorks what type of data to expect in the StatusCode element of the GetOrder response.</xs:documentation>
                </xs:annotation>
              </xs:attribute>
            </xs:complexType>
          </xs:element>
          <xs:element name="OnlineShipmentUpdate">
            <xs:annotation>
              <xs:documentation>ShipWorks can send shipment tracking information to the integration, either automatically when orders are shipped or manually by
the user.</xs:documentation>
```

```
            </xs:annotation>
            <xs:complexType>
              <xs:attribute name="supported" type="xs:boolean" use="required">
                <xs:annotation>
                  <xs:documentation>Dictates whether or not shipment details should be sent to the integration.  If this is true, no UpdateShipment calls will
ever be made by ShipWorks.</xs:documentation>
                </xs:annotation>
              </xs:attribute>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Communications" minOccurs="0">
        <xs:annotation>
          <xs:documentation>This is used to configure any communications-level settings ShipWorks needs in order to fully communicate with the
module.</xs:documentation>
        </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Http" minOccurs="0">
            <xs:complexType>
              <xs:attribute name="expect100Continue" type="xs:boolean" use="optional" default="true">
                <xs:annotation>
                  <xs:documentation>By default ShipWorks will send the expect:100Continue HTTP Header on its POST requests.  If this causes issues for your
system, set this to false and it will no longer be sent.</xs:documentation>
                </xs:annotation>
              </xs:attribute>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Store">
  <xs:annotation>
    <xs:documentation>Demographic information about the user's online store.  This information is made available to the ShipWorks user during shipping and
invoicing.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Name" type="xs:string">
        <xs:annotation>
          <xs:documentation>The name of the oniline store.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="CompanyOrOwner" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The individual or company owner's name.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Email" type="xs:string" minOccurs="0">
        <xs:annotation>
```

```
          <xs:documentation>Typically the support/customer service email address for the store.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Street1" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Physical street address.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Street2" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Physical street address.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Street3" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Physical street address.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="City" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>City where the store is located.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="State" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>State where the store is located.  State code or full name.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="PostalCode" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Postal code where the store is located.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Country" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Country where the store is located. Country code or full name.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Phone" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Typically the store's customer service phone number.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Website" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The online store's customer-facing URL.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="StatusCode">
    <xs:annotation>
      <xs:documentation>Container element for a single status code.</xs:documentation>
```

```
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Code" type="RequiredString">
        <xs:annotation>
          <xs:documentation>The internal representation of your system's status code.  For example 1001, SHIP, CANCEL2</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Name" type="RequiredString">
        <xs:annotation>
          <xs:documentation>The display name for the status code.  This value is presented as the status code in ShipWorks and should be identical to how it is
displayed in your system to avoid confusion.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Order">
    <xs:annotation>
      <xs:documentation>A single order to be imported into ShipWorks.</xs:documentation>
    </xs:annotation>
    <xs:all>
      <xs:element name="OrderNumber" type="xs:long">
        <xs:annotation>
          <xs:documentation>The numeric, unique order number identifying this order to ShipWorks.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="OrderDate" type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>The date and time the order was originally placed. The time is in UTC.</xs:documentation>
          </xs:annotation>
      </xs:element>
      <xs:element name="LastModified" type="xs:dateTime" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The date and time the order was last modified in the online store, by the end customer or staff.  The time is in UTC.  Required if
DownloadStrategy is defined as ByModifiedTime.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ShippingMethod" type="xs:string">
        <xs:annotation>
          <xs:documentation>Customer-requested shipping method for the order.  This is displayed in ShipWorks and should be human-readable.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StatusCode" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The status code for the current status of the order.  Required when OnlineStatus is designated as supported.  See complexType
Module.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="CustomerID" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>A value uniquely identifying the purchasing customer to ShipWorks.  Required when OnlineCustomerID is designated as supported.  See
complexType Module.</xs:documentation>
          </xs:annotation>
      </xs:element>
```

```
      <xs:element name="Notes" minOccurs="0" >
        <xs:annotation>
          <xs:documentation>A collection of any number of notes to attach to the order in ShipWorks.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Note" type="Note" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ShippingAddress" type="Address">
        <xs:annotation>
          <xs:documentation>The shipping address specified on the order.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="BillingAddress" type="Address">
        <xs:annotation>
          <xs:documentation>The billing address specified on the order.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Payment" minOccurs="0" >
        <xs:annotation>
          <xs:documentation>Container for payment information provided by the customer.  This information is displayed on the Payments panel in the ShipWorks user
interface.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:choice>
            <xs:sequence>
              <xs:element name="Method" type="xs:string">
                <xs:annotation>
                  <xs:documentation>Payment method displayed in ShipWorks.</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="CreditCard" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>If the Method of payment is some type of credit card, the cc information can be supplied here.</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Type" type="xs:string">
                      <xs:annotation>
                        <xs:documentation>The type of credit card, displayed to the user.</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="Owner" type="xs:string">
                      <xs:annotation>
                        <xs:documentation>The name listed as the owner of the card.</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="Number" type="xs:string">
                      <xs:annotation>
                        <xs:documentation>Credit card number provided by the customer.  If your integration allows unsecured connections, this value should not be
provided in whole to ShipWorks.</xs:documentation>
                      </xs:annotation>
```

```
                    </xs:element>
                    <xs:element name="Expires" type="xs:string">
                      <xs:annotation>
                        <xs:documentation>Card expiration date.</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="CCV" type="xs:string" minOccurs="0">
                      <xs:annotation>
                        <xs:documentation>Card verification number, on the back of the card.  Once again, this value should not be provided to ShipWorks if
communication is detected to be unsecure.</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:element name="Detail" minOccurs="0" maxOccurs="unbounded" >
              <xs:annotation>
                <xs:documentation>Used to specify other payment details, gift certificates or reward cards for example.</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:attribute name="name" type="xs:string">
                  <xs:annotation>
                    <xs:documentation>The name of the payment detail item displayed in ShipWorks.  Items are displayed as "Name: Value".  For example "Discount
Code: ABCDEFG"</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
                <xs:attribute name="value" type="xs:string">
                  <xs:annotation>
                    <xs:documentation>The value of the payment detail item displayed in ShipWorks.  Items are displayed as "Name: Value".  For example "Discount
Code: ABCDEFG"</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
              </xs:complexType>
            </xs:element>
          </xs:choice>
        </xs:complexType>
      </xs:element>
      <xs:element name="Items">
        <xs:annotation>
          <xs:documentation>The purchased items on the order; line items.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Item" minOccurs="0" maxOccurs="unbounded" type="Item"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Totals">
        <xs:annotation>
          <xs:documentation>Charges and fees applied to the order that impact its total cost.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
```

```
            <xs:element name="Total" minOccurs="0" maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>A single charge, fee, or discount that impacts the order total.</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:decimal">
                    <xs:annotation>
                      <xs:documentation>The monetary value of this charge.  Do not use negative numbers to apply discounts.  Rather, use the impact attribute to
specify how a value impacts the total.</xs:documentation>
                    </xs:annotation>
                    <xs:attribute name="id" type="xs:int" use="optional">
                      <xs:annotation>
                        <xs:documentation>An identifier for the charge in your system, if desired.</xs:documentation>
                      </xs:annotation>
                    </xs:attribute>
                    <xs:attribute name="name" type="xs:string" use="optional">
                      <xs:annotation>
                        <xs:documentation>Display text for the fee or charge. Ex. Shipping and Handling</xs:documentation>
                      </xs:annotation>
                    </xs:attribute>
                    <xs:attribute name="class" type="xs:string" use="optional">
                      <xs:annotation>
                        <xs:documentation>A code for the charge that can be keyed on in Templates, if desired.  Ex. HANDLING </xs:documentation>
                      </xs:annotation>
                    </xs:attribute>
                    <xs:attribute name="impact" type="xs:string" use="optional" default="add">
                      <xs:annotation>
                        <xs:documentation>Specifies how the value of this charge impacts the order total.  Possible values are: add, subtract, and
none.</xs:documentation>
                      </xs:annotation>
                    </xs:attribute>
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Debug" type="Debug" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Debugging and support data not processed by ShipWorks, but recorded in logs.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:all>
  </xs:complexType>

  <xs:complexType name="Item">
    <xs:annotation>
      <xs:documentation>A distinct purchased item on an order.  Also referred to as a Line Item or Order Item.</xs:documentation>
    </xs:annotation>
    <xs:all>
      <xs:element name="ItemID" type="xs:string" minOccurs="0">
        <xs:annotation>
```

```
          <xs:documentation>Identifier of the item in your system, if desired.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ProductID" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Value that uniquely identifies the item purchased.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Code" type="xs:string">
        <xs:annotation>
          <xs:documentation>Another value to identify the purchased item.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="SKU" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Stock Keeping Unit for the item.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Name" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Display text for the item.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Quantity" type="xs:integer">
        <xs:annotation>
          <xs:documentation>Number of these items purchased.  This value is multiplied by UnitPrice and Weight when calculating total order cost and
weight.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="UnitPrice" type="xs:decimal">
        <xs:annotation>
          <xs:documentation>The price of a single one of these items.  This value is multiplied by Quantity when total order value is
calculated.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="UnitCost" type="xs:decimal" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The cost of a single one of these items.  This value is multiplied by Quantity when total order cost is calculated.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Image" type="xs:anyURI" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The URL to the full product image.  This will allow an image of the item to be placed on invoices and reports.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ThumbnailImage" type="xs:anyURI" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The URL to the product thumbnail image.  This will allow the thumbnail image of the item to be placed on invoices and
reports.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Weight" type="xs:decimal">
        <xs:annotation>
```

```
          <xs:documentation>The weight of a single one of these items.  This value is multiplied by Quantity when the total order weight is
calculated.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Attributes" minOccurs="0" >
        <xs:annotation>
          <xs:documentation>Attributes are for product variations or options selected by the customer.  Ex. size, color, edition.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Attribute" minOccurs="0" maxOccurs="unbounded" type="ItemAttribute"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Debug" minOccurs="0"  type="Debug">
        <xs:annotation>
          <xs:documentation>Debugging and support data not processed by ShipWorks, but recorded in logs.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:all>
  </xs:complexType>

  <xs:complexType name="ItemAttribute">
    <xs:annotation>
      <xs:documentation>Declares a chosen product option or variation.</xs:documentation>
    </xs:annotation>
    <xs:all>
      <xs:element name="AttributeID" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Identifier of the option or variation in your system, if desired.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Name" type="xs:string">
        <xs:annotation>
          <xs:documentation>Display text for the option or variation type.  Ex. Size, Color, Edition</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Value" type="xs:string">
        <xs:annotation>
          <xs:documentation>The customer-chosen value for the option or variation.  Ex. Small, Red, Special</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Price" type="xs:decimal" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The charge for the option or variation.  This value impacts the calculated order total in ShipWorks.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Debug" minOccurs="0"  type="Debug">
        <xs:annotation>
          <xs:documentation>Debugging and support data not processed by ShipWorks, but recorded in logs.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:all>
  </xs:complexType>
```

```
  <xs:complexType name="Note">
    <xs:annotation>
      <xs:documentation>A textual note attached to orders.</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="date" type="xs:dateTime" use="optional">
          <xs:annotation>
            <xs:documentation>The date the note or comment was applied.</xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="public" type="xs:boolean" use="optional" default="false">
          <xs:annotation>
            <xs:documentation>Public notes are printed on invoices, for customer comments and the like.  Private notes are those entered and viewable by store
staff, as they are considered internal in nature.</xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="Address">
    <xs:annotation>
      <xs:documentation>Container for address information.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:group ref="Name">
        <xs:annotation>
          <xs:documentation>The addressee's name.</xs:documentation>
        </xs:annotation>
      </xs:group>
      <xs:element name="Company" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The addressee's company name.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Street1" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Street address.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Street2" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Street address.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Street3" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Street address.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="City" type="xs:string" minOccurs="0">
        <xs:annotation>
```

```
        <xs:documentation>The addressee's city.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="State" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The addressee's state code.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PostalCode" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The addressee's postal code.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Country" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The addressee's country code.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Residential" type="xs:boolean" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Indicator for this being a residential address.  This is used to configure how shipments are processed and rated.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Phone" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Teh addressee's phone number.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Fax" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The addressee's fax number.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Email" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The addressee's email address.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Website" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The addressee's website URL.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:group name="PersonName">
  <xs:annotation>
    <xs:documentation>A person's name broken out into its first, middle, and last name.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="FirstName" type="xs:string"/>
    <xs:element name="MiddleName" type="xs:string" minOccurs="0"/>
    <xs:element name="LastName" type="xs:string"/>
```

```
    </xs:sequence>
  </xs:group>

  <xs:group name="Name">
    <xs:annotation>
      <xs:documentation>A person's name.</xs:documentation>
    </xs:annotation>
    <xs:choice>
      <xs:element name="FullName" type="xs:string">
        <xs:annotation>
          <xs:documentation>When present, ShipWorks will parse the name into its first, middle, and last names.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:group ref="PersonName"/>
    </xs:choice>
  </xs:group>

  <xs:complexType name="Debug">
    <xs:sequence>
      <xs:any minOccurs="0" maxOccurs="unbounded" processContents="skip"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="Version">
    <xs:annotation>
      <xs:documentation>Describes a version string in the format x.y.z</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:pattern value="\d+(\.\d+){1,3}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="Impact">
    <xs:annotation>
      <xs:documentation>Defines how an item impacts the total.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="add"/>
      <xs:enumeration value="subtract"/>
      <xs:enumeration value="none"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="RequiredString">
    <xs:annotation>
      <xs:documentation>A non-empty string.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="VariantDataType">
    <xs:annotation>
```

```
      <xs:documentation>Data types allowed for a the variable pieces of data such as StatusCode and CustomerID.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="numeric"/>
      <xs:enumeration value="text"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```