HubXML USER GUIDE

Introduction

To implement XML messaging as part of your Rithum integrated connection, you'll need two things: HubXML schemas, and an XML toolset that your organization is comfortable using. You'll work directly with your Onboarding Representative to review and test your XML messaging, but this guide will give you the background you need to begin programming.

Tip: *If you're new to XML and/or HubXML, visit the appendix for an overview of these concepts.*

Prerequisites

You'll need the following in order to proceed:

- HubXML schemas (**.xsd** files)
- An XML Editor
- A validating parser tool (if you are using a master schema for validation)

HubXML Schemas

Your development effort will be driven by HubXML schemas, which are delivered as Rithum-maintained **.xsd** files. A pair of schemas is available for each message that you decide to integrate using HubXML, as described below.

Schema Type	Description	Where can I get this schema?
Master	Represents the full dataset for a given message type, regardless of which portion of the dataset is in use for a specific retailer. This is the only type of schema that should be used for automated validation of incoming messages.	Contact your Onboarding Representative.
Retailer-specific	Represents a subset of the master schema that the retailer intends to use. Because the schema is non-binding (e.g. the retailer may populate elements that they previously indicated that they did not intend to populate) and is subject to change without notice, it should not be used for automated validation of incoming messages.	Rithum Resource Center

Important: *If you receive data from a retailer that aren't included on their retailer-specific schema, you may ignore that data. However, you should not reject the message based on that data being present. Additionally, data sent to a retailer in an element that are defined in the master schema, but not in the retailer-specific schema, will be ignored unless there is an impact on financial calculations. If an impact on financial calculations is possible, the message may be rejected.*

HubXML Formatting

Rule	Example
XML version and encoding should be declared in the message header of all files. ***Note:*** Files **submitted** to Rithum should explicitly declare the encoding used by the process you used to generate the XML. Files **received** from Rithum will be encoded using UTF-8.	**<?xml version="1.0" encoding-"UTF-8"?>**
XML documents must have a root element that matches the root element of the schema used to validate the document.	**<ConfirmMessageBatch>**
XML elements where the schema has defined a data type (e.g. string) will have an opening and closing tag.	In the following example, the <shipDate> element is defined with a string data type. **<shipDate>20140325</shipDate>**
XML elements where the schema does not define a data type AND has no child elements are expressed with a single tag that ends with a forward slash before the closing bracket.	In the following example, the <shipTo> element has no defined data type, but does have an attribute of *personPlaceID* that is assigned a value of **PPI_01**. **<shipTo personPlaceID="PPI_01"/>**
Attribute values must be in quotes.	**<company industry = "automobile">**
XML tags are case-sensitive.	**<name>** and **<Name>** are not equivalent.
XML elements must be properly nested.	In the following example, the <shipTo> element is a child of the <shipmentDetail> element. **<shipmentDetail shipmentDetailID="SDI_01">** **<shipTo personPlaceID="PPI_01"/>** **</shipmentDetail>**

Rule	Example
XML elements can be linked (see the Linking section below)	In the following example, **<shipTo personPlaceID="PPI_01"/>** is an IDREF link to **<personPlace personPlaceID="PPI_01">** ID attribute **<name1>Joe</name1>** **<name2>Jackson</name2>** **</personPlace>**
Trading partners are identified using <participatingParty> elements. On messages submitted to Rithum, the *participationCode* attribute must be "To:", and the *roleType* attribute must indicate the type of partner that the message is being sent To:	**<participatingParty roleType="merchant" participationCode="To:">RetailerID</participatingParty>**
Dates and date/time elements are defined as string datatypes, and must be formatted appropriately.	Format dates as **CCYYMMDD** Format date/time according to ISO-8601 rules: **CCYY-MM-DDTHH:MM:SS+/-OH:OM** **Where** The **T** character separates the date and time. The **+** or **-** character indicates whether the UTC offset is positive or negative. **OH r**epresents the UTC offset hours. **OM** represents the UTC offset minutes.
Some characters have meaningful syntactical significance in XML. If the character is part of the data that is being expressed within the elements or attributes of the document, then it must be replaced with an entity reference.	Format greater than (**>**) as **>** Format less than (**<**) as **<** Format ampersand (**&**) as **&** Format apostrophe (**'**) as **'** Format quotation mark (**"**) as **"**

Linking

HubXML uses one type of element linking in particular: ID and IDREF attributes. This is used mostly to define addresses of business parties, such as a Ship-to and Bill-to location, or possibly a returns address. The XML standard for ID and IDREF attribute values includes, but is not limited to, the following rules:

- The value must begin with an alpha character

- ID attribute values must be unique **within the entire document** (or file)
- With HubXML messaging, IDREF links should only reference an ID element that exists within the same Rithum message as the link. Multiple IDREF links within a Hub message **can** reference the same ID (e.g. Ship-to and Bill-to for a given Purchase Order are commonly the same address), as long as that ID is contained in the same message as the link.

Whitespace

Whitespace (non-meaningful characters, such as spaces, tabs, carriage returns, and line feeds) is used to make XML data more human-readable. It can appear in a structure:

- After the opening tag of a compound element and before the opening tag of the first child element
- In between the closing tag of one element and the opening tag of the next element
- Automatic parsers will skip over whitespace. **An XML document should never be rejected because of the presence or lack of whitespace.**

Batching

Every HubXML message is structured with batching in mind, using a root element to wrap a collection of message elements within a single tag. The last element in a HubXML document will provide an audit count of messages that should be contained in the batch.

Special considerations apply for Functional Acknowledgement (FA) messages, as described below.

About the FA	Batching Consideration(s)
• Represents a handshake between producer and consumer of the business message(s) • Conveys an acknowledgement of receipt and an assessment of syntactical integrity • Establishes responsibility for the receiving party (i.e. the party generating an FA to indicate acceptance of an incoming message) to act on a message it has accepted	The batching structure defined in a schema determines the elements needed in a proper FA. For example: The <OrderMessageBatch> root element has an attribute of *batchNumber* Each <hubOrder> message element has an attribute of *transactionID* These attributes provide the unique identifiers that must be returned when building an FA response file.

Appendix

Important: *The information and examples below are for reference purposes only. The applications and methods listed are not endorsed or required by Rithum.*

New to XML?

XML is a specification for a generic data markup language. The standard specifies the rules for packaging and parsing data in XML format, but these rules say nothing about the business significance of the data. Business process, the purpose of specific messages in a business process, and the minimal data requirements of such messages are all outside the scope of the XML standard.

For more information, visit the World Wide Web Consortium (W3C) at http://www.w3.org/TR/xml11/. This site describes the standards that should be used across any XML message.

What is HubXML?

Our schemas define, in detail, how business messaging can be accomplished using XML, including:

- Data structure for a given business message
- Annotation of each individual element within the business message

XML Tools

In order to work with XML in the most efficient way possible, an XML toolset or XML viewing and/or editing application is necessary. Some examples:

- Freeware (a Google Search™ way)
- Proprietary OMS/Web Services applications (IBM WebSphere®, SAP®)
- Altova® (http://www.altova.com/xmlspy.html)