

DRAFT

NAXML Lottery Documents Implementation Guide

NAXML-LotDoc.DTD

The NAXML-LotDoc.DTD is an XML schema for the validation of three types of xml instance documents.

1. LotteryInvoice.xml – An XML instance document that is created by a state or provincial lottery authority and sent to a retailer. This document contains information for the retailer regarding the amount of money that will be drafted from the retailer’s bank account to settle lottery online and scratch transactions.
2. LotteryActivity.xml – An XML instance document that is created by a state or provincial lottery authority and sent to a retailer. This document contains information about lottery activity for the retailer, such as, scratch books sent, received, activated, settled, etc.
3. Lottery.xml – An XML instance document that is created by a state or provincial lottery authority and sent to a retailer. This document contains the information contained in both LotteryInvoice.xml and LotteryActivity.xml.

NAXML-DataDictionary.dtd

All NAXML dtDs use draw element and attribute declarations from a common data dictionary. This enables the amount and complexity of elements to be kept to a minimum yet still preserve uniqueness where necessary. In order to validate a lottery XML instance document NAXML-LotDoc.dtd and NAXML-DataDictionary.dtd must be in the same directory or the path of the NAXML-DataDictionary.dtd must be spelled out in NAXML-LotDoc.dtd. NAXML-LotDoc.dtd contains the complex elements, i.e. those whose content model contains simple elements.

Document Detail

The following information pertains to all lottery documents.

1. All NAXML XML instance documents use the following standard conventions:
 - a. Date and Time are in ISO 8601 format. Dates should be shown without hyphens, i.e. 20001201 v. 2000-12-01.
 - b. Country codes are in ISO format.
 - c. Language codes are in two digit ISO format, i.e. EN=English.
2. Every NAXML XML instance document contains a version number as an attribute of the root element. In the case of lottery documents the root element is

NAXML-LotDoc and the current version is 1.0. Any other value placed in here will not validate.

3. Every NAXML XML instance document contains a Transmission Header. This header is in lieu of using a particular transport, routing, and messaging protocol such as ebXML, SOAP, etc.. It will be used until there is industry acceptance of a standard protocol. The values for TransmissionId, TransmissionDate, TransmissionTime are left to the discretion of the lottery but the general use is:
 - a. To sequentially number the TransmissionId except on a “cancel” (see TransmissionStatus below).
 - b. To make the TransmissionDate and Time values equal to the creation date and time of the document.

TransmissionStatus is an empty element with an attribute of actionType whose acceptable values are: original – indicates an original transmission of the document, resend – indicates a retransmission of a previously transmitted document, and cancel – indicates that the previously transmitted document containing this TransmissionId should be cancelled.

4. The complex element Parties identifies both the Supplier and the Buyer. This is a generic complex element used in all NAXML XML instance documents. The only required element as per the dtd is “Name”. All other information is optional and empty elements should be omitted from the transmission. Nested within Parties is an additional complex element “Contact”. The use of this element and all of its children is optional. This element provides an opportunity to identify contact individuals within either the Supplier or the Buyer.

The following information pertains to the branch of the dtd that creates lotteryinvoice.xml.

1. Invoice is the parent of all additional information given in a LotteryInvoice.xml instance.
 - a. InvoiceNumber and InvoiceDate refer to this invoice.
 - b. SettlementDate refers to the date the retailer’s account will be drafted.
2. InvoiceDetail contains LineItem information and InvoiceSummary information.
3. LineItem contains a count attribute. The value of count should be incremented beginning with 1 each time LineItem begins anew. LineItem contains a complex element InvoiceUnit.
4. InvoiceUnit contains an attribute for gameType whose acceptable values are “Online” or “Scratch”. Because the retail value of each online ticket sold can vary depending upon the play, RetailPrice is an optional element.

- a. For “Online” InvoiceUnitQty has an attribute of lotteryUOMBasis whose value should be “GrossSales” and the value of the element should be “1”, InvoiceUnitCost should equal the total dollar amount of online tickets sold, and TicketCount should equal the number of individual online tickets sold. DateSettled indicates the date the tickets were sold as per the end of day lottery settlement.
 - b. For “Scratch” the InvoiceUnitQty has an attribute of lotteryUOMBasis whose value should be “Packs” or “Tickets” and the element value should be the number of Packs/Tickets being settled. The RetailUnitQty should reflect the individual tickets in the pack. The RetailPrice is equal to the retail selling price of the scratch ticket. The PackNumber element has an attribute of dateSettled. This should be the date the pack was settled per the procedures of the individual lottery.
5. InvoiceSummary contains a LineItems element with an attribute of count. This should be equal to the total number of line items in the InvoiceDetail. The TotalLineItemNetAmt has an attribute with acceptable values of “Debit” or “Credit”. The amount shown should equal the total of the individual LineItemNetAmt values shown in InvoiceDetail. AllowancesOrCharges which do not apply to the entire invoice and not individual line items are shown next. After applying all AllowancesOrCharges to the TotalLineItemNetAmt a TotalInvoiceDueAmt is derived.

The following information pertains to the branch of the dtd that creates lotteryactivity.xml.

1. GameActivity is the complex element that contains all of the information in the activity branch. The branch is constructed such that multiple locations within the Buyer may be identified.
 - a. Location contains the same elements as does Buyer with “Name” as the only element being mandatory.
 - b. ReportPeriod contains a date value which should be the closing date of the period being reported in this document. It is also possible to put a date range as a value. This would indicate the starting and ending dates of the period being reported.
2. ActivityDetail contains the complex element LineItem.
3. LineItem contains an attribute count whose value increments beginning with 1 each time the LineItem begins anew.

- a. GameDescription contains a gameType attribute whose acceptable values are “Online” or “Scratch”. The element value is free form text to describe the game by either number or name.
- b. GameId contains an identType attribute whose acceptable values are “Pack-BookNumber” or “Tickets” and an attribute of ident which can be used to indicate the Pack-BookNumber or Ticket number. The value of the element can be empty.
- c. Activity as a complex element contains ActivityStatus and ActivityDate. ActivityStatus contains a status attribute whose acceptable values are:
 - i. QuantityOrdered
 - ii. QuantityReturned
 - iii. QuantityInTransit
 - iv. QuantityStolen
 - v. QuantityDamaged
 - vi. QuantitySold
 - vii. QuantityReceived
 - viii. QuantityCommitted
 - ix. QuantityAvailableForSale
- d. ActivityDate contains a status attribute whose acceptable values are:
 - i. Ordered
 - ii. Issued
 - iii. Settlement
 - iv. Shipped
 - v. Received
 - vi. Returned
- e. ActivityDate also contains “method” and “referenceNumber” attributes that are optional can be used in conjunction with status. For example, status=”Shipped” method=”ViaUPS” referenceNumber=”123”.