WCIO WORKERS COMPENSATION

XML IMPLEMENTATION GUIDE

WORKERS COMPENSATION CRITICISM INFORMATION SPECIFICATIONS

(WCCRIT)

XML IMPLEMENTATION GUIDE

SECTION 1 - INTRODUCTION

A. Background / Overview

About the WCIO

The WCIO (Workers Compensation Insurance Organizations) is a voluntary association of statutorily authorized or licensed rating, advisory, or data service organizations that collect workers compensation insurance information in one or more states. The WCIO is composed of the managers of the various boards and jurisdictions. The purpose of the WCIO is to provide a forum for the exchange of information about workers' compensation insurance.

WCIO's XML Initiative

The WCIO XML schemas are based on the WCIO flat file products for WCPOLS, WCSTAT, WCRATING, WCNOA, and WCCRIT. The flat file reporting standards will continue to be maintained and utilized by WCIO members for data reporting. The XML products are an alternative standard that is available for use by the workers' compensation industry.
WCRATINGVALUES is only available in XML format.

B. Purpose and Use

This XML Implementation Guide includes specifications for the use of the WCCRIT message. This guide is not intended to be used in place of any instructional manual or rules. However, exceptions are noted within these specifications for your convenience.

Note that this guide is not, nor was it ever intended to be, a comprehensive guide to the reporting requirements of the individual state data collection organizations. The WCIO Data Specifications Manual, WCIO Data Dictionary and WCIO XML Schema are to be used in conjunction with this guide.

SECTION 2 - GENERAL TECHNICAL INFORMATION

A. What is XML?

The e**X**tensible **M**arkup **L**anguage (XML) is a simple, flexible, general-purpose markup language. (A markup language provides a way to combine text and extra information about that text, such as structure, layout, and so on.) It is classified as an extensible language because it allows you to define your own elements (elements are the basic structure for XML markup, and have two basic properties: attributes and content). XML's primary purpose is to facilitate the sharing of structured data across different information systems, particularly via the Internet.

XML is a fee-free open standard recommended by the World Wide Web Consortium (W3C www.w3.org; W3C Recommendation is the final stage of a ratification process of the W3C—it is the equivalent of a published standard in other industries). W3C's technical recommendation about XML specifies both the lexical grammar, and requirements for parsing data (the process of analyzing a sequence of tokens [blocks of text] to determine its grammatical structure with respect to a given formal grammar [precise description of an application language]).

XML started as a simplified subset of the Standard Generalized Markup Language (SGML), and is designed to be relatively human-legible. By adding semantic constraints, application languages can be implemented in XML.

1. Types of XML Documents

There are two types of XML documents: well-formed and valid. XML documents must adhere to the general rules of XML to ensure that all XML-aware software can read and understand the relative arrangement of information within the documents.

a.) Well-Formed Document
A well-formed document conforms to all of XML's syntax rules. For example, if a start-tag (such as <BodyText>) appears without a corresponding end-tag (</BodyText >), it is not well-formed.

b.) Valid Document
A valid document, beyond being well-formed, additionally conforms to some semantic rules. These rules are either user-defined or included as an XML schema (XSD). For example, if a document contains an undefined element, then it is not valid and a validating parser cannot process it.

The schema supplements the syntax rules with a set of constraints, typically restricting element and attribute names and their allowable containment hierarchies, such as only allowing an element, for example, named <effectiveDate> to contain one element named <year>, one element named <month> and one element named <day>, each of which has to contain only numeric character data.

The constraints in a schema may also include data type assignments that affect how information is processed. For example, the <month> element's character data may be defined as being a month according to a particular schema language's conventions, meaning that it must be formatted a particular way and must not be processed as if it were some other type of data.

An XML document that complies with a particular schema or DTD, in addition to being well-formed, is said to be valid.

B. XML Schemas

In general, a schema is a file that is used to describe the elements in an XML (eXtensible Markup Language) message or document. It is an abstract representation of characteristics and relationships in another XML document. The schema both specifies and validates that the content and order of the elements adheres to the content description. Schemas, also referred to as XSD's (XML Schema Definition), are a recommendation of the World Wide Web Consortium (W3C).

1. Schema Examples

The example below defines a tag name ReleaseDate. This tag is defined as having to conform to the format of a native schema type called “date”.

```
<xs:element name="ReleaseDate" type="xs:date"/>
```

The date is in the format of ISO 8601, and must appear in the XML document in the following format: YYYYMMDD

The other function of the schema is to show the order of elements, as well as the cardinality of those elements that occur within an aggregate. The following example shows the order of elements within an aggregate called ExposureTotalForAllClasses.

```
<xs:element name="ExposureTotalForAllClasses">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ExposureAmount" type=“xs:long” minOccurs=“0”/>
            <xs:element name="ExpectedLossTotal" type=“xs:long” minOccurs=“0”/>
            <xs:element name="ExpectedPrimaryLossAmount" type=“xs:long” minOccurs=“0”/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Within this aggregate, the following tags appear in this EXACT order:

- ExposureAmount (This is an optional element and native schema type “long”).
- ExpectedLossTotal (This is an optional element and native schema type “long”).
- ExpectedPrimaryLossAmount (This is an optional element and native schema type “long”).

2. Schema Validation

The process of checking to see if an XML document conforms to a schema is called validation.

XML documents are only considered valid if they satisfy the requirements of the schema with which they have been associated.

Schemas **will**:

- Validate the data format of a tag
- Validate tag order
- Validate code values

Schemas **will not** validate the contents of a tag.

XML Schema validations are performed using specialized parsers.

C. Use of Aggregates

Aggregates are used to group one or more XML elements. An aggregate can be referenced and utilized multiple times in a single XML message. An example of an aggregate would be the Name or Address.

D. Message Design and Use

1. Use of Code Lists

Code lists are used to ensure only valid values are used for certain XML elements. The list of elements that use code values can be found in **Section 6 - Code Lists**. These code lists are derived from the WCIO Data Specifications Manual.

2. System Rules for XML and Formats

a.) Include an element name in plural form to signify repeating records or entities.

b.) Name/Address and code lists will be global elements so they can be reused per schema.

c.) Element names should be taken from the product using proper case (all words are first-letter capitalized). For example, “CARRIER CODE” should be “CarrierCode”. The element names should be taken regardless of length. Commas, dashes, slashes, apostrophes should be removed from the element name. Any wording in parentheses, i.e. abbreviations, should be removed.

d.) New data elements for XML will follow the same naming convention as the flat file specifications.

e.) All elements/attributes are optional except ETR and link data fields that are required in the flat file specifications.

f.) A field labeled “numeric” in the flat file is translated to either int, long, or decimal data types in XML. Any field that has potential to be greater than 2,147,483,647 now or in the future is labeled as a long data type.

g.) A field labeled “alphanumeric” in the flat file is translated to a token data type in XML.

h.) A date field labeled as “numeric” in the flat file is translated to a date data type in XML.

Dates need to be translated using the list below:

XML DATA TYPE	FLAT FILE SPEC
Date	Year/Month/Day
dateTime	Year/Month/Day/Time
gDay	Day
gMonth	Month
gMonthDay	Month/Day
gYear	Year
gYearMonth	Year/Month

i.) Element name has been labeled with the word ‘Code’ at the end, even though the flat file field has a code list and the word ‘code’ is not part of the field name.

j.) Link Data is declared once and is not repeated on each Record Type.

3. Key / Reference Attributes

Many of the aggregates defined in the XML structure have a “key” attribute. The key attribute will have a unique value across all records for a given aggregate type. In the WCCRIT message, the Name aggregate will always have a key attribute. The key attribute is similar to the concept of a primary key for a database table. An example of a key attribute can be seen below:

```
<Name key="1">
    <OrganizationName>XYZ Company</OrganizationName>
</Name>
```

The key “1”, in this example, will be unique across all Name aggregate in a particular message. It is possible to reference a unique record from other XML structures because the key uniquely identifies a record. The key/reference attributes allow a data aggregate to be created once and referenced multiple times without duplicating the data.

The syntax for the references that are used in the WCCRIT message can be seen below:

⍰ nameReference=”UniqueValue”

4. Sample XML Structure

The following is an example of how name information is stored in the XML structure.

```
<Name nameReference="ID_1">
    <OrganizationName>XYZ Company</OrganizationName>
</Name>
```

The following is an example of how the Insured uses the nameReference to refer back to the Name key above.

```
<Insured nameReference="ID_1"/>
```

SECTION 3 - TECHNICAL STRUCTURE OF WCCRIT MESSAGE

A. General Message Structure

The WCCRIT XML file's main structure is composed of a submission header, transmittal information, the individual WCCRIT messages and file control record.

1. Submission Header

The XML file will always include a "submission wrapper" which will include one Wccrits aggregate node <Wccrits> containing one or more WCCRIT messages <Wccrit>, one Transmittal Information aggregate node <Transmittal> and one File Control Record aggregate node <FileControl>.

```
<xs:element name="Submission">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Transmittal"/>
            <xs:element name="Wccrits">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Wccrit" maxOccurs="unbounded">…</xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="FileControl">…</xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

2. Transmittal Information

The <Transmittal> aggregate node contains important information on the sender, receiver and file creation date. This section is the replacement for the flat file Electronic Transmittal Record (ETR). There will be only one <Transmittal> aggregate node per <Submission>.

3. Wccrits

The <Wccrits> aggregate node contains one or more individual <Wccrit> messages. Each <Wccrit> node contains the CRIT information.

```
<xs:element name="Wccrit" maxOccurs="unbounded">
    <xs:element ref="Name" minOccurs="0"/>
    <xs:element name="CarrierCode" type="xs:token"/>
    <xs:element name="PolicyNumberIdentifier" type="xs:token"/>
    <xs:element name="ExposureStateCode" type="xs:int" minOccurs="0"/>
    <xs:element name="PolicyEffectiveDate" type="xs:date"/>
    <xs:element name="ProductDataTypeCode" type="ProductDataTypeCode_Type"/>
    <xs:element name="ReportLevelCodeReportNumber" ……… />
    <xs:element name="TransactionCode" type="TransactionCode_Type" minOccurs="0"/>
    <xs:element name="CorrectionSequenceNumberCode" ……… />
    <xs:element name="TransactionIssueDate" type="xs:date" minOccurs="0"/>
    ……………………………….
    <xs:element name="WcstatErrorsInformation" minOccurs="0">
    <xs:element name="WcpolsErrorsInformation" minOccurs="0">
    <xs:element name="InsurerAddress" minOccurs="0">
    <xs:element name="SubsequentNotification" minOccurs="0">
```

4. FileControl

The <FileControl> aggregate node contains the Criticism Total in this Submission.

B. List of Aggregates and Elements for WCCRIT

The following section contains a detailed listing of the WCCRIT XML aggregates and tag elements in the order that they will appear within the submission header <Submission> of the XML file.

1. Transmittal

The <Transmittal> contains important information on the sender, receiver and file creation date and contains these elements:

DataProviderContactEMailAddress
DataTypeCode
DataReceiverCode
TransmissionVersionIdentifier
SubmissionTypeCode
SubmissionReplacementIdentifier
DataProviderCode
NameOfDataProviderContact
ElectronicOrPaperReceiptCode
PhoneNumber
PhoneNumberExtension
FaxNumber
ProcessedDate
AddressOfContactStreet
AddressOfContactCity
AddressOfContactState
AddressOfContactZipCode
DataProviderTypeCode
ThirdPartyEntityFederalEmployerIdentificationNumber

2. Wccrits

The <Wccrits> contains one or more individual <Wccrit> messages.

- *Wccrits*
  - *Wccrit*
    - *Name*
      - *OrganizationName*
    - *CarrierCode*
    - *PolicyNumberIdentifier*
    - *ExposureStateCode*
    - *PolicyEffectiveDate*
    - *ProductDataTypeCode*
    - *ReportLevelCodeReportNumber*
    - *TransactionCode*
    - *CorrectionSequenceNumberCode*
    - *TransactionIssueDate*
    - *PolicyExpirationOrCancellationDate*
    - *Insured*
    - *ErrorStatusCode*

```
ProcessedDate
RespondByDate
NameOfDcoContact
DcoContactPhoneNumber
DcoContactEmailAddress
NextFineAmount
ReportedPolicyEffectiveDate
IndependentDcoRiskIdNumberFileNumberAccountNumber
InternalSubmissionId
LetterId
LetterIssuedDate
ThirdPartyEntityFederalEmployerIdentificationNumber
BusinessSegmentIdentifier
CorrectionTypeCode
ReplacementReportCode
WcstatErrorsInformation
    WcstatErrorInformation
        RecordTypeSequenceNumber
        ErrorCode
        MessageTypeCode
        ErrorCodeTolerance
        ErrorMessage
        MultipleRecordIdentifiers
        ErrorMessageSequenceNumber
WcpolsErrorsInformation
    WcpolsErrorInformation
        RecordTypeSequenceNumber
        ErrorCode
        MessageTypeCode
        ErrorCodeTolerance
        ErrorMessage
        MultipleRecordIdentifiers
        ErrorMessageSequenceNumber
InsurerAddress
    NameOfDataProviderContact
    Title
    NameOfDataProviderCompany
    AddressOfContactStreet
    AddressOfContactCity
    AddressOfContactState
    AddressOfContactZipCode
SubsequentNotification
    NotificationNumber
    FineNotificationNumber
    FineAmount
    NotificationTypeCode
```

3. FileControl

The <FileControl> aggregate node contains the Criticism Total in this Submission.

```
FileControl
    CriticismTotal
```

SECTION 4 - GLOSSARY

A list of acronyms, abbreviations, and terms used in this Implementation Guide is given below.

Term	Definition
Aggregate	Aggregates are used to contain one or more XML elements.
DCO	Data Collection Organization
DTD	Document Type Definition. An XML schema that is native to XML. A DTD is primarily used for the expression of an XML schema through a set of declarations that conform to a particular markup syntax and that describe a class, or type, of XML document.
element	The basic structure for XML markup, which has two basic properties: attributes and content.
extensible language	A programming language that allows you to define your own elements.
markup language	A markup language provides a way to combine text and extra information about that text, such as structure, layout, and so on.
message	A collection of data fields sent or received together between software applications. A message contains a header (which stores control information about the message) and a payload (the actual content of message).
namespace	A namespace is an abstract container or environment created to hold a logical grouping of unique identifiers (that is, names).
schema	*See*: XML schema.
schema validation	The process of checking to see if an XML document conforms to a schema.
Tag	A markup construct that begins with "<" and ends with ">". Tags come in three flavors: *start-tags*, for example `<section>`, *end-tags*, for example `</section>`, and *empty-element tags*, for example `<line-break/>`.
valid XML document	A valid document, beyond being well-formed, additionally conforms to some semantic rules.
W3C	World Wide Web Consortium; website: www.w3.org
WCIO	Workers Compensation Insurance Organization
well-formed XML document	A well-formed document conforms to all of XML's syntax rules.
XML	eXtensible Markup Language. A fee-free open standard recommended by the World Wide Web Consortium that is a simple, flexible, general-purpose markup language. XML's primary purpose is to facilitate the sharing of structured data across different information systems, particularly via the Internet.
XML document	An electronic file that contains XML code and text. There are two levels of correctness in an XML document: well-formed, and valid.
XML schema	An XML schema is the description of a type of XML document that is used for document validation.
XML syntax	The rules that govern the structure of XML. XML syntax determines the structure and content of an XML document.

SECTION 5 – CODE LISTS

WCIO Code Lists - Code lists are derived from the WCIO Data Specifications Manual. For assistance in determining the valid code values please refer to the appropriate data specification manual posted on the WCIO website at www.wcio.org.

SECTION 6 - XML DEVIATIONS COMPARED TO FLAT FILE WCCRIT DATA SPECIFICATION

A. General Deviations from the Flat File

1. Certain record types have a different structure in the XML schema (i.e. Name/Address). Name/Address contains an attribute with the data type(ID/IDREF) for linking purposes.

```
<Name nameReference="ID_1">
 <OrganizationName>WCRIBMA</OrganizationN
ame>
</Name>
<Address addressReference="ID_2">
 <Street1>101 Arch Street</Street1>
 <Street2>PO Box 111</Street2>
 <City>Boston</City>
 <State>MA</State>
 <ZipCode>02110</ZipCode>
</Address>
```

Note: *The tag name OrganizationName was chosen so that it could be used globally across products.*

2. Address string is converted into structured format. It is split into Street1, Street2, City, State and ZipCode nodes.

3. DETAIL RECORD COUNT TOTAL field is not included in the schema because the concept of the flat file record types does not exist in the XML schema.

B. WCCRIT Specific Deviations from the Flat File

The following section contains a detailed listing of the specific deviations between the XML file and the WCIO WCCRIT Data Specification.

1. Additions / Modifications

(None)