

2.0 Design Guidelines

IMPORTANT: The Chemical Industry Data Exchange ("CIDX") is a nonprofit corporation, incorporated in the State of New Jersey, which is exempt from federal taxation under Section 501(c)(6) of the Internal Revenue Code. The Chem eStandards and other publications of CIDX have been developed in furtherance of CIDX's nonprofit and tax-exempt purposes. CIDX has taken reasonable measures to develop the Chem eStandards and other CIDX publications in a fair, reasonable, open, unbiased, and objective manner for the purpose of enhancing the ease, speed and cost aspects of securely conducting business electronically in the Chemical Sector. The Chem eStandards and other CIDX publications are developed in compliance with all applicable laws, including antitrust laws, and pursuant to the CIDX Antitrust and Intellectual Property Policies ensuring open access and use, while at the same time preserving contributing companies' individual rights to their own intellectual property. The Chem eStandards may be copied and distributed in the form published and copyrighted in accordance with the license agreed to upon download, but no permission is granted for any change or modification to the Chem eStandards.

The nature of appropriate practices or guidance is likely to change over time and with developments in technology. CIDX cannot guarantee the accuracy, completeness, efficacy, timeliness, or correct sequencing of the Chem eStandards or other publications. Use of the Chem eStandards and any CIDX guidance document or other publication is voluntary, and reliance on them should be undertaken only after an independent review by the user. Inclusion of material in the Chem eStandards or other publications does not constitute a guarantee, warranty, or endorsement by CIDX regarding any guidance, methodologies, or preferences for conducting business, implementing any CIDX standards, or enhancing computer security. Further, neither CIDX nor its officers, directors, members, employees, or agents shall be liable for any loss, damage, or claim with respect to any such documents, work, or services; all such liabilities, including direct, special, indirect, or consequential damages, are expressly disclaimed. Information provided in the Chem eStandards is "as is" without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or freedom from infringement. The Chem eStandards and related trademarks are the property of CIDX. Copyright 2004 Chemical Industry Data Exchange. All rights reserved.

DESIGN GUIDELINES TABLE OF CONTENTS

| | |
|---|-----------|
| 2.1 Introduction and Scope of Effort | 5 |
| 2.1.1 Universal Business Language Naming and Design Rules | 5 |
| 2.1.2 Design Guidelines | 5 |
| 2.1.3 Scope of Effort..... | 5 |
| 2.2 Chem eStandards Message Guidelines | 6 |
| 2.2.1 Message Definition | 6 |
| 2.2.2 Message Naming Conventions..... | 7 |
| <i>All Messages</i> | 7 |
| <i>Reply messages</i> | 7 |
| 2.2.3 Message Structure | 8 |
| 2.3 Chem eStandards Structure Guidelines | 9 |
| 2.3.1 Structure Definition..... | 9 |
| 2.3.2 Partner Definition | 12 |
| 2.4 Chem eStandards Data Element Guidelines | 13 |
| 2.4.1 Data Element Names | 13 |
| 2.4.2 Data Element Definitions..... | 14 |
| 2.4.3 Data Types | 15 |
| 2.4.4 Data Length | 16 |
| 2.4.5 Code List Naming..... | 16 |
| 2.4.6 Code List | 16 |
| 2.4.7 Sample Data | 16 |
| 2.4.8 Synonyms | 17 |
| 2.4.9 Comments..... | 17 |
| 2.4.10 Attributes..... | 17 |

2.1 Introduction and Scope of Effort

Note: Throughout this document *schema* generally refers to *W3C XML Schema*.

This section provides guidelines for creating Chem eStandards schemas.

2.1.1 Universal Business Language Naming and Design Rules

CIDX used the OASIS Technical Committee Universal Business Language's Naming and Design Rules (UBLNDR) as general guidance in designing the Chem eStandards v4.0 schema architecture. At the time the Chem eStandards v4.0 draft schemas were made available for public review, the UBLNDR was not finalized. As a result, there are no guarantees that the Chem eStandards v4.0 conform to any given naming or design rule. Additionally, the UBLNDR conforms to the UN/CEFACT Core Components Technical Specification v2.0 and make use of multiple namespaces—two architectural characteristics that CIDX was not ready to implement in Chem eStandards v4.0.

2.1.2 Design Guidelines

The following topics are discussed in this section:

| | |
|--------------------------------|--|
| Message Guidelines | <p>These guidelines cover the definition, structure, and naming conventions of Chem eStandards messages.</p> <p>These guidelines are primarily intended for those who help develop or maintain CIDX messages.</p> |
| Structure Guidelines | <p>These guidelines cover the definition of data structures (data elements that contains child elements), in general, and also describe the format and the proper use of the Partner Information structure.</p> <p>These guidelines are primarily intended for those who help develop or maintain CIDX messages.</p> |
| Data Element Guidelines | <p>These guidelines cover the definition, documentation, naming, and use of data elements and attributes.</p> <p>These guidelines are primarily intended for those who help develop or maintain CIDX messages.</p> <p>These guidelines are also intended for anyone wishing to implement the Chem eStandards messages, as adherence to these guidelines is a prerequisite to compliance.</p> |

2.1.3 Scope of Effort

Chem standards messages are defined and maintained independent of the potential constraints for business partners implementing these standards with regard to the use of bandwidth or processing efficiency.

2.2 Chem eStandards Message Guidelines

The guidelines described in this section have evolved with the creation and maintenance of the Chem eStandards schemas through Version 4.0 and are future Chem eStandards schema development.

2.2.1 Message Definition

Individual messages should be created for every business transaction, rather than using the same message for multiple purposes. There are two facets of this guiding principle:

- 1) An individual message supports only one business function. The message should not have more than one functional or business meaning based on the value of a code contained within the message structure (e.g., a separate message should exist for creating an order and changing an order).
- 2) Data values within individual messages that support a single business function can modify the nature of the transaction or data exchange being performed (e.g., an OrderCreate message supports a regular order as well as a blanket order, based on data values contained within the message).

The key advantages and disadvantages of this approach are summarized in the following table:

| Advantages | Disadvantages |
|---|--|
| <ul style="list-style-type: none"> • More clearly defined use of Chem eStandards messages. • Reduces the number of optional fields in given message, which reduces the amount of negotiations involved in trying to apply the Chem eStandards properly. • Ease of implementing specific solutions. • Message independence leads to greater flexibility for modifications over time. | <ul style="list-style-type: none"> • A large number of message models must be created. However, given an appropriate set of guidelines/principles, this is considered a controllable issue. |

Note: Exceptions to the design guidelines, as defined in this document, do exist within the Chem eStandards, based on the design teams' understanding of the requirements within the chemical industry. The Chem eStandards design teams had the freedom to propose designs that were appropriate for their defined business processes and messages. For example, the Order Team (from the Phase 1 effort) agreed to create separate messages for creating and changing an order, while the Catalog Team utilized a single message to create, update, and delete product and customer specific catalog information.

2.2.2 Message Naming Conventions

The following guidelines are used in naming Chem eStandards messages:

All Messages

- Message names should be created by combining two or more words.
- Each message name should start with a noun (rather than a verb).
- Each message name should be descriptive.
- Names should be spelled out completely. Abbreviations or acronyms should not be used.
- The first letter of each word should be capitalized.
- Numbers, spaces, underscores, dashes or other special characters should not be used within message names.

Note: To enhance readability, message names may be found to be represented with spaces between words in various forms of documentation and diagrams, but would never contain spaces in a machine-readable environment, such as SCHEMAS or sample XML messages.

| Not Used | Used |
|--|--|
| <ul style="list-style-type: none"> • CreateOrder • ChangeOrder • AcceptPosting • RequestShipmentStatus • RequestPayment | <ul style="list-style-type: none"> • OrderCreate • OrderChange • PostingAccept • ShipmentStatusRequest • PaymentRequest |

Reply messages

The Chem eStandards publication discusses two types of message responses: technical responses and transaction or business responses.

Technical responses are returned to the source system from the destination system upon the successful delivery of a message and its syntactical validation, commonly referred to as RA (Receipt Acknowledgement) or RAE (Receipt Acknowledgement Exception). Business partner software systems must follow the specifications defined and documented in the Envelope and Security publication of the Chem eStandards, and business partners are responsible for ensuring a technical response is returned for all received messages.

Transaction responses are unique messages used as part of a two-way business dialog as a reply or response to some requested action. In these cases, the reply message should have the word 'Response' appended to it. For example, a Payment message would typically require a PaymentResponse message be returned to the sending business partner.

2.2.3 Message Structure

- 1) Chem eStandards messages shall be represented as SCHEMAS.

Each Chem eStandards message is an XML document conforming to a specific Document Type Definition (SCHEMA) for the transaction or data exchange being executed. An XML specification describes structured data. Within an XML document, data elements may either contain other data elements or data. In addition, Chem eStandards message data elements may contain attributes that describe the data within the data element.

Note: An XML Schema format was approved by the World-Wide-Web Consortium (W3C) immediately prior to the release of v2.0.1 of Chem eStandards in May 2001. In the future, CIDX may choose to replace the Chem eStandards SCHEMAS with an equivalent set of XML Schemas.

- 2) Chem eStandards messages shall conform to a header/body structure.

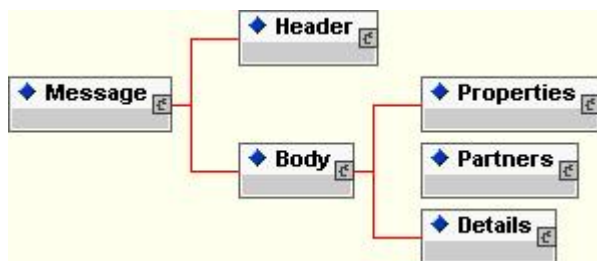


Figure 2.2.3.a: Data model of Chem eStandards message structure

Each Chem eStandards message payload (business content) begins with a root element with the same name as the message itself. For example, the root element of the OrderCreate message is named OrderCreate.

Below this root element, every message contains a Header and a Body structure. The Header structure is common across all Chem eStandards messages and contains information on the disposition and identification of the message itself (e.g., document identifiers, document date and time stamps, etc.), but does not contain specifics on the data exchanged.

The Body structure, which contains the message-specific data, may have any/all of the following three elements: Properties, Partners, and Details. The inclusion of each of these structures in a message design is optional. These structures provide data and information to be used in the data exchange.

The Properties structure contains data that applies to the entire message, Partners structure contains information about the business partners involved in the data exchange, and Details structure contains the line item level information. In each of the Chem eStandards messages, these structure names are prefixed with the name of the message in which they occur. These elements are functionally the same, but have been given unique names to ensure their uniqueness, and to allow for their inclusion and management in data and object modeling software packages.

Use of a common header and a body message structure was evaluated prior to selecting this particular approach for Chem eStandards messages. For example, other XML specifications

that have chosen to use a header and body structure include RosettaNet, cXML (Ariba), and xCBL (CommerceOne). Many other data exchange specifications also include a header/body structure of one form or another.

- 3) A data element specified at the Properties or Partners level shall pertain to the entire message, unless that specific data element is specified and found with a different value in the Details section; in which case the value in the Details level overrides the previous value for that structure only.
- 4) Data models are the hierarchical structures of information that will be exchanged by business partners using the Chem eStandards and are represented in a schema.
- 5) A unique data model or SCHEMA should be created for each message.
- 6) Messages that are logically different should have their own data model, even if the message information is identical.
- 7) Some data exchanges only require a single message to be considered complete (i.e., a one-way communication). These types of messages do not require a business reply of any sort. Examples of one-way communication include the PaymentDetail, RequestForQuote, and ProductCatalogUpdate messages.

2.3 Chem eStandards Structure Guidelines

Structures are aggregate data elements that contain child elements and are used as the building blocks of the Chem eStandards messages. The following are guidelines for creating Chem eStandards message structures:

2.3.1 Structure Definition

- 1) Existing structure definitions should be re-used wherever possible. Data designs already exist for many of the Chem eStandards messages and/or structures within the messages.

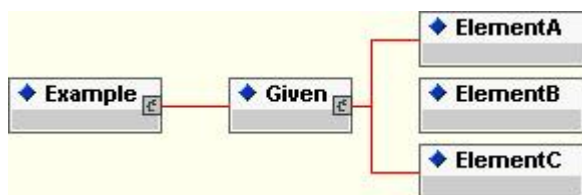


Figure 2.3.1.a: Example of an existing message structure

- 2) Existing designs from other XML initiatives and sources should also be consulted before inventing new designs, such as the following:
 - RosettaNet – our most leveraged external XML standard
 - ebXML, xCBL, etc. – other external XML standards
 - OASIS, OAG – other prime external XML resources
- 3) Existing Chem eStandards message models and the Data Dictionary must always be checked to ensure that a duplicate of an existing structure or data element is not being created. When there is not an exact match that meets the needs of a new message for a structure, an attempt

should be made to re-use applicable pieces to build the necessary components of a new structure.

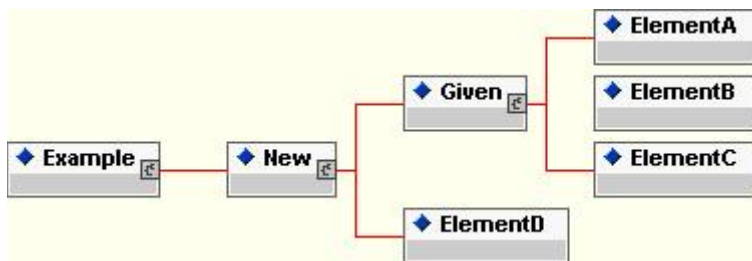


Figure 2.31.b: Example of the creation of a new message structure

- 4) To ensure that Chem eStandards messages are consistent across different segments of the industry, the use of any given structure is identical in all messages. The addition or omission of elements, attributes, or other child structures is not permitted.

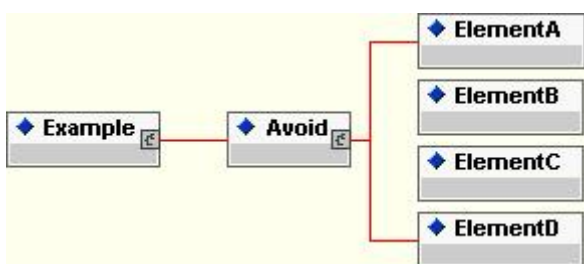


Figure 2.31.c: Example of the type of new message creation that should be avoided

- 5) Related data items should be grouped together within a container structure using self-explanatory data elements.

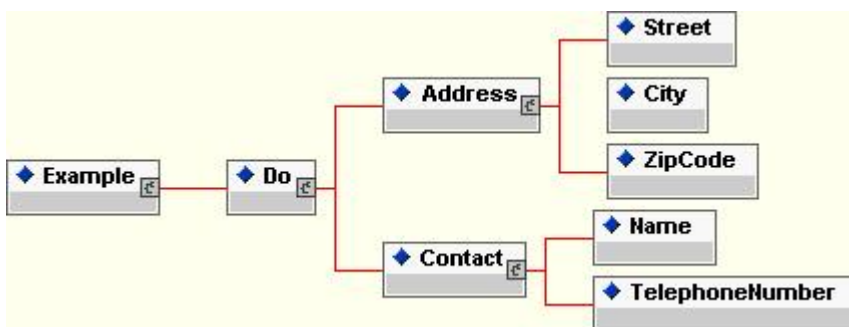


Figure 2.31.d: Example of the creation of a named message structure

- 6) The creation of nameless structures should not be allowed.

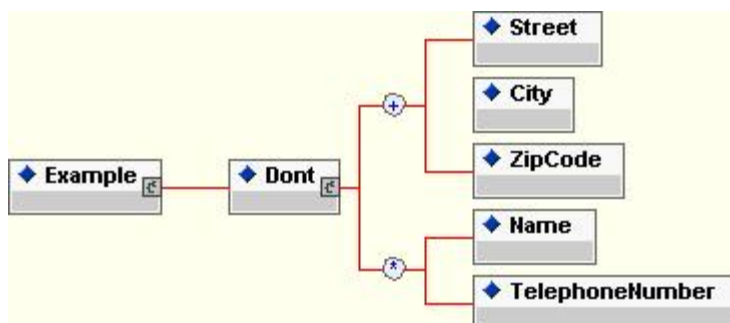


Figure 2.3.1.e: Example of a message structure that should be avoided

- 7) When a structure is defined, at least one child data element should be required (mandatory).

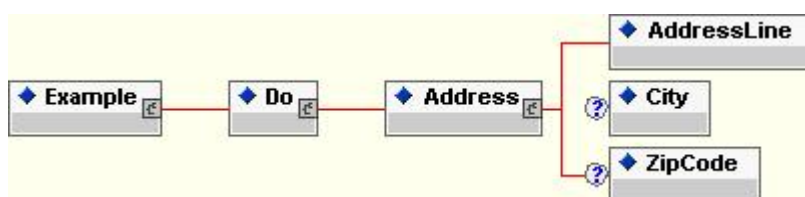


Figure 2.3.1.f: Example of a structure with at least one mandatory element

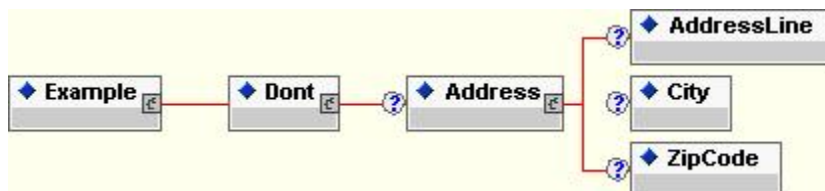


Figure 2.3.1.g: Example of a structure with no mandatory elements (all optional)

- 8) When the use of data elements is required in a child structure that is nested within a parent structure that contains the same data elements, the nested element should be defined as a sub-component of the parent. In other words, recursive structures should be avoided.



Figure 2.3.1.h: Example of a nested sub-component structure

Note: The exception to this rule is the ShipmentPackaging structure, in which the recursive structure ShipmentPackaging has been included to allow for multiple nested levels of packaging of goods. For example, a shipment could consist of a pallet that contains multiple cartons, which are packed with several boxes, that contain cylinders inside the boxes. The recursive structure was intentionally designed, as opposed to creating named layers, e.g.: Layer1, Layer 2, etc., to provide the flexibility needed to define many different types of packaging schemes, and to allow the number of levels to be designated as needed.

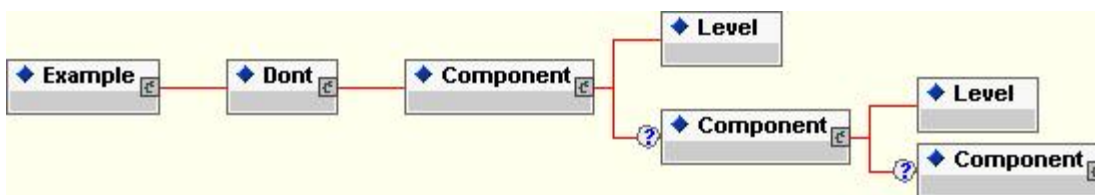


Figure 2.3.1.h: Example of a recursive component structure

- 9) When a new structure is introduced, new data elements should be evaluated using several criteria:
- Can the requirement be fulfilled with an existing data element?
 - Can an existing data element be modified to meet the need?
 - What additional information is needed from the design team to ensure a completely defined new data element or structure?

The Message Development Committee is ultimately responsible for approving the Chem eStandards Data Dictionary with guidance from the Repository Manager, through a process of technical assessment that includes the involvement and approval of all technical committees.

2.3.2 Partner Definition

- 1) A single structure for PartnerInformation has been defined for Chem eStandards messages, with the mandatory/ non-mandatory elements, cardinality of elements, etc. defined once within this structure and implemented across all messages.
- 2) The PartnerInformation structure should be the only structure used for the definition of any partner.
- 3) Chem eStandards messages include two types of partners; mandatory and optional.
- 4) Partners that are mandatory within a message should be explicitly defined in the Partners section of the SCHEMA.
- 5) Partners that are optional within a message should be listed in the documentation of each message and should use the OtherPartner structure.
- 6) The function of the optional partner is indicated by the PartnerRole attribute on the OtherPartner structure.
- 7) The OtherPartner structure should always be defined as 0 to many.
- 8) Any additional information required for a particular partner should be defined separately from the partner structure.

2.4 Chem eStandards Data Element Guidelines

The following guidelines should be followed when creating and defining data elements needed for Chem eStandards messages:

- All message data elements are part of a common Data Dictionary. While some data elements may be used only once, there are no elements that are private to a particular message.
- A complete data element description (or documentation) shall consist of:

NAME
DEFINITION
DATA TYPE
DATA LENGTH
CODE LIST NAME
CODE LIST
SAMPLE DATA
SYNONYMS
COMMENTS
ATTRIBUTES

2.4.1 Data Element Names

- 1) Several words have been generally 'reserved' for specific uses within data element names and their use should be avoided elsewhere:
 - Information – used within a structure name to group identification-type elements (e.g. PartnerInformation). Equates to relatively static data (master file data).
 - Identifier – used within an element name to indicate a key reference by which the element structure is known (e.g. ProductIdentifier uniquely identifies a product).
 - Code – used within an element name to indicate an externally defined code list (e.g. CountryCode defined by ISO 3166).
 - Date – used within an element name to indicate a field containing a date and/or time (e.g. ToDateTime).
 - Flag – used within an element name to indicate a data element that provides an either/or choice, such as that offered by a Boolean data type.
- 2) Data element names should consist of an adjective and a noun (e.g., PartnerInformation, PaymentTerms), unless the use of one or several words is more appropriate.
- 3) When a match exists between a RosettaNet message and the Chem eStandards, the use or adoption of RosettaNet element names and data structures should be evaluated.
- 4) When new data elements need to be defined, the RosettaNet naming conventions should be followed:
 - Element names comprise one to several words.
 - Conflicts with data elements and structures in other messages having the same name but other subsequent data elements are prohibited.

- 5) Data element names should be descriptive and as specific as possible.
- 6) The use of abbreviations or acronyms as data element names should be avoided. There are exceptions of course, particularly where the abbreviation or acronym is much better known than the term it represents (e.g. EDI, VAT, etc.).
- 7) Numbers (e.g., AddressLine1, AddressLine2), underscores, spaces, dashes or other special characters should not be used within message names unless the number is intrinsic to the name (e.g.: OT55Flag).
- 8) The first letter of each word should be capitalized.
- 9) The singular should be used for a structure with non-repeating subsequent data elements, even if multiple instances of that structure (e.g., OrderCreateProductLineItem) are needed.



Figure 2.41.a: Example of singular data element name (OrderCreateProductLineItem)

- 10) If a structure with single cardinality contains multiple data elements with identical structures behind it, the plural (e.g., OrderCreatePartners) should then be used.

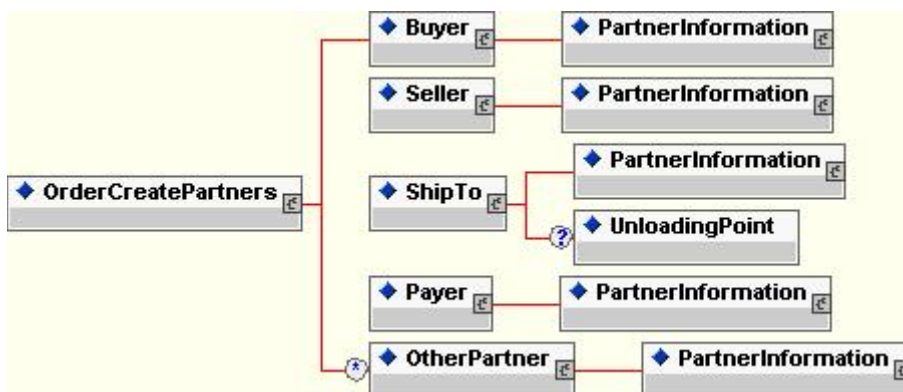


Figure 2.41.b: Example of a plural data element name (OrderCreatePartners)

2.4.2 Data Element Definitions

- 1) Data element descriptions should be completed as fully as possible. They should be more than simply a repeat of the element name.
- 2) Definitions should be one to several sentences describing the use/purpose, contents, and source of data for a data element. They should contain all the information necessary to uniquely define the element, describe its purpose and use, include sample and allowable values with a view to removing any ambiguity in the interpretation of meaning/use/format of the element.
- 3) Data element definitions should be clear enough to allow for data element re-use.

2.4.3 Data Types

Data elements must be described as one of the following data types:

1) String

- Anything intended to be considered a string, free format, all characters possible.
- This data type is also to be used for a key field or a field containing numeric data if it is to be interpreted exactly as entered; e.g. a telephone number, purchase order number with/without punctuation or leading zeros.
- DATALENGTH should specify the exact number or minimum and maximum number of characters including spaces and punctuation (e.g. 10 or 3,40; default is 1,255).

2) Boolean

- The only allowable values are “true”, or “false”.
- DATALENGTH must be 4, 5.

3) Integer

- To be used for fields containing whole number values only, with or without a sign value but with no other punctuation. Spaces or characters used as thousands etc. separators **must not** be used.
- Leading zeros should not be entered.
- Sign values (+ or -) should immediately precede the integer value.
- Absence of a sign indicates positive value.
- DATALENGTH should specify the exact number or minimum and maximum number of digits including sign value.

4) Real

- To be used for fields containing quantity or amount values (where these are not always whole numbers).
- Data should be entered as floating point numeric values.
- These fields would contain a sign value (if appropriate), a decimal symbol (to separate the integer portion from the decimal portion if not a whole number) but no other punctuation. Spaces or characters used as thousands etc. separators **must not** be used.
- Leading zeros should be omitted unless the value is less than 1. In this case a significant digit (0) should always be specified (e.g.: 0.15).
- Any sign (+ or -) should immediately precede the value.
- Absence of a sign indicates positive value.
- The “.” symbol should normally be used as the decimal symbol unless otherwise agreed between trading partners. The “.” symbol should always be used as the decimal symbol if the symbol in general use differs between sender and receiver.
- DATALENGTH should specify the minimum and maximum field length including sign and decimal symbol values. Default is 1,17.

5) Dates and Time-Related Data Type

Date and time-related data types shall conform to the applicable built-in schema data type. Additionally, all DateTime datatypes shall include time zone information.

2.4.4 Data Length

Data length should specify the exact number or minimum and maximum number of characters including spaces and punctuation (e.g. 10 or 3,40; default is 1,255).

2.4.5 Code List Naming

- 1) Where appropriate, data elements are specified to have values based on industry standard code sources (e.g., ISO 3166 or ANSI ASC X12 335). The definitions and code sources of all elements should be re-used when elements have been borrowed from existing standards. In addition, all data elements that reference an external code source should append the word Code at the end of the element (e.g., LanguageCode, LocationCode).
- 2) Data elements that reference an external code list contain the attribute Domain, which identifies the code source in the value(s) of the attribute. While most Code elements have only one source, some elements (e.g.: LocationCode) provide a choice between code lists in the enumerated list of values.
- 3) External code lists should be named in a consistent way, following the format shown below (where spaces are replaced with -):
 - ISO-3166
 - ANSI-ASC-X12-353
 - UN-Rec-20
- 4) Within the element description, however, the code list should be given its full name (e.g. UN/CEFACT Recommendation 20), with clarification of which code within the code list is to be used if there is more than one (e.g. Common Code).

2.4.6 Code List

Any data shown in the sample data section of the element description should follow these guidelines:

- 1) The code source shall be specifically identified within the Data Dictionary. The organization, the standard, and the item in the standard shall be identified where possible (e.g., ANSI-ASC-X12-335).
- 2) If there is a need to create a custom code list, (i.e., not supported by an industry standard code source) the proposal should be compared to the code lists in the current version of the repository. A fixed list of custom code values for that particular data element must be provided in the description, together with an explanation of each value's use/meaning.

2.4.7 Sample Data

Any data shown in the sample data section of the element description should follow these guidelines:

- 1) Each line of the SAMPLEDATA field should contain a single value, representing a valid data entry.

- 2) If more than one value is given, each value should be shown on a separate line. Multiple values should only be specified where these are required to show the full range of possible valid formats.
- 3) Each value listed under SAMPLEDATA could be copied and pasted into the associated element or attribute and would conform to both the specification defined by the SCHEMA and all other CIDX defined data requirements.
- 4) If a more verbose explanation is needed to clarify the types of values that can be used, this information should be included in the COMMENTS field of the element or attribute definition.

2.4.8 Synonyms

Some data elements will list other commonly recognized names for a data element, however this feature is not widely used in the Chem eStandards.

2.4.9 Comments

The COMMENTS section of the element description should be used to convey any pertinent information on the element not given elsewhere, e.g. clarification of codes, meanings of attributes, examples of use etc.

2.4.10 Attributes

- 1) In general, attributes are shorter strings than data elements (usually a single word value). When designating the value of an attribute, one should be aware that information stored in attributes can only contain 255 characters.
- 2) Attributes must always be defined as required values.
- 3) There is no absolute rule on when to encode information as data element contents or as attributes, but the following guidelines will help in decision making:
 - Chem eStandards models data in the style of RosettaNet, which significantly limits the use of attributes.
 - An attribute is used to supply “metadata” or information that modifies or further explains a data element, such as PartnerRole (which may be Buyer, Marketplace, Shipper, etc.) for a Partner data element, or InstructionType (which may be ShippingInstructions, or PaymentInstructions, etc.) for a SpecialInstruction data element, or Version (which may be 2.0.2, 3.0) for the message.
 - Processing or data handling: Attributes are used to assist in the disposition or handling of data elements such as Action (which may be Add, Replace or Delete) for a ProductCatalogUpdateProductLineItem element.
 - Persistence: Data elements are for storing information kept by business partners, attributes are to help information get from the source to the destination.
- 4) A definition for the use of each attribute in the list must be provided to the Repository Manager for inclusion in the Data Dictionary.
- 5) Attributes should follow the same naming convention as data elements.

- 6) Certain attribute names are reserved for specific uses, such as the Domain attribute, which is used to indicate the source of an external code list, and the Agency attribute, which is used to indicate the organization that has supplied an identifying number (e.g.: PartnerIdentifier, or ProductIdentifier).