AEX cfiXML LT Specifications

14 December 2011

Version 1.1

MAK Turton, Alar Engineering Software Inc.; RW Turton, Alar Engineering Software Inc.; T Dahl, Intelliquip; S Frees, Intelliquip; C Cragg, EPRI; K Chen, NIST; ME Palmer, NIST.

Executive Summary

The cfiXML Light (LT) schema version has been developed to support the use of simple XML serialization tools and so enable developers with limited XML experience to be able to develop data exchange systems easily and cost-effectively.

The purpose of this document is to:

- aid application developers new to cfiXML in making the decision on whether full or LT schema will meet their requirements;
- detail specifications for the use of cfiXML Light (LT) schema;
- provide recommendations on testing criteria to be met to confirm a successful data exchange system has been implemented;
- outline the process to create LT and Full versions of cfiXML and the LT-tool functionality.

The authors consist of a team of programmers recently introduced to cfiXML, cfiXML developers and technical advisors and specialists.

References

Information on cfiXML is available from:

https://sourceforge.net/projects/cfidev/

and http://www.cfixml.org/ which includes detailed guidelines for full cfiXML at

http://www.cfixml.org/documents/AEX%20XMLGuide%202010-08-12.pdf

HI – www.pumps.org

Alar – www.alarsoftware.com

Intelliquip – www.intelliquip.com

EPRI – www.epri.com

NIST - www.nist.gov

AEX Project - http://fiatech.org/aex.html

GVCC Project - http://fiatech.org/active-projects/53-global-valve-cross-reference-e-catalog-gvcc-roadmap-element-3.html

CONTENTS

1 Introduction

The FIATECH Automating Equipment Information Exchange (AEX) Project was launched in 2002 with the objective of automating equipment design and delivery through software interoperability.

1.1 AEX capital facilities industry XML

The AEX capital facilities industry XML (cfiXML) model developed to be capable of handling any equipment item, engineering document and material property for any information exchange usage scenario across the entire facilities life cycle.

The goal of XML is to be application-independent. Being text-based ensures that most software applications should be able to be defined in terms of XML. Additionally, text files will always be readable, so given the decades involved in a facility life cycle, data will always be accessible.

The cfiXML model is a layered structure to ensure that there is a common consistent foundation, and maximum potential for the reuse of items for efficient schema development. Another key part of the design of cfiXML schemas is the built-in flexibility provided by relatively simple methods to extend schema files. Additional features of the schema provide support for version management so that records can be kept throughout the life cycle of the equipment item, and a means to record any changes to data items (change tracking). The cfiXML schemas also include units of measurement which enables the required units to be set for any defined physical property throughout the schema. Details of the development and structure of cfiXML are provided in the guidelines.

The cfiXML schemas have been used in demonstrations of electronic data exchange of real capital project data for the procurement of centrifugal pumps, to create catalogues of block valve specifications and in other proprietary applications (interface to heat exchanger simulation program, procurement 'data sheets').

1.2 cfiXML LT

The cfiXML LT initiative is a NIST sponsored project to encourage further implementation of cfiXML by establishing ways to simplify and reduce the effort of building practical applications that use cfiXML. The primary goal of cfiXML LT is to provide an easier and quicker route to developing standard-based data exchange applications. There are now a number of relatively simple XML application development tools available, which provide developers with little XML experience a means to build software applications. The LT project has focused on the challenges encountered by two parties developing applications using XML serialization tools to assess whether possible alterations in schema design may support easier implementation. Also, the contents and support provided by the full schema is extensive. Many relatively simple data exchange usage scenarios can be supported by a subset of the current schema model. For specific equipment items and usage scenarios, the LT project has aimed to define the common subset of required data specifications that an application should support.

This document developed to describe the goals, requirements and constraints for cfiXML LT and to be used as the input into the solution design phase, as well as providing the criteria for judging completion and success of the project. This information forms the core of the current document, with additions to: (1) support decision-making on whether LT or full schema will meet a user's requirements; (2) describe considerations, recommended practices and testing criteria when developing a cfiXML LT-based data exchange system, and (3) explain the LT-tool and process used to create schema versions.

1.3 Choosing between LT and Full cfiXML Schemas

When choosing between adopting the LT or Full cfiXML Schema, implementers must examine their system's needs and understand which version is best suited for their use-case. Most implementers approaching the use of cfiXML as a means of *transacting* or *exchanging* point-in-time data between 3rd party systems will find that the LT schema meets their immediate needs. Systems developers contemplating relying on cfiXML for data repository and management needs and for their underlying business models, will find full cfiXML schema more suitable.

For the implementer looking to adopt cfiXML in an existing application (import, export, or both), the easiest approach is to begin by supporting LT. Implementing LT will not preclude the implementer from upgrading the application to support full cfiXML if needed. The following is a summary of the differences in key features of full versus LT cfiXML. More detail can be found in section 5 and 6 of this document.

1.3.1 XML with SI, Metric, US Engineering or customized unit sets.

LT cfiXML focuses on supporting point-in-time data exchange only, and to simplify implementation it prohibits specification of unit sets within the XML document. All physical quantities are required to be stored in SI units within an LT cfiXML document and unit conversions are left to the applications working with the XML. Full cfiXML supports SI (the default), Metric and US Engineering unit sets and customization, which mean applications supporting full cfiXML must be capable of reading data in any unit set supported by the schema. See Section 4.3 for more details.

1.3.2 Revision History and Data Change Tracking

LT cfiXML documents are specifically **not** designed to store the history of an equipment data set over time. LT documents are designed to be *snapshots* of the data being exchanged between parties, where each application likely stores master copies of the data (perhaps with revision history) in their own backend systems. Revision support, which is embedded in full cfiXML schema, is removed from LT to provide a simplified data model. Applications which are required to include revision and version data, and change tracking of individual data items, within the XML should consider using the full schema version. LT cfiXML documents do contain XML elements suitable for time-stamping and commenting document-wide revisions. See Section 4.2 for more details.

1.3.3 Referencing Data Sets and Identifiers

The key idea of a cfiXML object is that it can be thought of as a "smart container" which can be accessed by its name, like understanding the contents of a box without having to open the box. Providing the mechanism to uniquely identify objects anywhere in the XML enables software to refer to and use these data objects (smart containers). So, data objects, can reference other objects elsewhere in the XML or even other resources. For example, a reference to a specific object via its unique identification can be used to locate data that is held elsewhere, either in the same file, or in different files over the network without having to replicate the schema or the data. (See schema guidelines 'Object Basics'.)

Full cfiXML guidelines require an implementer to uniquely identify objects by (at least) creating objectID attributes on most core data sets (objects) to enable object referencing, if required. This feature is useful in a variety of applications, as it allow a *software* artifact to be traced within an XML document, or across several XML documents and applications. While optional in LT, this function is not seen as required for simple data exchange and implementers are not required to use this functionality. When using cfiXML as an application's primary data model, the full version may be more suitable. See Section 4.1 for more details.

2 Goals and Constraints

The primary goal is to encourage increased use of cfiXML schema by providing a simplified version. This enables application developers to create initial data exchange systems easily, with a reduction in development time and costs.

2.1 Objectives

The following objectives describe the features and design requirements of cfiXML LT. Details of use cases for which cfiXML LT is recommended are described in Section 3.1 and Appendix 5.2.2.

1) Application developers with basic XML skills are to implement data exchange systems using simple XML serialization tools (e.g. XML Beans). This is to include the ability to use such tools to automatically generate object structures representing the complete schema for easy generation of valid XML.
2) XML documents based on cfiXML LT should validate against both LT and full schemas using standard XML validators.
3) Domain specific data dictionaries defining data items for a specified data exchange usage scenario, including associated XPaths, should form the primary reference and be provided in the public domain. There should be no need to refer to the schema or schema reference guide to create cfiXML LT data exchange applications[1].
4) Application developers with XML schema experience are to directly use schema LT without having to see or be concerned with items that support full cfiXML features.
5) One data item is to be represented by one XPath. The XPaths are to be unambiguous and standalone (no processing order dependencies). This objective excludes the use of references to external objects or to associated property values when defining a data item.
6) Each XML document is to contain the specifications for only one primary equipment item[2]. The specifications may include the number of identical items required.
7) The XML document is to only include data which represent a single point in time exchange in a defined usage scenario.
8) The XML document is to include all data found in a typical procurement data sheet revision table. For example, this would consist of the document revision identifier, date, description and the person who did the modification, checking or issuing.
9) All engineering values will be represented in SI units. Temperatures will be in Kelvin. No SI prefixes are to be used. For details see NIST reference (http://physics.nist.gov/cuu/Units/units.html).
10) Future maintenance, extension and upgrade of cfiXML LT are to involve only minimal additional effort beyond that required for full cfiXML.

[1] Note: Core semantic definitions of data elements are provided in cfiXML schema. The industry specific data dictionaries add clarification on the use of these elements in XPaths. These XPaths define the industry specific usage. During development of the industry specific data dictionaries core semantic definitions need to be considered, i.e. schema annotation should be read.

[2] Example 1: The equivalent of a procurement data sheet for a centrifugal pump assembly will include the specifications for the primary equipment item, that is, the centrifugal pump. The data set will not contain specifications for more than one type of centrifugal pump. The data set may contain details of the driver, gear, coupling, seal etc. Example 2: A motor-operated valve (MOV) has the primary equipment item of a valve with a motor operator.

2.2 Limitations

In order to achieve the goal to make LT easier to learn and use, the following features and capabilities of full cfiXML *will not* be supported by cfiXML LT

1) Unique identification of cfiXML objects and all object referencing mechanisms.
2) XML documents containing specifications for multiple primary equipment items. For example, this would include a document containing specifications for more than one type of centrifugal pump, or catalogues, such as the GVCC block valve catalogues, which contain multiple valves in the same document.
3) Revision and version management data, or change tracking of individual data items, within an XML document.
4) Bid tab comparison tables and matching functionality involving XML documents which contain multiple primary objects are supported by full cfiXML. (A bid tab application based on cfiXML LT will involve comparison of the contents of different XML documents, each of which describes one primary equipment item.)
5) Processing instructions, requirements, constraints and relationships defined in the AEX cfiXML guidelines and external to the schema, will be supported by full cfiXML. (As above, cfiXML LT will be defined entirely by schema file content.)

3 Criteria

Successful implementation of the cfiXML LT schema can be confirmed by use of a check list of measurable and testable criteria.

Usage scenarios which are supported by cfiXML LT were first analyzed in order to create tests based on real use cases and industry examples. The check list was then defined by consideration of the objectives outlined in Section 2.1 and a series of these use case tests. Passing all criteria indicates that the project is complete and successful.

3.1 Usage Scenarios

Bid, Quote, Purchase, and As Built data exchanges are the basis for the test cases for cfiXML LT. As Installed was considered for inclusion in this testing, but discussions with industry experts have not yet indicated any data requirements that are not also included in Purchase and As Built.

The tests are based on the following assumptions:

(1) The LT schema is to support use cases that are an exchange of data between 2 parties at a point in time.

(2) Each XML document is to include data found in a typical procurement data sheet.

(3) The required data elements for each exchange scenario are documented, based on input from industry experts and analysis of example exchange transactions for the different types of equipment.

The testing of exchanges of pump data will use the HI-EDE Standard 50.7 data dictionary for the procurement of centrifugal pumps, in which Required, Desired and Supplementary data items are defined. The aim is to test the full contents of the HI-EDE standard in 3 phases: (1) Required data items only; (2) Required and Desired items, and (3) Required, Desired and Supplementary items. Any sections of schema which are significantly altered or extended during the cfiXML Review project will also be tested.

Data dictionaries for other equipment (Motor Operated Valves, Control Valves, Electric Motors, Centrifugal Compressors and Centrifugal Fans) are being compiled. In the tests proposed below, the HI-EDE Standard 50.7 is used as an example.

Additional use cases for cfiXML LT are described in Appendix 5.2.2, including comments on limitations and reduced features of schema LT when compared with full schema support.

3.1.1 Bid

The Bid work process involves the compilation of process simulation, equipment design and other specifications into a data package which forms the Bid Request For Quotation (RFQ).

The Bid RFQ "is defined as the transmittal of technical requirements from purchaser to supplier to initiate a Request for Quotation (RFQ). This transaction requires the purchaser to convey all critical information needed for the supplier to understand the application, select a pump, and respond with a qualified pump quotation, with complete confidence."(HI-EDE Standard 50.7, section 50.7.3.3).

3.1.1.1 Proposed Test Case

- Required, Desired and Supplementary data items, as defined in the HI-EDE Standard 50.7 for centrifugal pumps BidRFQ, to be created as an XML document and exported to the supplier.

3.1.2 Quote

The Bid Quote, "is defined as the transmittal of pump performance and configuration data typically contained in a technical quotation. This transmittal shall have sufficient detail to permit the purchaser to assess technical quotations from different suppliers" (HI-EDE Standard 50.7, section 50.7.3.3).

3.1.2.1 Proposed Test Case

- Supplier to import the BidRFQ 'data sheet' XML document.
- Required, Desired and Supplementary data items, as defined in the HI-EDE Standard 50.7 for centrifugal pumps BidQuote, to be added to the XML document, which is then exported to purchaser.

3.1.3 Purchase

Technical quotations from different suppliers are usually assessed by the purchaser through bid tabulation. The purchaser will then issue a Purchase Order (PO) with the data sheet to the selected supplier.

3.1.3.1 Proposed Test Case

- Purchaser to import the BidQuote 'data sheet' XML document.
- Purchaser to add Purchase Order (PO) document, PO number, and any additional data required for defining the purchase request, to the Required, Desired and Supplementary data items, as defined in the HI-EDE Standard 50.7 for centrifugal pumps BidQuote. The XML document is then exported to the supplier.

3.1.4 As Built or As Installed/Handover

Upon completing the manufacture of the purchased equipment, the supplier will issue an As Built datasheet to the purchaser. A list of spare parts, General Arrangement Drawings, and Operating and Maintenance

Manuals are commonly provided with the As-Built datasheet[3]. QA specifications such as material certifications, test certification, and other test data may also be included with the As-Built data. These documents are generally provided in digital format and reference to them should be included within the XML document.

The As Installed/Handover use case is expected to contain the same type of data as the As Built use case, except the data is provided at a different point in time (after the equipment has been installed).

3.1.4.1 Proposed Test Case

- Supplier to create an 'As-Built' data package by adding references to material and test certification documents from the API data set ('A' designated fields), to the BidQuote data set, as defined in the HI-EDE Standard 50.7. The XML document is then exported to the purchaser.

3.2 Check list

All criteria listed below, which need to be measurable and testable, have been passed to show that the cfiXML LT development project is complete and successful. The HI-EDE Standard 50.7 data dictionary is used as an example, reference to which can be substituted by data dictionaries for other equipment items and use cases, as required.

1) Conversion from full cfiXML schema to LT involves a simple process of less than 30 minutes and publically available scripts. [Objective 2.1]
2) Confirm all XPaths in current HI-EDE Standard 50.7 data dictionary covering BidRFQ and BidQuote specifications for centrifugal pumps are unambiguous and standalone and no explanatory comments are required. [Objective 2.1]
3) Create cfiXML LT test files for BidRFQ and BidQuote use cases using XML Beans and only using the HI-EDE data dictionary. The test files should include revision 'table' data and all physical property values should be in SI units[4]. [Objectives 2.1, 2.1, 2.1, 2.1]
4) The LT schema version will only contain the elements or attributes which support the cfiXML LT feature set (for example, LT will not contain elements or attributes which support object-level revisions and versions). [Objective 2.1]
5) Validate all cfiXML LT test case documents, created for at least 2 of the following 4 'real usage scenarios', against both LT and full schema using at least 3 standard XML validators. Bid RFQ and Bid Quote are the preferred 2 data exchange scenarios as they should include the greatest variation and content of data specifications. The remaining 2 options fundamentally test referencing requirements to external, supporting documents.

   This will confirm schema design, 'fit-for-purpose' and XPath validity. [And Objectives 2.1, 2.1, 2.1]

   Point in time data exchanges to be represented by XML documents are:

BID RFQ

[3] For pumping equipment, these data are not supported in the HI-EDE 50.7 standard by definitions or Required, Desired and Supplementary designations. However, the HI-EDE 50.7 standard does include cfiXML XPaths ('A' fields) to describe additional data items for the more highly engineered process equipment as specified in the API Mechanical Equipment Standards 610 10th and 11th edition

[4] This item can form part of the tests outlined in item (5).

a. **Purchaser:** BidRFQ data set created and exported from 1st application (XML document 1a);

b. **Supplier:** BidRFQ data imported into 2nd application and exported (XML 1b);

(Confirm XML 1a = XML 1b)

BID Quote

c. **Supplier:** BidRFQ data imported; BidQuote data added, file exported (XML 2a);

d. **Purchaser:** BidQuote data imported and exported (XML 2b);

(Confirm XML 2a = XML 2b)

Purchase

e. **Purchaser** BidQuote data imported, Purchase Order (PO) data added, file exported (XML 3a);

f. **Supplier:** PO and BidQuote data imported and exported (XML 3b);

(Confirm XML 3a = XML 3b)

As Built / As Installed

g. **Supplier:** PO and BidQuote data imported; As-Built data added, file exported (XML 4a);

h. **Purchaser:** PO and BidQuote data imported and exported (XML 4b);

(Confirm XML 4a = XML 4b)

Note: In the As Built use case, the Supplier is usually the equipment manufacturer and the Purchaser is the End User or EPC. In the As Installed use case involving handover, the Supplier is typically the EPC and the Purchaser is the End User.

4 cfiXML LT Features

This section provides details on each feature of cfiXML LT compared to the features of full cfiXML schema.

Developing cfiXML-based implementations using XML serialization tools, and the associated challenges, highlighted a few key design elements in full schema to assess. These included the use of polymorphic derivable types and the required object identifier. Following detailed consideration of potential changes to schema to simplify application development, it was agreed that full schema would be changed. This resolution ensured, for example, that LT schema would be compatible with full cfiXML schema. Therefore, LT and full schema no longer include polymorphic derivable types and as the replacement design structure (substitution groups) is common to both, it is not discussed further in this document. If any of the alterations made to full schema to support the use of XML serialization tools still require a different approach when a developer is using LT versus full schema, then these points are discussed below. The alterations to full schema associated with this project are included in cfiXML Schema Public Release v 4.0 and are described in the associated release notes[5].

[5] See also MantisBT Support for the project at https://sourceforge.net/projects/cfidev/. The next version of the schema guidelines will also include details.

LT is a sub-set of full schema, so it is conceptually acceptable for an LT XML file to use the full cfiXml namespace. The LT cfiXML schema can be thought of as a convenience for implementers who are generating LT XML and as a machine readable means of defining the LT subset of full.

LT schema files include the namespace declaration 'xmlns:aexLt=http://www.cfixml.org/Light/aexLt' which can be used by software to identify LT xsd files.

To indicate that an XML file adheres to the LT subset schema, an additional, optional, Boolean attribute 'isLightXml' has been added to the objb:BaseObject.

Below, each cfiXML LT feature is described, with the rationale for it. If there are any points that an application developer should consider when deciding whether to base their software on LT or full schema, these are also outlined, with illustrative examples.

4.1 Optional ObjectID

The objectID attribute in cfiXML objects is altered from 'required' to 'optional'.

Though this alteration occurs in full cfiXML, full schema should adhere to the schema guidelines and so objectIDs remain a required attribute. This ensures consistency in all full cfiXML-based applications and resulting XML documents.

Unless parties involved in a data exchange sequence have agreed to include objectIDs, then it must be assumed that objectIDs will be omitted in cfiXML LT documents.

In addition to this fundamental change, the objectID attribute is no longer restricted by pattern to enable LT schema users to enter any string, if required. Again, full schema users should adhere to the requirements outlined in the guidelines to maintain a consistent approach.

(See Section 1.3.3 for an explanation of cfiXML objects.)

4.1.1 Rationale

LT Schema-based XML documents can be created with no reference or presence of objectID attributes in the document and still validate against both LT and full schemas using standard XML validators.

The format required to create unique objectIDs is specified in the schema guidelines: there is a defined pattern for users of full schema. A cfiXML LT user will not need to use guidelines.

An application developer will use schema LT without having to see or be concerned with objectID attributes.

4.1.2 Implications

Objects in cfiXML LT cannot be uniquely identified. For the development of data exchange systems typified by the work processes defined in section 3.1, object identification was considered by the XML serialization tools users as unnecessary in most cases.

The cfiXML object identifying attributes were designed to support a comprehensive data exchange system and associated applications. This might, for example, include an interface to an equipment process simulation program and reference to a local database of material or fluid properties. The cfiXML object can be uniquely identified globally through the use of the provided attributes.

A fundamental purpose of cfiXML objects are "to pass and maintain data on any single object throughout the life cycle of a facility item, from one domain expert to the next, across multiple usage scenarios.

Schema object design also includes support to record and track data changes and maintain a full version and revision system. Without the capability of a persistent, object-oriented structure, such functionality becomes extremely complex and difficult." (See schema guidelines.)

4.1.2.1 Object Referencing

The full cfiXML mechanism to reference a specific object via its unique identification is not supported by cfiXML LT.

The full cfiXML mechanism can be used to locate persistent data that is held elsewhere, either in the same file, or in different files over the network without having to replicate the schema or the data.

a. Example (1) a process simulation program with a pre-existing database of fluid properties or material stream components used in the simulation might be referenced via the cfiXML full object mechanism.

b. Example (2) use of external XML standards to extend cfiXML is also facilitated by use of objectID attributes.

4.1.2.2 Compatibility of LT documents and Full schema-based Applications

Conversion or inclusion of data from cfiXML LT documents into full schema –based documents will be significantly (potentially prohibitively) more challenging for application developers due to the lack of identifiable cfiXML objects in the LT documents.

XML documents without objectIDs will not follow the full schema guidelines requirements. Applications requiring all XML documents to adhere to the full cfiXML guidelines will not be required to process XML documents which omit objectIDs if they make no claim of supporting cfiXML LT.

4.2 Revisions, versions and change tracking

The cfiXML LT schema defines all data items found in a typical procurement data sheet revision table. For example, this would include document revision identifier, date, description and the person who did the modification, checking or issuing.

All elements and attributes associated with full schema support for revision, versioning and change tracking at the object level, and not required in the above usage, will not be available in cfiXML LT.

4.2.1 Rationale

The primary purpose is to simplify cfiXML full. The elements and attributes associated with revisions, versioning and change tracking are mainly located in cfiXML objects. Additional attributes on all elements flag changes made to data. An application developer will use schema LT without having to see or be concerned with these schema components.

Versions of XML documents based on cfiXML LT can be managed in document management systems. Users in an organization will be able to create an audit trial using familiar in-house procedures.

4.2.2 Implications

4.2.2.1 Use of In-house management systems

When using cfiXML LT, parties who wish to manage different versions of documents, will use a document management system. The XML documents will not contain the information on who changed what and in which version or revision, as supported by full schema.

The consequences of the use of in-house document management systems in place of containing these data within the XML may include:

1) Change, version and revision information will only be available to the users of the in-house document management system and will not be available to any partners, associates, customers or other parties with whom data is exchanged. That is, the change, version and revision data are not being considered part of the data sheet information to be exchanged.
2) Audit trails of who made what change when will not be available if the data were exchanged. This then removes the benefits for legal purposes.
3) Contractors, suppliers to large organizations and other service providers may be required to maintain multiple document management systems to be able to be compatible with all their clients' systems.

4.3 SI units of measurement

Only SI units of measurement are to be used for physical property values in cfiXML LT documents. Temperatures will be in Kelvin. No SI prefixes are to be used. For details see NIST reference (http://physics.nist.gov/cuu/Units/units.html).

4.3.1 Rationale

The rule that only SI units are to be entered into cfiXML LT documents results in significant simplification of application development.

Attributes used to support dynamic units of measurement in full schema can be filtered out to simplify schema structure. The removal of these attributes ensures that users do not have the option of using other unit sets, as provided in the full schema version.

4.3.2 Implications

Dynamic units of measurement are not supported by cfiXML LT.

In full schema each physical quantity type has a defined attribute 'symbol', which is used to set the units for the measured quantity and is recorded in the instance file. Objects include the attribute 'unitSet', which can be set to the name of a unit set such as 'SI', 'USEngineering' or 'Metric'. This provides a means of specifying units for contained elements. The default unit set in full schema is 'SI' which ensures that if an application imports a cfiXML LT-based document, the units of measurement for physical property values can be correctly supported. (See schema guidelines for further details.)

4.4 Schema simplification

In addition to removal of elements and attributes associated with cfiXML objects, revisions, versions and change tracking, and unit of measurement support, particular schema files and schema components have been identified as not required for cfiXML LT. All items that were removed or altered to create the version of LT schema are listed in Appendix 5.2.2, with rationale. LT schema downloads will always include release notes specifying these details.

Nillable: This feature is to enable recording of change from a value to no value as a user deletes. It is supported in full schema as low level functionality so that schema developers do not have to specifically implement nillable to every new element. It has been removed from cfiXML LT to simplify application handling of numeric data items.

5 Process and Tool Requirements for the Creation of cfiXML LT

5.1 Process to create LT and Full Schema

From Public Release v 4.0, two versions of cfiXML Schema are available to download from https://sourceforge.net/projects/cfidev/.

The following describes the contents of the schema downloads and the process used for the creation of the full and LT schema versions. All the zip files contain a 'schema' and an 'examples' directory.

5.1.1 Base schema in CVS

The base schema is the working version of schema in which all extensions and developments are undertaken. This is shared between parties developing schema in WinCVS (see cfiXML Process Document http://cfidev.sourceforge.net/documents/cfiXML%20Process%20Document.pdf).

Following all validation checks, the LT-tool is used to create the LT and Full schema versions. This tool adds specific ('aexLt') attributes into the base schema files marking schema components which are to be removed or altered when the LT Schema version is created.

5.1.2 cfiXmlFull[version].zip

This is the full cfiXML schema. This is created from the base schema by using the LT-tool 'clean' functionality, which removes any aexLt attributes and the aexLt namespace declaration from the base schema files.

5.1.3 cfiXmlLT[version].zip

This is the cfiXML Light (LT) schema, created from the base schema by removing or altering components marked by aexLT attributes. Appendix 5.2.2 lists the alterations made to full schema to create the first version of schema LT. Future alterations to LT schema will be detailed in the release notes associated with the cfiXmlLT.zip download.

5.2 The LT-Tool

The LT-tool was created in Microsoft Visual Studio 2008 to meet the following specifications.

5.2.1 Functionality

The tool functionality can:

1. Filter out elements, global complex and simple types and attributes using an easily extended and changed specification;
2. Process base cfiXML into a specified version of cfiXML LT in a single batch run without requiring user interaction;
3. Change the minOccurs and maxOccurs attribute values;
4. Automatically remove comments that are immediately above a node that is being removed.
5. Open multiple files, and undertake global search / change across all open documents;

6. Automatically list all schema components that have been removed or altered to create cfiXML LT (to form release notes).

The definition of the LT schema is maintained in the base cfiXML xsd files as additional attributes associated with a namespace “aexLt”.

The tool strips out all the “aexLt” additions in the base cfiXML so that the published full cfiXML does not contain any items required for the Lt.

All aexLt additions are included in such a way that they can be manually added, updated and removed using a standard Schema editor, or even a text editor. Additionally the processing to generate the Lt schema and the full cfiXML for publication should be possible using W3C standard XSLT 2.0, in addition to any public domain programs that are made available.

5.2.2 Availability / Development – Recommended Components and Quality Process

The tool, and components it is dependent on, are:

1. Publically available for free - the finished executable is a download on the cfixml.org web site;
2. The source is uploaded to CVS on SourceForge under the cfidev project (i.e. it is available from the same Web location where cfiXML development files are maintained);

The development and maintenance of the tool is supported by:

1. User documentation;
2. Programmer documentation for others to be able to maintain and enhance the software, e.g. development IDE required;
3. Production quality to include code reviews so that it is clear to any MS Visual Basic programmer;
4. Quality assurance testing to locate any major bugs and ensure the software works reliably;
5. Keeping it simple to reduce learning curves.

A. Applicability of cfiXML LT for Example Use Cases

Key limitations of cfiXML LT are associated with the lack of uniquely identified objects, and revision and version management and change tracking data not being contained within the LT XML documents. The subsequent recommended uses for cfiXML LT are described below.

[The primary source for the use cases is the HI-EDE Standard 50.7 section 50.7.2 and 50.7.3.]

Use Case	Critical schema features required to support use case	Applicability to the LT Schema
• Process simulation	Connectivity between objects via object references may be required. Example: material stream components data set in simulation program linked via object referencing to unit streams in schema.	Not recommended – LT schema will not support object referencing.
• Equipment design	Data that currently appears on equipment design data sheets.	Yes, as applicable to an equipment datasheet
• Equipment data sheet production[6]	–	Yes, as applicable to an equipment datasheet
• Equipment and instrument tracking	Data sheet identifiers, unique within the scope of the equipment and instrument information flow.	Yes, excluding revision and version management, change tracking contained within the XML.
• Intelligent piping and instrumentation diagram (P&ID) production	Connectivity between items via object references. Inclusion of links to graphical representations.	Not suitable for LT. (Use of full cfiXML possible with inclusion of data or referencing external XML object data set.)
• Cost estimating	Inclusion of economic information.	Not recommended but possible.
• Procurement[7]	–	Yes, related to the bid, quote, and as-built engineering equipment specification data.
• Bid tabulation	Identification of the source of each bid and the original request for bid. Unique object or data sheet identification and tracking.	Yes, by the comparison of XML document contents, but not object content.

[6] Equipment data sheet production is considered as allowing import, export and exchange of XML data between software systems.

[7] Procurement is considered as the exchange of engineering specifications for items being purchased and excluding management of the procurement process.

Use Case	Critical schema features required to support use case	Applicability to the LT Schema
• Order entry and tracking	Unique identifiers, data management and data flow management meta data.	Yes, related to the bid, quote, and as-built data if the required data is included at data sheet level and no revision, version or change tracking is required.
• Collaboration	Unique identifiers and meta data providing information on the source and history of a document.	cfiXML LT can provide basic electronic data exchange and the required data can be included at data sheet level.
• E-Marketplace	(Limited to support for the electronic data exchange of engineering specifications related to this usage scenario.) Unique identification of data sheets.	Yes, with unique data sheet /XML document ids.
• Planning, scheduling, and tracking	Integrated use of other schemas and data formats for planning and scheduling standards. Revision and version management data and change tracking.	Not suitable for LT.
• Enterprise resource planning (ERP) – asset management	Integrated use of other schemas and data formats for ERP systems. Unique item identifiers and object referencing. Revision and version management data and change tracking.	Not suitable for LT.
• Maintenance management	Integrated use of other schemas and data formats for ERP systems. Unique item identifiers and object referencing. Revision and version management data and change tracking.	Not suitable for LT.
• Integration	Unique object identifiers. Object referencing. Source history and meta data.	Not suitable for LT.
• Data warehouse	Unique object identifiers. Exposing meta data for data warehouse mining. Object referencing. Revision and version management and change tracking.	Not suitable for LT.

B. Applicability of cfiXML LT to Common Engineering Document Types

Document Type	LT Applicable?
• Process simulation reports	No
• Equipment design reports	Yes
• Equipment data sheet	Yes
• Equipment list	No
• Bill of material	No
• Intelligent drawing	No
• Equipment status update	No

C. Files, elements, attributes filtered out from Full Schema to Create cfiXML LT

File	Element or Attribute Names	Comment
ALL FILES	customXXX elements (not enums)	No extension should be done to LT version. Full should be checked first as the requirements may be available.
/document/cfiXml.xsd	xsd:import - eqDoc.xsd; mtrlDoc.xsd; procSimDoc.xsd	If remove, then the following 3 files can be excluded from LT Schema
/document/eqDoc.xsd	_	Whole file not required in LT. (NB: eqDoc.xsd imported in cfiXml.xsd and idx.xsd)
/document/mtrlDoc.xsd	_	Whole file not required in LT (NB: only imported into cfiXml.xsd)
/document/procSimDoc.xsd	_	Whole file not required in LT (NB: only imported into cfiXml.xsd)
/document/eqElecDoc.xsd, eqHvacDoc.xsd, eqHxDoc.xsd, eqPvfDoc.xsd, eqRotDoc.xsd	-EquipmentList, -EquipmentOrder elements and types	Not required for LT
/document/eqPvfDoc.xsd	valveCatalog*, entry, equivalentValveEntry, organizationList	All associated with valve catalogues which require unique cfiXML objectIDs
(extCur.xsd)	_	(Whole file not required in LT but is part of ext namespace and is included in ext.xsd –therefore to be considered as option for the future.)
idx.xsd	_	Whole file not required in LT
objb.xsd	sourceObject, copiedObject, **obj:BaseObject elements:** sourceObjectReference (is element ref); customBaseObject	Elements and attributes in cfiXML objects, which support identification, units, referencing, versions, revisions etc. which are not available in cfiXML LT.

	obj:BaseObject attributes: approvedNameSpace, contentType, contextURL, currencyCode, directReferenceURI, language, objectState, organizationID, ownerUserSchemaRelease, namespace, schemaRelease, unitSet, version, versionIndex, xmlContent, ##other. sourceObjectReference (element), baseObjectReference (type + element), EObjectState (and element), EObjectStateReference (and element), EContentType (and element), EXMLContent (and element), EApprovedNameSpace (and element)	
ext.xsd	**BaseAttributeGroup attributes**: comment, commentReference, changed, recipientToProvide, revChanged. **BaseObjAttributeGroup attributes**: comment, commentReference, changed, revChanged. **BaseEnumerationAttributeGroup attributes**: comment, commentReference, changed, recipientToProvide, revChanged, ##other [retaining alternativeOption, otherValue] NilBaseA and BooleanNilA, ByteNilA, DoubleNilA, DoubleWordNilA, DurationNilA, IntegerNilA, LongIntegerNilA, ShortIntegerNilA, StringNilA, URINilA, Base1IndexNilA	Elements and attributes which support change tracking and revisions, comment and comment referencing and schema extension - not available in cfiXML LT. Nillable feature is not available in cfiXML LT: NilBaseA removed; [dataType]NilA types are edited to remove 'NilBaseA' from the union from which they are formed.
obj.xsd	obj:Object: previousVersion (element)	Revision and versioning support – not required
Schema/custom	folder and all contents	Full schema provides full customization options. These files illustrate how to customize schema and are referred to in the schema guidelines.