
eCx XML Specification

A Requisite Technology White Paper

October 1998

Version 1.0

Overview

This document provides an outline of the Requisite eCatalog XML Specification for electronic catalog interoperability and electronic content interchange.

Purpose

The purpose of the eCatalog XML specification is to provide a method of updating a catalog's structure, or schema, and its electronic content from a variety of sources and content formats. The XML format has been designed to allow import of eContent into a BugsEye catalog or any catalog format. It supports the following functionality:

- Modification of catalog schema
 1. Add, rename, and delete a category
 2. Add, rename, and delete a descriptor, or attribute of a category
- Addition of items
 1. Place into specified category
 2. Place into "holding" category
 3. Internationalized data supported
- Deletion and modification of items

Specification

The design goal of this specification is to allow maximum flexibility by not specifying absolute tags, catalog schema, types, or formats in the XML document type definition (DTD).

Document Type Definition (DTD)

```
<?xml version="1.0"?>
<!DOCTYPE CATALOG [

  <!ELEMENT CATALOG ( ADMIN, SCHEMA?, DATA? ) >
  <!ATTLIST CATALOG xml:lang NMTOKEN #IMPLIED >

  <!ELEMENT ADMIN ( NAME, INFORMATION) >
  <!ELEMENT SCHEMA (CATEGORY | DESCRIPTOR | UNIT)* >
  <!ELEMENT DATA (ITEM, DATAEXTENSION?)*>

  <!ELEMENT NAME (#PCDATA) >
  <!ATTLIST NAME xml:lang NMTOKEN #IMPLIED >
  <!ELEMENT INFORMATION ( DATE, SOURCE, AUTHOR?, TITLE, TITLELOGO?, DOMAIN? ) >
  <!ELEMENT DATE (#PCDATA) >
  <!ELEMENT SOURCE (#PCDATA) >
```

```

<!ELEMENT AUTHOR (#PCDATA) >
<!ELEMENT TITLE (#PCDATA) >
<!ELEMENT TITLELOGO (#PCDATA) >
<!ELEMENT DOMAIN (#PCDATA) >

<!ELEMENT CATEGORY (NAME | KEY | TYPE | UPDATE | OWNER? )* >
<!ATTLIST CATEGORY ACTION (ADD | DELETE | UPDATE ) #REQUIRED>

<!ELEMENT DESCRIPTOR (NAME | KEY | OWNER | TYPE | LENGTH |
VISIBILITY | SEARCHABLE | MULTIVALUE | UPDATE)* >
<!ATTLIST DESCRIPTOR ACTION (ADD | DELETE | UPDATE | NONE) #REQUIRED>
<!ELEMENT OWNER ( (NAME, KEY? ) | KEY) >
<!ELEMENT MULTIVALUE (VALUE)+ >
<!ELEMENT KEY (#PCDATA) >
<!ELEMENT TYPE (#PCDATA) >
<!ELEMENT LENGTH (#PCDATA) >
<!ELEMENT VISIBILITY (#PCDATA) >
<!ELEMENT SEARCHABLE (#PCDATA) >

<!ELEMENT UNIT (NAME | KEY | OWNER | TYPE | UNITELEMENT | UPDATE )* >
<!ATTLIST UNIT ACTION (ADD | DELETE | UPDATE) #REQUIRED>
<!ELEMENT UNITELEMENT ( NAME | KEY | MULTIPLIER | ADDER | UPDATE )* >
<!ATTLIST UNITELEMENT ACTION (ADD | DELETE | UPDATE) #REQUIRED>
<!ELEMENT MULTIPLIER ( #PCDATA ) >
<!ELEMENT ADDER ( #PCDATA ) >

<!ELEMENT ITEM (OWNER?, NAMEVALUE*, UPDATE?, DATAEXTENSION*) >
<!ATTLIST ITEM ACTION (ADD | DELETE | UPDATE) #REQUIRED>

<!ELEMENT UPDATE (NAME | KEY | LENGTH | VISIBILITY | TYPE |
SEARCHABLE | NAMEVALUE | DATAEXTENSION | UNITELEMENT)* >
<!ELEMENT DATAEXTENSION (DATAELEMENT | NAME | TYPE )* >
<!ELEMENT DATAELEMENT (NAMEVALUE | UPDATE )* >
<!ATTLIST DATAELEMENT ACTION (ADD | DELETE | UPDATE ) #REQUIRED>

<!ELEMENT NAMEVALUE ( NAME, VALUE, UNITTYPE? ) >
<!ELEMENT VALUE (#PCDATA)* >
<!ATTLIST VALUE xml:lang NMTOKEN #IMPLIED >
<!ELEMENT UNITTYPE (#PCDATA) >
<!ATTLIST UNITTYPE xml:lang NMTOKEN #IMPLIED >
]>

```

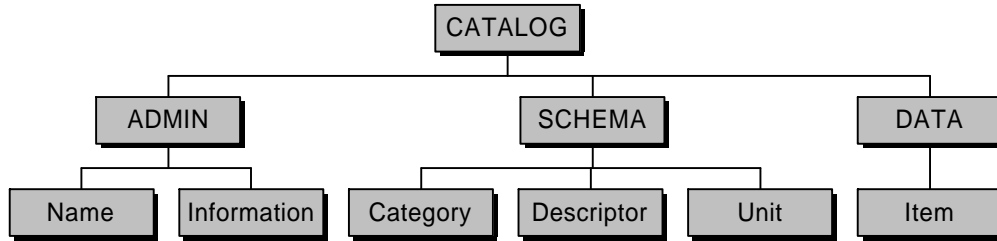
The DTD is intentionally built with use of the OR construct and PCDATA. While this makes the data definition somewhat less "type-safe", it provides maximum flexibility. This approach also allows arbitrary ordering. Alternative approaches require a very stringent data definition and severely restrict flexibility. Still other alternatives do not use a DTD for validation.

Catalog information is divided into three main categories:

1. Administrative information <ADMIN>
2. Catalog Schema <SCHEMA>
3. Catalog Data Items <DATA>

The following diagram shows a hierarchical diagram of the data type definition.

eCatalog's Data Type Definition



Administrative Information

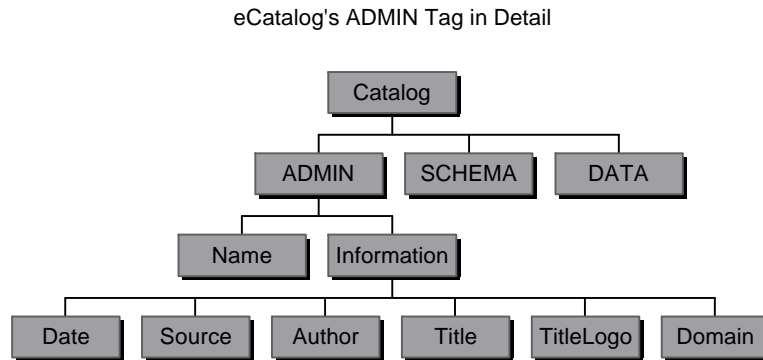
The admin tag is used to identify the catalog, and provide additional information about the catalog such as date, source, author, title, titlelogo, and domain. This tag is required for catalogs. The following is an example of the admin tag

```
<ADMIN>
  <NAME>Business Essentials Reference Catalog</NAME>
  <INFORMATION>
    <DATE>August 5, 1998</DATE>
    <SOURCE>Requisite Technology Inc.</SOURCE>
    <AUTHOR>eContent Manager 2.0</AUTHOR>
    <TITLE>BERC</TITLE>
    <TITLELOGO>CorporateExpress.gif</TITLELOGO>
    <DOMAIN>Products</DOMAIN>
  </INFORMATION>
</ADMIN>
```

The information contained in this section is source catalog-specific and does not impact the loading of schema or data for other catalogs. This administration information can be used by the catalog load process for a variety of purposes. For a Requisite BugsEye catalog, we recommend the following:

- **Name** is the descriptive name of the catalog.
- **Date** is catalog creation, or extraction, date. We recommend the following formats:
August 5, 1998, or 08/05/1998.
- **Source** is the company (content provider, supplier, etc.) that created the XML.
- **Author** is the author of the document. It could contain a tool name and version number, or person.
- **Title** represents the name of the catalog.
- **Titlelogo** is an optional tag. It defines a logo to be associated with the title for content branding..
- **Domain** represents the type of content in the catalog. Some representative values are Products, Services, People, Courses, and other content types.

The following diagram shows a hierarchical diagram of the ADMIN tag.



Catalog Schema and Schema

A <SCHEMA> tag contains the sub-tags Category, Descriptor, and Units, which are referenced as schema. Each schema tag has an attribute that specifies a command and sub-tags that define a key and actionable data. The commands are ADD, UPDATE, DELETE, and NONE, which are dependent on the schema tag used.

Schema may consist of Categories, Descriptors, or Units. They need not be in any particular order. A Category is a specifically defined division in a classification. It allows you to group classes of objects for organization. Descriptors are attributes that describe the categories, and Units can be used to define a descriptor's standard of measure. Below are descriptions of each section along with examples for clarification

Category

A Category may consist of a name, key, or both. When a key is provided, it is given precedence in identifying a category. Keys are important when the text representation, or name, or a category is subject to change (e.g., pen is changed to pens and content exchange would fail). A category may also contain a type, an owner, or an update tag. Category types can be genus, navigation, query, or other application specific types. If the type tag is not provided, the default type is a "Genus". The owner tag is optional and can be used for categories for other catalog technologies that require a hierarchical schema. If the owner tag is supplied it will provide the capability to create a table of contents for the catalog. Below is an example of using the owner tag to build a table of contents

```

<SCHEMA>
  <CATEGORY ACTION="ADD">
    <OWNER> <NAME>Root</NAME> </OWNER>
    <NAME> Office Supplies </NAME>
    <TYPE>NAVIGATION</TYPE>
  </CATEGORY>
  <CATEGORY ACTION="ADD">
    <OWNER> <NAME>Office Supplies</NAME> </OWNER>
    <NAME>Pen Gift Sets</NAME>
    <TYPE>GENUS</TYPE>
  </CATEGORY>
  <CATEGORY ACTION="ADD">
    <OWNER> <NAME>Office Supplies </NAME> </OWNER>
    <NAME>Notebooks</NAME>
  </CATEGORY>
  
```

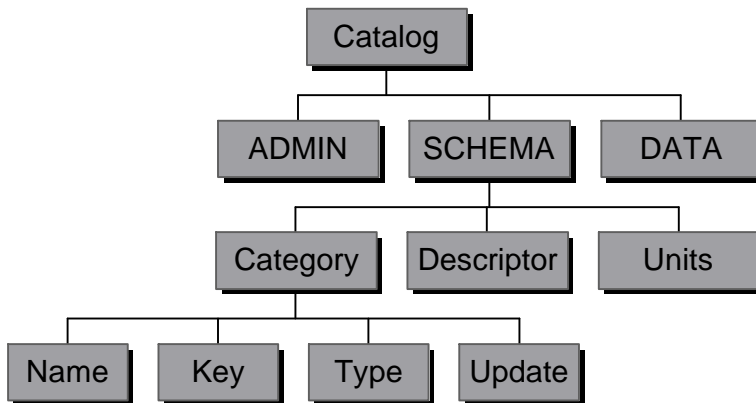
```
<TYPE>GENUS</TYPE>
</CATEGORY>
<CATEGORY ACTION="ADD">
  <OWNER> <NAME>Office Supplies </NAME> </OWNER>
  <KEY>Dry Erase Markers</KEY>
  <TYPE>NAVIGATION</TYPE>
</CATEGORY>
</SCHEMA>
```

Specific category types are not specified. The following types are recommended for categories:

- Genus (default)
- Navigation
- Template
- or other application specific types

The following diagram shows a hierarchical diagram of the SCHEMA Category tag.

Category tag in detail



Descriptor

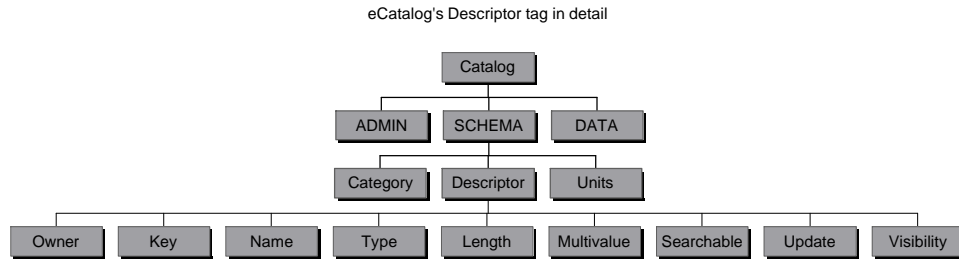
A Descriptor may consist of a name, key, or both. When a key is provided, it is given precedence in identifying a descriptor. A descriptor also contains a type, an owner, or an update tag. The type tag for the descriptor should not be confused with the category type. If the type tag is not provided, the catalog load process will determine the default type. For BugsEye catalog loading, the default type is a "String". The owner tag is required for descriptors. It associates the descriptor with a category. If you specify an owner of "Root", or "0", the descriptor will be available for all categories and items. When a descriptor has an owner of "Root", this means that the descriptor is a root, or base, descriptor. Base descriptors are common among all items in a catalog. Examples of a Base descriptor are a part number, description, or Stock Keeping Unit (SKU). A local descriptor is specific to a particular category (e.g., Ink Color for Pens).

The following are recommended types to be used for descriptors:

- String (default for BugsEye catalogs)
- Numeric

- Enumerated
- Graphic
- URL
- Date
- Currency
- International String

The following diagram shows a hierarchical diagram of the SCHEMA Descriptor tag.



Unit

A Unit is necessary for describing a descriptor's standard of measure. A Unit consists of an owner which associates it with a descriptor, a name or key, and a unit element. A Unit may also contain an update tag. Below is an example of how to define a unit "Currency" for Price.

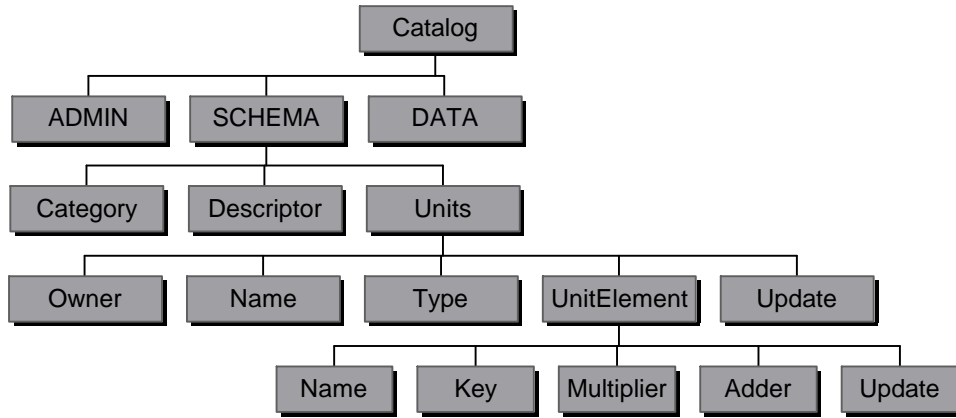
```

<SCHEMA>
  <UNIT ACTION="ADD">
    <OWNER> <NAME>Price</NAME> </OWNER>
    <NAME> Currency</NAME>
    <UNITELEMENT ACTION="ADD">
      <NAME> Dollar </NAME>
      <MULTIPLIER>1</MULTIPLIER>
      <ADDER> 0</ADDER>
    </UNITELEMENT>
    <UNITELEMENT ACTION="ADD">
      <NAME> Canadian Dollar </NAME>
      <MULTIPLIER>1.47</MULTIPLIER>
      <ADDER> 0</ADDER>
    </UNITELEMENT>
    <UNITELEMENT ACTION="ADD">
      <NAME> Franc </NAME>
      <MULTIPLIER>5.5</MULTIPLIER>
      <ADDER> 0</ADDER>
    </UNITELEMENT>
  </UNIT>
</SCHEMA>
    
```

Note the Multiplier and adder tags. These are used to specify a conversion for derived units from the base unit. In the example above, a Franc is multiplied by 5.5 to equal the base unit Dollar.

The following diagram shows a hierarchical diagram of the SCHEMA Unit tag.

eCatalog's Unit tag in detail



Examples: Add Category, Descriptor, or Unit

The example below adds a new category and descriptors to the new category and an existing category.

```

<SCHEMA>
  <CATEGORY ACTION="ADD">
    <NAME>Pen Gift Sets</NAME>
    <TYPE>TEMPLATE</TYPE>
  </CATEGORY>
  <CATEGORY ACTION="ADD">
    <NAME>Notebook</NAME>
    <TYPE>GENUS</TYPE>
  </CATEGORY>
  <CATEGORY ACTION="ADD">
    <KEY>Laptop Computer</KEY>
    <TYPE>NAVIGATION</TYPE>
  </CATEGORY>

  <DESCRIPTOR ACTION="ADD">
    <OWNER>
      <NAME>Pen Gift Sets</NAME>
    </OWNER>
    <NAME>Product Shipping Weight</NAME>
    <TYPE>String</TYPE>
    <LENGTH>500</LENGTH>
    <VISIBILITY>YES</VISIBILITY>
    <SEARCHABLE>YES</SEARCHABLE>
  </DESCRIPTOR>
  <DESCRIPTOR ACTION="ADD">
    <OWNER>
    
```

```

        <NAME> Pens </NAME>
    </OWNER>
    <NAME>Pen Color</NAME>
    <TYPE>Enumerated</TYPE>
    <MULTIVALUE>
        <VALUE> Red </VALUE>
        <VALUE>Green</VALUE>
        <VALUE> Blue</VALUE>
        <VALUE> Indigo</VALUE>
    </MULTIVALUE>
</DESCRIPTOR>

<UNIT ACTION="ADD">
    <OWNER>
        <NAME> Product Shipping Weight </NAME>
    </OWNER>
    <NAME>Weight</NAME>
    <TYPE>Numeric</TYPE>
    <UNITELEMENT ACTION="ADD">
        <NAME> Pounds </NAME>
        <MULTIPLIER>1</MULTIPLIER>
        <ADDER> 0</ADDER>
    </UNITELEMENT>
    <UNITELEMENT ACTION="ADD">
        <NAME> Gram </NAME>
        <MULTIPLIER>453.5</MULTIPLIER>
        <ADDER> 0</ADDER>
    </UNITELEMENT>
</UNIT>

</SCHEMA>

```

Examples: Delete Category, Descriptor, or Unit

Schema require a NAME, or KEY, tag to identify what is to be deleted. The key can be a name, a known catalog key, a database key, or a combination of these. For descriptors, this identity is required for the category that contains the descriptor, and for units, this identity is required for the descriptor that contains the unit.

```

<SCHEMA>

    <CATEGORY ACTION="DELETE">
        <NAME>Pen Gifts Sets</NAME>
    </CATEGORY>

    <CATEGORY ACTION="DELETE">
        <NAME>Pen Gift Collections</NAME>
        <KEY>3245</KEY>
    </CATEGORY>

    <CATEGORY ACTION="DELETE">

```

```

        <KEY>Laptop Computers</KEY>
    </CATEGORY>

    <DESCRIPTOR ACTION="DELETE">
        <OWNER>
            <NAME>Pens</NAME>
        </OWNER>
        <NAME>Barrel Color</NAME>
    </DESCRIPTOR>

    <DESCRIPTOR ACTION="DELETE">
        <OWNER>
            <KEY>22343</KEY>
        </OWNER>
        <NAME>Barrel Color</NAME>
        <KEY>887665</KEY>
        <LENGTH>500</LENGTH>
    </DESCRIPTOR>

    <UNIT ACTION="DELETE">
        <OWNER>
            <NAME> Product Shipping Weight </NAME>
        </OWNER>
        <NAME>Weight</NAME>
        <TYPE>Numeric</TYPE>
        <UNITELEMENT ACTION="DELETE">
            <NAME> Pounds </NAME>
        </UNITELEMENT>
        <UNITELEMENT ACTION="DELETE">
            <NAME> Gram </NAME>
        </UNITELEMENT>
    </UNIT>

</SCHEMA>

```

Examples: Update a Category, Descriptor, or Unit

Schema require a name, or key, tag to identify what is to be updated. There are also tags to tell the catalog if the descriptor is searchable or visible for display. These tags are needed when the catalog technology supports these specific features. They are ignored for less capable catalog engines.

```

<SCHEMA>
    <CATEGORY ACTION="UPDATE" >
        <NAME>Pen Gifts Set</NAME>
        <UPDATE>
            <NAME>Pen Gift Sets</NAME>
        </UPDATE>
    </CATEGORY>

    <CATEGORY ACTION="UPDATE">
        <KEY>C440911</KEY>
        <UPDATE>
            <NAME>Compressor Motors</NAME>
        </UPDATE>

```

```

</CATEGORY>
<CATEGORY ACTION="UPDATE">
  <KEY>C4490</KEY>
  <TYPE> TEMPLATE</TYPE>
  <UPDATE>
    <KEY>C7491</KEY>
    <TYPE>GENUS</TYPE>
  </UPDATE>
</CATEGORY>

<DESCRIPTOR ACTION="UPDATE">
  <OWNER>
    <NAME>Pens</NAME>
  </OWNER>
  <NAME>Ink Color</NAME>
  <SEARCHABLE>YES</SEARCHABLE>
  <VISIBILITY>NO</VISIBILITY>

  <UPDATE>
    <NAME>Ink Colors</NAME>
    <SEARCHABLE>NO</SEARCHABLE>
    <VISIBILITY>YES</VISIBILITY>
  </UPDATE>
</DESCRIPTOR>

<UNIT ACTION="UPDATE">
  <OWNER>
    <NAME>Price</NAME>
  </OWNER>
  <NAME> Local Price</NAME>

  <UNITELEMENT ACTION="UPDATE">
    <NAME> Franc</NAME>
  </UNITELEMENT>

  <UPDATE>
    <NAME> Euro Price</NAME>
  <UNITELEMENT ACTION="UPDATE">
    <NAME> Euro</NAME>
    <MULTIPLIER>3.21</MULTIPLIER>
    <ADDER> 5</ADDER>
  </UNITELEMENT>
  </UPDATE>
</UNIT>
</SCHEMA>

```

Data Item Management

Data items can be added to a category, deleted, and modified. All data items must be placed between the <DATA></DATA> tag. The identity feature (specifying a <NAME> and/or <KEY>) allows the wholesale change, or bulk updating, of values by effectively wildcarding updates. The update example below provides a method to change all items with a manufacturer Bic to Bic, Inc.

Example: Add Items

```

<DATA>
  <ITEM ACTION="ADD">
    <OWNER>
      <NAME>Pens</NAME>
    </OWNER>
    <NAMEVALUE>
      <NAME>Mfg Name</NAME>
      <VALUE>Bic</VALUE>
    </NAMEVALUE>
    <NAMEVALUE>
      <NAME>Price</NAME>
      <VALUE>1.55</VALUE>
      <UNITTYPE>Dollar</UNITTYPE>
    </NAMEVALUE>

    <NAMEVALUE>
      <NAME>Barrel Color</NAME>
      <VALUE>Blue</VALUE>
    </NAMEVALUE>
  </ITEM>
  <ITEM ACTION="ADD">
    <OWNER>
      <NAME>Pencils</NAME>
    </OWNER>
    <NAMEVALUE>
      <NAME>Mfg Name</NAME>
      <VALUE>Bic</VALUE>
    </NAMEVALUE>
  </ITEM>
</DATA>

```

For reference, the owner tag associates the item with a specific category. The name tag in the name/value pair associates the item with a descriptor and the value tag in the name/value pair fills the descriptor field with the value given. It is the use of the name/value pair that eliminates the need for catalog vendors or XML "standards" to "agree" on specific tags. The responsibility for the interchange is left to the catalog loader vendor and not the community of content providers, suppliers, and buyers.

Example: Delete Items

```

<DATA>
  <ITEM ACTION="DELETE">
    <OWNER>
      <NAME>Pens</NAME>
    </OWNER>
    <NAMEVALUE>
      <NAME>Mfg Name</NAME>
      <VALUE>Bic</VALUE>
    </NAMEVALUE> <!--removes all items by Bic mfg from Pens -->
    <NAMEVALUE>
      <NAME>Barrel Color</NAME>
      <VALUE>Blue</VALUE>
    </NAMEVALUE>
  </ITEM>
</DATA>

```

```

        </NAMEVALUE>

    </ITEM>
</DATA>

```

Example: Update Items

When updating an item, the owner tag is optional. Thus allowing for updates of multiple items with a specific value.

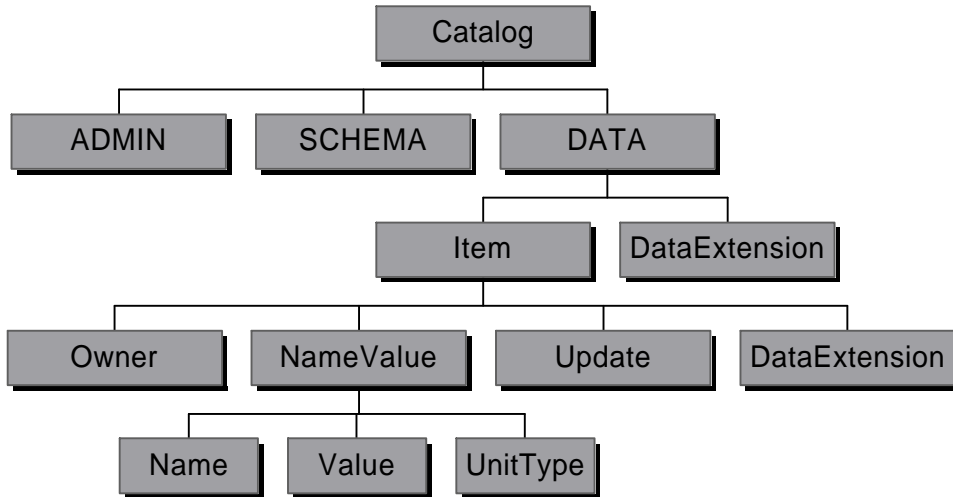
```

<DATA>
  <!-- Updating Item by Item -->
  <ITEM ACTION="UPDATE">
    <OWNER>
      <NAME>Pens</NAME>
    </OWNER>
    <NAMEVALUE>
      <NAME>Mfg Name</NAME>
      <VALUE>Bic </VALUE>
    </NAMEVALUE>
    <NAMEVALUE>
      <NAME>Sup Part Num</NAME>
      <VALUE>B145C7</VALUE>
    </NAMEVALUE>
    <UPDATE>
      <NAMEVALUE>
        <NAME>Mfg Name</NAME>
        <VALUE>Bic Inc.</VALUE>
      </NAMEVALUE>
      <NAMEVALUE>
        <NAME>Barrel Color</NAME>
        <VALUE>Red</VALUE>
      </NAMEVALUE>
    </UPDATE>
  </ITEM>
  <!-- This section below updates items with Mfg Name= Bic -->
  <ITEM ACTION="UPDATE">
    <OWNER><NAME>Pens</NAME> </OWNER>
    <NAMEVALUE>
      <NAME>Mfg Name</NAME>
      <VALUE>Bic </VALUE>
    </NAMEVALUE>
    <UPDATE>
      <NAMEVALUE>
        <NAME>Mfg Name</NAME>
        <VALUE>Bic Inc.</VALUE>
      </NAMEVALUE>
    </UPDATE>
  </ITEM>
</DATA>

```

The following diagram shows a hierarchical diagram of the Item tag.

eCatalog's Item tag in detail



Special Character Handling

Special characters and non-XML markup must be escaped for XML parsers and catalog loaders to function correctly. Specifically, the & and < characters must be escaped with a CDATA tag.

<![CDATA[**your data here**]]> inserted in any tag for special characters.

```

<SCHEMA>
  <CATEGORY ACTION="DELETE">
    <NAME><![CDATA[Pen & Pencil Gifts Sets]]></NAME>
  </CATEGORY>
</SCHEMA>
    
```

International Characters

This specification allows the setting of default languages by use of the xml:lang attribute on the CATALOG element. Multiple languages can be contained in this specification by the addition of xml:lang to the NAME and VALUE elements which contain the actual data to be described.

This language encoding follows the ISO standards 639, 3161 for codes and sub codes. This is based on the following language from the "Extensible Markup Language (XML) 1.0" W3C Recommendation 10-February-1998 (section 2-12). It states:

The intent declared with xml:lang is considered to apply to all attributes and content of the element where it is specified, unless overridden with an instance of xml:lang on another element within that content.

The following is a table from the XML 1.0 specification stating "how" language identification is specified.

Language Identification

LanguageID	::=	Langcode (- Subcode)*
Langcode	::=	ISO639Code IanaCode UserCode
ISO639Code	::=	([a-z] [A-Z]) ([a-z] [A-Z])

```
IanaCode ::= ('i' | 'I') '-' ([a-z] | [A-Z])+
UserCode ::= ('x' | 'X') '-' ([a-z] | [A-Z])+
SubCode ::= ([a-z] | [A-Z])+
```

Here's an example where language is specified for the entire catalog where the language = US English and the country is United States:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<CATALOG xml:lang="EN-US">
```

We recommend that you check with your XML parser and its documentation on how it handles international characters.

Data Extensions

There arises the need in catalog data exchange to provide a mechanism to support extensions and application-specific data input requirements. One approach would be to invent tags for each one of these options. This specification provides the mechanism by allowing a data extension tag that allows for application unique data. One example is products associations via a query linkage (i.e., batteries for a palm pilot). The tags for this extension are shown below:

Add DataExtension

```
<ITEM ACTION="ADD">
  <OWNER>
    <NAME>Pens</NAME>
  </OWNER>
  <NAMEVALUE>
    <NAME>Mfg Name</NAME>
    <VALUE>Bic</VALUE>
  </NAMEVALUE>

  <DATAEXTENSION>
    <TYPE>Association</TYPE>
    <DATAELEMENT ACTION="ADD">
      <NAMEVALUE>
        <NAME>TYPE</NAME>
        <VALUE>BugsEye</VALUE>
      </NAMEVALUE>
      <NAMEVALUE>
        <NAME>QUERY</NAME>
        <VALUE>Ink Refill</VALUE>
      </NAMEVALUE>
      <NAMEVALUE>
        <NAME>COLUMN_NAME</NAME>
        <VALUE>Accessories</VALUE>
      </NAMEVALUE>
      <NAMEVALUE>
        <NAME>GRAPHIC</NAME>
        <VALUE>access.gif</VALUE>
      </NAMEVALUE>

      <NAMEVALUE>
        <NAME>LABEL</NAME>
        <VALUE>Accesroies Text</VALUE>
      </NAMEVALUE>
    </DATAELEMENT>
  </DATAELEMENT ACTION="ADD">
    <NAMEVALUE>
```

```

                <NAME>TYPE</NAME>
                <VALUE>URL</VALUE>
            </NAMEVALUE>
            <NAMEVALUE>
                <NAME>QUERY</NAME>
                <VALUE>http://www.requisite.com/</VALUE>
            </NAMEVALUE>
            <NAMEVALUE>
                <NAME>LOCATION</NAME>
                <VALUE>Accessories</VALUE>
            </NAMEVALUE>
        </DATAELEMENT>
    </DATAEXTENSION>
    <DATAEXTENSION>
        <TYPE>Table</TYPE>
        <NAME>New Table</NAME>
        <DATAELEMENT ACTION="ADD">
            <NAMEVALUE>
                <NAME>Accessories</NAME>
                <VALUE>Batteries; Leather Case; 238776516</VALUE>
            </NAMEVALUE>
        </DATAELEMENT>
    </DATAEXTENSION>
</ITEM>

```

DataExtension types can be Table, Association, or other application specific types. For the BugsEye catalog, it is recommended to use at least three name/value pairs: TYPE, QUERY, and LABEL. If the data extension type is table, a NAME, or KEY, tag is required. If the data extension type is association, the NAME or KEY tag is not required.

The "TYPE" label can be BugsEye, URL, or other application specific types.

The "QUERY" label contains information based on the TYPE specified. For example, if the Type value specified was "BugsEye", the query value contains the string to match during the search. If the Type value was "URL", the query value can contain an actual URL, <http://www.requisite.com>.

The "Location" label is the field to add, or an existing field in the catalog.

The "Graphic" label is optional and allows the specification of a graphic to be displayed when the item is displayed.

Update DataExtension

```

<ITEM ACTION="UPDATE">
    <OWNER>
        <NAME>Pens</NAME>
    </OWNER>
    <NAMEVALUE>
        <NAME>Mfg Name</NAME>
        <VALUE>Bic</VALUE>
    </NAMEVALUE>
    <UPDATE>
    </UPDATE>
</DATAEXTENSION>

```

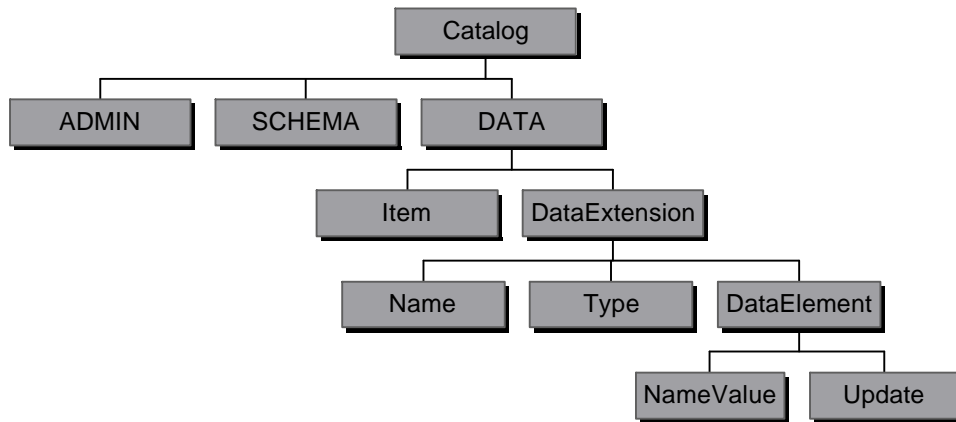
```

<TYPE>TABLE</TYPE>
<NAME> BugsEye_config </NAME>
<DATAELEMENT ACTION="UPDATE">
  <NAMEVALUE>
    <NAME>NAME</NAME>
    <VALUE>edit_config</VALUE>
  </NAMEVALUE>
  <NAMEVALUE>
    <NAME>VALUE</NAME>
    <VALUE>com.requisite.ecmanager.bugedit.EditConfigCommand</VA
    LUE>
  </NAMEVALUE>
  <NAMEVALUE>
    <NAME>CONFIG_NAME</NAME>
    <VALUE>ecm</VALUE>
  </NAMEVALUE>
  <UPDATE>
    <NAMEVALUE>
      <NAME>NAME</NAME>
      <VALUE>my_config</VALUE>
    </NAMEVALUE>
    <NAMEVALUE>
      <NAME>VALUE</NAME>
      <VALUE>com.requisite.MyConfig</VALUE>
    </NAMEVALUE>
    <NAMEVALUE>
      <NAME>CONFIG_NAME</NAME>
      <VALUE>my_ecm</VALUE>
    </NAMEVALUE>
  </UPDATE>
</DATAELEMENT>
<DATAELEMENT ACTION="UPDATE">
  <NAMEVALUE>
    <NAME>LOT Price</NAME>
    <VALUE>33.5</VALUE>
  </NAMEVALUE>
  <UPDATE>
    <NAMEVALUE>
      <NAME>Car LOT Price</NAME>
      <VALUE>$101.25</VALUE>
    </NAMEVALUE>
  </UPDATE>
</DATAELEMENT>
</DATAEXTENSION>
</UPDATE>
</ITEM>

```

The following diagram shows a hierarchical diagram of the DataExtension tag.

eCatalog's DataExtension tag in detail



Conclusion

The eCatalog XML specification is currently used by Requisite Technology and its partners to exchange catalog structure and electronic content. It is the intent of this XML specification to promote an open, catalog interoperability standard. The meta-data nature of this specification will allow eCatalog XML to incorporate the many and varied XML "standards" emerging in electronic commerce and provide true and complete catalog interoperability and content exchange.

Please direct all questions and comments to Requisite Technology.