xCBL 3.5 for Content:

ProductCatalog

Functional Specification

July 9, 2002

Contents

1   Introduction

Beginning with xCBL 3.0, xCBL for Content provided a method of handling arbitrary attributes on products in catalogs.   It provides a common, or *core,* set of product attributes that most catalog data formats support: product name and identifier fields, manufacturer information, vendor information, and price data, for example.   But rich catalog search capability requires the ability to represent many more attributes depending on the class or category of product.   For example a particular product category might contain: weight, length, size, color, hardness, and warranty length fields. This ability to have category-dependent attribution is fundamental to many electronic catalog search technologies including Commerce One's Content Engine (the catalog in Commerce One's Net MarketMaker and MarketSite products), Requisite Software's BugsEye product, and A2i's xCat content management and catalog publishing system.

In addition to allowing additional attributes on products, the XML format for representing catalog information should allow a method to define rich validating schemas: hierarchical taxonomies of categories and the attributes available on each category.   Individual products transmitted with extended attributes should be able to have those attributes validated against the defined schema, given a category selected for the product. These two primary requirements drove the creation of the xCBL 3.0 schema for product and price catalog information.

Commerce One and SAP have continued to work together to define an updated xCBL document that address these needs, provides a more powerful pricing model, and addresses some inadequacies of 3.0 – xCBL 3.5 ProductCatalog.   The goal of this specification is to describe the purpose, meaning, and optimal usage of the elements of xCBL3 catalog documents. We first describe the overall design goals for providing catalog information in xCBL 3.5.   This is followed by some annotated sample xCBL product documents. Next we present a detailed breakdown of each the elements present in xCBL 3.0 for content. The next section separately describes how product information and prices are updated.   This is followed by formal XML schema definitions in DTD, XDR, and SOX forms. These schemas both validate instances of these documents and help to document the structure. Finally, to ease implementation and conversion from existing catalog formats, we describe mappings between the xCBL 3.0 content documents and other popular catalog content data formats. These formats currently include Requisite Technology's eCX XML format, Commerce One's xCBL 2.0, SAP's XML for content, EDI X12 832, and RosettaNet's PIP2A1for product resource update document.

1.1   Changes to xCBL ProductCatalog in xCBL 3.5

The xCBL 3.5 introduces substantive changes to ProductCatalog. A new Product sub-element (ExternalItemRef) enables one to reference an item defined in another catalog (see Section 1.2 Referencing Product Definitions from Other Catalogs). Other catalog types that allow for introducing different kinds of information about catalog items for different uses will likely be introduced later.

1.2   Referencing Product Definitions from Other Catalogs

A catalog is a collection of information about a set of items. There may be more than one collection of information about the same set of items. For example, there may be information that relates how a set of items fits into a taxonomy, including values for attributes specific to that taxonomy. Another collection may be additional information specific to a particular user.

To manage the many different types of information that may be associated with a set of items, the ability to have separate catalogs for each type of information that can reference definitions in other catalogs has been introduced (see ExternalItemRef). Examples of this usage are included in the last section of this document.

Note!! Circular references and recursive references are undefined, and should be considered an error.

2   An overview of xCBL 3.5 ProductCatalog

2.1   Changes for xCBL ProductCatalog 3.5

Commerce One and SAP have continued to collaborate on the specification of xCBL ProductCatalog. This document specifies changes to ProductCatalog which are backwards compatible with documents created using 3.0.

There are three principle areas of change.
- One is to align ProductCatalog syntax with that adopted by the remainder of the xCBL language, wherever possible, in particular deprecating the use of XML ID and XML IDREF(S). XML ID/IDREF(S) are defined as *document* specific, and hence do not persist. Yet their use in xCBL 3.0 contradicts this – for example, SchemaCaterory has only one piece of required information, an XML ID (the CategoryID XML attribute), hence the XML ID would be the only logical key to uniquely identify the SchemaCategory.
- Second is to provide a richer structures for describing price information, which better match the needs of catalog providers, and thus simplify their catalog price specification and maintenance tasks: PricingInformation and SupplierAccountInformation.
- Finally, a new sub-element of CatalogData is introduced which can be used in addition to the "Product" sub-element, namely "Pricing". It facilitates differentiation of two kinds of the information which are confounded in the original "Product" element: information describing the offering, and information describing the pricing of that offering (the ProductPrice).

This final change greatly facilitates users performance of catalog maintainance tasks as it significantly simplifies the update semantics. With the introduction of this improved pricing model and features to simplify the update semantics, we deprecate the use (in combination) of (a) ProductPrice at the Product level and (b) IsPriceUpdate at the CatalogHeader level: IsPriceUpdate only has meaning with ProductPrice used at the Product level, and it is possible to express and maintain the same price information as in 3.0 without using these features.

This document also contains changes which aren't changes to the language itself, but instead provide guidance to catalog authors as to the kinds of values expected for certain language elements.

2.2   Design Goals

xCBL 3.0 introduced two major capabilities to the communication of catalog information: it lets catalog creators define extended attributes for products, and it allows catalog providers to transmit hierarchical taxonomies of categories.   These two capabilities are related, since part of the purpose of providing the list of categories is to define the attributes that can or should be available for a given category.

2.3   Base Schema

xCBL 3.0 for Content defines a base set of attributes for all products.   It also provides the capability to extend the structure by allowing a catalog provider to define attributes on products in addition to those anticipated in the xCBL definition.   The only required subelement of **Product** is the **ProductID** element, which is assumed to be unique with the document. **ProductID** is required because it is necessary for many operations such as updates of catalogs and references from other products. It may be necessary for the catalog supplier to programmatically generate these unique **ProductID**s. This may often be the case. For example, a multi-supplier catalog provider might append the supplier part number and the supplier name

together to build the unique **ProductID**.   Another catalog supplier might append the manufacturer name, manufacturer part number, and unit of measure to create the **ProductID**.

Other subelements of **Product** (such as **ProductName**, short and long description, manufacturer, dates of validity, prices, per vendor information, attachments, and related product information) are *all optional.* This is to allow xCBL 3.0 to incorporate the schema used by any catalog supplier.   None of these fields are required, but creators of the xCBL catalog documents should make every effort to use one of the existing fields (as opposed to creating **ObjectAttribute**s as described in the next subsection).   Receiving catalog applications and other applications that use catalog information (such as downstream procurement applications) can perform richer business logic with the information *if they know exactly what the supplied product attributes mean*.

2.4   Extensibility

Whenever a product catalog has additional information that is not present in the set of subelements provided in the xCBL Product element, the catalog creator can and should create an **ObjectAttribute** subelement, which contains a name/value pair to represent the extended attribute.   In principle any information that is available by the catalog supplier can be added as an **ObjectAttribute**.   So the simplest mapping of a supplier's catalog to xCBL3 can be to just make all product fields in the catalog into **ObjectAttribute** subelements of the same name in the xCBL document. But as noted above, the "built-in" subelements of Product should be used wherever possible.

2.5   Schemas Representing Taxonomies

xCBL 3.0 also allows a catalog schema of categories and their associated attributes to be sent along with the catalog product information. A catalog schema defines a taxonomy of catalog categories and the attributes that are allowed (or required) for each of those categories.   Catalog schemas are not required to be sent in xCBL 3.0 documents, but they are a useful way of communicating the structure of various classes of products. In fact, catalog documents may be sent with just a catalog schema listing valid categories and their associated attributes. This might be sent by a marketplace to their suppliers. The supplier can then send its own catalog document, including the catalog schema document provided by a marketplace. Catalog schemas sent in product catalog documents by suppliers can reference an external URL that contains the schema. For example, the catalog schema defined by, and hosted on, a marketplace.

2.6   Leverage the Rest of xCBL

The set of "xCBL 3.0 for content documents" are really just the **ProductCatalog** element and its full set of subelements.   Some of these subelements (such as the **Party** and **UOM** elements) are defined elsewhere in xCBL, and are just used within the catalog document. This avoids redefining some complex elements, such as with the **Party** element, and stops the catalog documents from repeating maintaining large enumerations of valid values, such as with the **UOM** element. The **DefaultCurrency** element and the **ProductPrice** element both contain the **Currency** element from elsewhere in xCBL 3.0. The **CountryOfOrigin** element uses the **Country** element from the rest of xCBL 3.0. For the most part, however, the product catalog-related documents are separate and described here.

2.7   Easily Understandable Top Level Structure

The **ProductCatalog** element is divided into three main sections:
- the **CatalogHeader** element which contains a variety of administrative information, including the list of parties that are involved with this catalog. This may be a set of buyers and suppliers, that can then be referenced from individual products later in the document.

- the optional **CatalogSchema** element, which contains the set of valid categories of products and the attributes of those products. The **CatalogSchema** element may use the **SchemaSource** subelement to refer to another xCBL 3.0 document containing the actual schema. The
- a **CatalogData** element (optional since sometimes one may wish only to communicate schema information) which contains the actual set of products in the catalog

2.8   Handle Internationalized Content

xCBL 3.0 for catalog content was also designed from the beginning to support internationalized content. That is, it can transmit information about products in several languages. Several elements, including all description and name elements, contain an **xml:lang** attribute which is used to describe the language contained. The values inside **xml:lang** must be valid RFC 1766 language (and optional locale) codes as documented at http://www.ietf.org/rfc/rfc1766.txt?number=1766 . RFC1766 currently defines use of ISO standard 639 two letter codes for languages and optional presence of ISO 3166 abbreviations for country codes. Doublebyte character language content inside the internationalized elements can be expressed using standard HTML/XML escape sequences (e.g. #nnn;), where "nnn" is interpreted as a Unicode character.In addition, one may use the "encoding" attribute in the XML declaration of the XML prolog to declare that the entire contents of the document are expressed using a recognized character set (per the XML 1.0 specification).

3   A Sample xCBL 3.5 Content Document

Now that we've described the overall approach of the xCBL 3.5 **ProductCatalog** element, let's take a look at an example document. The first portion of this document is the **CatalogHeader** section. It contains a variety of administrative information: a unique catalog ID, the date of creation of the catalog, and a list of partners to whom this catalog is relevant. In the example below the partners include the buyer, supplier and manufacturer. Each of these partners can then be referred to later in the document. For example per-buyer prices can reference the **Partner** element from the **CatalogHeader**.

This example also stands in xCBL 3.5 ProductCatalog. It's been annotated in a couple of important places where an optional element has been changed to a mandatory one, and for how to use the alternative to XML ID/IDREF(S):

```
<ProductCatalog>
<CatalogHeader>
       <CatalogID>Acme Laptops Catalog</CatalogID>
       <CatalogDate>20000816</CatalogDate>
<!—in xCBL 3.5, CatalogProvider is a required element 
<CatalogProvider>
<Party>
<PartyID>
       <Identifier>
             <Agency>
                    <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
             </Agency>
             <Ident>f94be98e-77b6-1000-92af-ac1408350001</Ident>
       </Identifier>
</PartyID>
</Party>
</CatalogProvider>
       <ListOfPartners>
             <!—Partner's been changed to include an optional PartnerRelationship 
             <!—this enables specifying more than one role for a partner. Also, 
             <!—PartnerID is deprecated in favor of using Party 
             <Partner >
                <Party>
                   <PartyID>
                          <Identifier>
                                <Agency>
```

```
                                    <AgencyCoded>AssigneByMarketPlace</AgencyCod
                            ed>
                            </Agency>
                            <Ident>f94be98e-77b6-1000-92af-ac1408350002</Ident>
                        </Identifier>
                    </PartyID>
                </Party>
                <PartnerRelationship>Buyer</PartnerRelationship>
            </Partner>
            <Partner >
                <Party>
                    <PartyID>
                            <Identifier>
                                    <Agency>
                                            <AgencyCoded>AssignedByMarketPlace
                                            </AgencyCoded>
                                    </Agency>
                                    <Ident>f94be98e-77b6-1000-92af-ac1408350010</Ident>
                            </Identifier>
                    </PartyID>
                </Party>
                <PartnerRelationship>Manufacturer</PartnerRelationship>
            </Partner>
            <Partner >
                <Party>
                    <PartyID>
                            <Identifier>
                                    <Agency>
                                            <AgencyCoded>AssigneByMarketPlace</AgencyCod
                                    ed>
                                    </Agency>
                                    <Ident>f94be98e-77b6-1000-92af-ac1408350001</Ident>
                            </Identifier>
                    </PartyID>
                </Party>
                <PartnerRelationship>Supplier</PartnerRelationship>
            </Partner>
        </ListOfPartners>
        <ValidFrom>20000816</ValidFrom>
        <ValidUntil>20001231</ValidUntil>
        <CatalogVersion>1.0</CatalogVersion>
        <DefaultLanguage xml:lang="en"/>
        <DefaultCurrency>
                <Currency><CurrencyCoded>USD</CurrencyCoded>
                </Currency>
        </DefaultCurrency>
        <IsMultiVendor/>
        <ShortDescription xml:lang="en">This is Acme Laptops&apos;s catalog of Compaq
        Laptops, for General Motors. </ShortDescription>
</CatalogHeader>
```

The optional CatalogSchema element lists all of the valid categories of products. The optional SchemaStandard subelement of CatalogSchema lists which product catalog taxonomy that this particular schema may be a subset of or based upon. Since the CatalogData element is itself optional, the CatalogSchema element *can* be used alone to communicate taxonomies, such as UNSPSC or eClass. In the example below the catalog schema is subsetted to just have a few computer-related categories.

Note that using the CatalogID element means that CategoryID values aren't XML ID's and don't have to obey the XML ID syntactic restrictions. This also means that the CategoryName can be used for the printing name, rather than the ShortDescription (as was done in the previous version of this example).

```
<CatalogSchema>
<SchemaName>Acme&apos;s UNSPSC Subset for Computers</SchemaName>
<SchemaVersion>1.0</SchemaVersion>
<SchemaStandard>UNSPSC</SchemaStandard>
<!—XML ID use is deprecated →
<SchemaCategory >
<!—Since CategoryID element is not an XML ID, →
```

```
<!-- it doesn't need to obey XML name production →
<CategoryID>43000000</CategoryID>
<CategoryName> Communications, Computer Equipment, Peripherals, Components and Supplies
</CategoryName>
            </SchemaCategory>
        <SchemaCategory >
        <CategoryID>43170000</CategoryID>
        <ParentCategoryRefList>
                <CategoryIDRef>43000000</CategoryIDRef>
        </ParentCategoryRefList>
            <CategoryName> Hardware and accessories </CategoryName>

        </SchemaCategory>
        <SchemaCategory >
                <CategoryID>43171800</CategoryID>
                <ParentCategoryRefList>
                        <CategoryIDRef>43170000</CategoryIDRef>
                </ParentCategoryRefList>
                <CategoryName> Computers </CategoryName>

                <CategoryAttribute>
                        <AttributeID>Processor Speed</AttributeID>
                        <AttributeName xml:lang="en">Processor Speed</AttributeName>
                        <AttributeType ScalarType="String"></AttributeType>
                </CategoryAttribute>
        </SchemaCategory>
        <SchemaCategory >
                <CategoryID>43171801</CategoryID>
                <ParentCategoryRefList>
                        <CategoryIDRef>43171800</CategoryIDRef>
                </ParentCategoryRefList>
                <CategoryName> Notebook computers </CategoryName>

                <CategoryAttribute>
                        <AttributeID>Battery Life</AttributeID>
                        <AttributeName xml:lang="en">Battery Life</AttributeName>
                        <AttributeType ScalarType="String"></AttributeType>
                </CategoryAttribute>
        </SchemaCategory>
        <SchemaCategory >                   <CategoryID>43171803</CategoryID>
                <ParentCategoryRefList>
                        <CategoryIDRef>43171800</CategoryIDRef>
                </ParentCategoryRefList>
                <CategoryName> Workstations, desktop computers </CategoryName>

        </SchemaCategory>
</CatalogSchema>
```

The CatalogData element lists requests to add, update or delete products to the catalog, via the Action tag.
Detailed semantics of the catalog update process are presented in section 5.

```
<CatalogData>
        <Product>
                <Action Value="Delete"/>
                <ProductID>118003-008</ProductID>
        </Product>
        <!—use of SchemaCategoryRef is deprecated →
        <Product Type="Good">
                <ProductID>140141-002</ProductID>
                <SchemaCategoryRefList>
                        <CategoryIDRef>43171801</CategoryIDRef>
                </SchemaCategoryRefList>
                <UOM><UOMCoded>EA</UOMCoded></UOM>
                <Manufacturer>Compaq</Manufacturer>
                <LeadTime>2</LeadTime>
                <CountryOfOrigin>
                        <Country><CountryCoded>US</CountryCoded></Country>
                </CountryOfOrigin>
                <ShortDescription xml:lang="en">
                        Armada M700 PIII 500 12GB
```

```
        </ShortDescription>
        <LongDescription xml:lang="en">
        This light, thin powerhouse delivers no-compromise performance in a sub-
        five pound form factor. Size and Weight(HxWxD):    12.4 X 9.8 X 1.1 in
        4.3 - 4.9 lbs (depending on configuration) Processor:   500-MHZ Intel
        Pentium III Processor with 256K   integrated cache Memory:   128MB of
        RAM, expandable to 576MB  Hard Drive:    12.0GB Removable SMART Hard
        Drive  Display Graphics:    14.1-inch color TFT with 1024 x 768 resolution
        (up to 16M colors internal) Communication:     Mini-PCI V.90 Modem/Nic
        Combo   Operating System:    Dual Installation of Microsoft Windows 95
        &amp; Microsoft Windows 98
        </LongDescription>
        <ProductAttachment>
                <AttachmentURL>file:\5931.jpg</AttachmentURL>
                <AttachmentPurpose>PicName</AttachmentPurpose>
                <AttachmentMIMEType>jpg</AttachmentMIMEType>
        </ProductAttachment>
        <ObjectAttribute>
        <AttributeID> Processor Speed</AttributeID>
        <AttributeValue>500MHZ</AttributeValue>
        </ObjectAttribute>
        <ObjectAttribute>
        <AttributeID>Battery Life</AttributeID>
        <AttributeValue>6 hours</AttributeValue>
        </ObjectAttribute>
</Product>
<Product Type="Good ">
        <ProductID>154723-005</ProductID>
        <SchemaCategoryRefList>
                <CategoryIDRef>43171803</CategoryIDRef>
        </SchemaCategoryRefList>
        <UOM><UOMCoded>EA</UOMCoded></UOM>
        <!—depricate use of XML IDREF, using PCDATA here is preferred →
        <Manufacturer>f94be98e-77b6-1000-92af-ac1408350010</Manufacturer>
        <LeadTime>5</LeadTime>
        <CountryOfOrigin>
                <Country><CountryCoded>US</CountryCoded></Country>
        </CountryOfOrigin>
        <ShortDescription xml:lang="en">
                Deskpro EN SFF Pentium III 600MHz/10.0GB/64MB/24X/NIC/NT 4.0
        </ShortDescription>
        <LongDescription xml:lang="en">
        Pentium® III 600 MHZ deskpro system with a 10GB hard drive, running
        Windows® NT 4.0       Pentium III Processor, 512K Secondary Cache, 64MB,
        Bays/Slots 3x2, Intelligent Manageability, ATI RAGE PRO TURBO AGP 2X
        Graphics, SMART II Ultra ATA Hard Drive, Compaq PremierSound using ESS
        1869, MS Windows NT 4.0, Mouse, Compaq 10/100 TX PCI Intel WOL UTP, 24X
        Slimline CD ROM
        </LongDescription>
        <ProductVendorData>
                <VendorIDRef>f94be98e-77b6-1000-92af-ac1408350001</VendorIDRef>
                <VendorPartNumber>12345</VendorPartNumber>
                <ProductPrice>
                        <Amount>1000</Amount>
                        <Currency><CurrencyCoded>USD</CurrencyCoded></Currency>
                        <UOM><UOMCoded>EA</UOMCoded></UOM>
                </ProductPrice>
        </ProductVendorData>
        <ProductAttachment>
                <AttachmentURL>file:\5922.jpg</AttachmentURL>
                <AttachmentPurpose>PicName</AttachmentPurpose>
                <AttachmentMIMEType>jpg</AttachmentMIMEType>
        </ProductAttachment>
        <ObjectAttribute>
        <AttributeID>Processor Speed</AttributeID>
        <AttributeValue>600MHZ</AttributeValue>
        </ObjectAttribute>
</Product>
</CatalogData>

</ProductCatalog>
```

4   Document Structure

The previous example should give you an idea of what information is available in an example xCBL 3.0 catalog, but it is not meant to be exhaustive.   This section breaks down all of the available catalog subelements, their meaning, and their range of values.

4.1   The ProductCatalog Document

All of these capabilities are provided within the context of a **ProductCatalog** document. Unlike xCBL 2.0, xCBL 3.0 provides only one top level document for both product and price-related catalog updates – the **ProductCatalog** document.   The **ProductCatalog** document consists of a **CatalogHeader** element containing administrative information, an optional **CatalogSchema** element, an optional **CatalogData** element.

ProductCatalog Subelements	Description	Cardinality
CatalogHeader	Administrative information about the catalog including its provider, and suppliers and other trading partners present in the catalog	1..1
CatalogSchema	Optional information about the structure of products in the catalog	0..1
CatalogData	Container for instances of products and pricings	0..1

4.2   The CatalogHeader Element

Each **CatalogHeader** starts with a unique **CatalogID** (presumed to be unique within the space of catalogs supplied by a given **CatalogProvider**).   There is also optionally a **CatalogDate**, that identifies the date of creation of this information. It is logically equivalent to a file date/time stamp, and it is necessary since the catalog XML may be transmitted over the wire.   Next is a required **CatalogProvider** element that wraps a standard xCBL **Party** element to identify the organization name and contact information of the providing catalog party.

There is also an optional **ListOfPartners** element which lists suppliers, information providers, and manufacturers providing content in this document, as well as identified buyer consumers of this catalog. Each **Partner** element in the **ListOfPartners** has a **Partner** element. The **Partner** element contains a standard xCBL 3.0 **Party** subelement, which is used to uniquely identify the Partner, and an optional PartnerRelationship subelement, which is used to used to specify the type of relationships applicable to a partner.

The optional **ValidFrom** and **ValidUntil** elements identify the time period during which the catalog applies. **CatalogVersion** presents a version identifier for the catalog, which should ideally be specified in sequentially increasing order to take advantage of potential syndication services, which manage distribution of differences between successive versions of catalogs.   **DefaultLanguage** and **DefaultCurrency** identify what language and currency to be assumed *if these are not specified on an individual product.* **ShortDescription** and **LongDescription** identify a language locale and can be present multiple times to facilitate translation of description content. The optional presence of **ObjectAttribute** allows representation of aspects of the overall catalog or catalog administration that we did not anticipate.

CatalogHeader Subelements	Description	Cardinality
CatalogID	Required unique identifier for catalog. Must be unique within the set of catalogs provided by this catalog provider.	1..1
CatalogDate	Date that this catalog information was created. Needs to be in the format used by all of xCBL; *yyyymmdd* (note the lack of slashes)	0..1

CatalogHeader Subelements	Description	Cardinality
CatalogProvider	Providing party of this catalog. Contains an optional **Party** subelement. Has optional attribute of **ProviderID** whose ID overrides the ID present on a **Party** element. Either a **ProviderID** attribute or **Party** subelement must be present.	1..1
CatalogPrettyName	A name for display purposes	0..1
CatalogLogoURL	An image which could be used with the display of the catalog	0..1
ListOfPartners	Identified suppliers, manufacturers used by this catalog and potentially buyers that this catalog is targeted to. Contains one or more **Partner** elements. Each **Partner** element contains a **Party** subelement. The definition of the **ListOfPartners** subelement is presented in the table below.	0..1
CatalogAudience	Has an attribute (CatalogAudienceCoded) which is essentially a binary flag, values restricted to either "Public" or "EnumeratedBuyersOnly": is access to information defined as public (i.e., price catalogs associated with an IsPublicAccount) restricted to only the Partner's of type Buyer listed in the ListOfPartners (EnumeratedBuyersOnly); or is access unrestricted (Public). Default is Public.	0..1
PricingInformation	A list of PriceCatalog definitions (see sub-element tables for definition) NOTE – without mention in a SupplierAccount, the audience for prices in a PriceCatalog is undefined.	0..1
SupplierAccountInformation	Supplier can provide a within-catalog unique name for an account associated with a particular Buyer, that has a specific pricing rule associated with it. This supports supplier/buyer relationships that may have more than one pricing agreement as well as those suppliers which only have a single pricing agreement with each buyer.	0..1
ValidFrom	Date that products and price in this catalog begin to be available. Needs to be in the format used by all of xCBL; *yyyymmdd* (note the lack of slashes)	0..1
ValidUntil	Date that catalog stops applying. Needs to be in the format used by all of xCBL; *yyyymmdd* (note the lack of slashes)	0..1
CatalogVersion	Version number of catalog specified as single integer. Should be specified as a higher number than any previous catalog updates that have been sent.	0..1
DefaultLanguage	Contains attribute *xml:lang* indicating the language that should be assumed for data elements in this catalog document (and subsequent documents targeting this catalog, if it is not respecified) when it is not specified explicitly on a particular element. The value of *xml:lang* must be a valid RFC 1766 language code.	0..1

CatalogHeader Subelements	Description	Cardinality
DefaultCurrency	ISO currency code indicating the currency that should be assumed in this catalog document (and subsequent documents targeting this catalog, if it is not respecified). Contains a *Currency* element from the rest of xCBL, which itself contains a *CurrencyCoded* and a *CurrencyCodedOther* element. Only *CurrencyCoded* is required to appear in the *Currency* element.	0..1
IsReplacement	Indicates that this is a complete replacement for an existing catalog's Schema and Products – Pricing's are unaffected by this flag (see the PriceCatalog Action sub-element for how to effect bulk deletions of Pricings). The independence between Product and Pricing replacement is a feature enabling a new set of Product information to be added without needing to restate all the pricing information.	0..1
IsPriceUpdate	Indicates that the catalog is only being used to update prices. The use of this flag only affects prices created using ProductPrices as a sub-element of Product. Use of this flag is deprecated (see 2.1 Changes for xCBL ProductCatalog 3.5)	0..1
IsMultiVendor	Indicates that the catalog is a multivendor catalog. If a product is multivendor, price should only appear under ProductVendorData.	0..1
ShortDescription	Short description of the catalog. Has *xml:lang* attribute so there can be more than one ShortDescription in multiple languages. *xml:lang* must contain valid RFC 1766 language code.	0..n
LongDescription	Long description of the catalog. Has xml:lang attribute for multilanguage support. *xml:lang* must contain valid RFC 1766 language code. DescriptionPurpose is an attribute that attempts to give describe what the description is for. Examples of valid values are "sales" and "internal use only".	0..n
ObjectAttribute	Other aspects of the overall catalog or catalog administration that are not in subelements listed above	0..n

ListOfPartners Subelements	Description	Cardinality
Partner	**Partner** has an optional **PartnerID** attribute that is an XML ID. *Use of any XML ID or IDREF(S) is deprectated, including PartnerID.* Although the **PartnerID** attribute is optional, an ID must be present as either the **PartnerID** or in the **Party** subelement. The ID in the **Party** subelement takes precedence over the **PartnerID**. The **Partner** subelement definition is shown in the table below.	1..n

Partner Subelements	Description	Cardinality
Action	See update semantics for Partner. Takes one of four values Add, Update, Replace, or Delete.	0..1

Partner Subelements	Description	Cardinality
Party	The **Party** element is not strictly part of the catalog content portion of xCBL 3.0 so its definition is not presented here. However, section 6.1.1 includes the DTD for the **Party** element for convenience. Note that the ID present in the Party subelement takes precedence over the Partner ID.	0..1
PartnerRelationship	One of the following values: Supplier, SupplierAgent, Buyer, InfoProvider, Manufacturer, and CatalogOwner. Defaults to Supplier	1..n

SupplierAccountInformation Subelements	Description	Cardinality
SupplierAccount	A pair linking a supplier's id for an account that a buyer may purchase against, and the buyer's TPID (see SupplierAccount sub-element table).	1..n

SupplierAccount Subelements	Description	Cardinality
Action	Add and Delete Action values are acceptable and no Action value is implicitly an Add. The extent of the update semantics is restricted to membership in this list.	0..1
SupplierAccountID	A catalog-unique ID. An identifier for referencing to the prices defined under a particular purchasing agreement or contract. NOTE: the ID is the key so there is only one association between a particular Buyer and a particular SupplierAccountID. NOTE: if the ID is not specified, it defaults to the "NULL" id, which is also considered a unique ID (subsequent SupplierAccount entries with no ID are assumed to reference the same SupplierAccount, even though it is nameless). The "null" id was allowed primarily to accommodate the declaration of a PriceCatalog set as being the one available to the anyone, as opposed to those available via a negogiated agreement between the supplier and a particular buyer, in which case a supplier account id would be established.	0..1
BuyerIdentifier OR IsPublicAccount	Either of these two elements: BuyerIdentifier is an element of type Identifier referring to a buyer. For a catalog to be rendered usefully by a Commerce One catalog engine, the Ident sub-element should be a TPID; IsPublicAccount is a empty element meaning that all prices in the PriceCatalog(s) with PriceCatalogIDRef(s) are available to the public.	1..1
PriceCatalogIDRef	Reference to a PriceCatalog defined in PricingInformation	0..n

PricingInformation Subelements	Description	Cardinality

PriceCatalog	When used in conjunction with the Pricing element in Product, this provides a means of naming a set of ProductPrices.	1..n

PriceCatalog Subelements	Description	Cardinality
Action	The values Add, Replace, Update, and Delete additionally apply to all the Pricing elements whose PriceCatalogIDRef matches this PriceCatalogID	0..1
PriceCatalogID	A catalog-unique ID to refer to a set of ProductPrices denoted by grouping them in a Pricing element with a PriceCatalogIDRef matching this one. NOTE: PriceCatalog's have no established audience unless SupplierAccount is defined which references it.	1..1
ValidFrom	Date from which the Pricings in this catalog are effective – overrides individual ProductPrice ValidFrom dates if and only if they are *earlier* than this date. Needs to be in the format used by all of xCBL; *yyyymmdd* (note the lack of slashes)	0..1
ValidUntil	Date until which the Pricings in this catalog are effective – overrides individual ProductPrice ValidFrom dates if and only if they are *later* than this date. Needs to be in the format used by all of xCBL; *yyyymmdd* (note the lack of slashes)	0..1

4.3 The CatalogSchema Element

The CatalogSchema is a set of categories used for products in this catalog. It has a Type attribute that indicates whether the schema was created by a buyer, a supplier or "other" (such as a marketplace or manufacturer).

CatalogSchema Subelements	Description	Cardinality
SchemaName	Name of the schema.	1..1
SchemaVersion	Version number of the schema.	0..1
SchemaStandard	Taxonomy standard that this catalog may be based upon or a subset of, e.g. UNSPSC or eClass.	0..1
ValidateAttributes	Indicates that the category attributes should be validated.	0..1
ShortDescription	Short description of the schema. There may be more than one *ShortDescription* element to allow for internationalized content, which is effect by use of the *xml:lang* attribute. *xml:lang* must contain a value that is a valid RFC 1766 language code.	0..n
LongDescription	Long description of the schema. There may be more than one *ShortDescription* element to allow for internationalized content, which is effect by use of the *xml:lang* attribute. *xml:lang* must contain a value that is a valid RFC 1766 language code.	0..n

SchemaSource	URL to external schema. This external schema should either be a ProductCatalog document containing a schema or a CatalogSchema document. Use of an external SchemaSource is necessary to validate Product instances when ExternalCategory is used on a Product in lieu of a SchemaCategoryRef on Product. *Note: If this element is present in the CatalogSchema, no SchemaCategory's should be present in the CatalogSchema, and SchemaCategoryRef should not be used on any of the Products in the document.* This element is useful when a supplier or other catalog provider is providing data that needs to be validated against someone else's (such as the net market maker's) schema.	0..1
SchemaURN	URN identifying the Schema defined in this document	0..1
ExtensionToSchemaURN	URN to Schema for which definitions in this catalog are monotonic additions	0..1
SchemaCategory	List of categories. Should not be present if there is a SchemaSource element supplied to an external schema.	0..n

SchemaCategory Subelements	Description	Cardinality
CategoryID	A schema-unique ID for the category (alternative for deprecated use of CategoryID XML ID attribute).	0..1
ParentCategoryRefList	A list of CategoryIDRefs for SchemaCategories which are proper superclasses of this category, and from which this category inherits attributes (alternative for deprecated use of ParentCategoryRef XML IDREFS attribute). No value for this implies that this category is a sub-category of Product. Hence, ultimately, every category is a sub-category of Product. NOTE: this is a multiple inheritance model. The order of the categories establishes precedence, the list being ordered in descending precedence. This is most important in determing which class an undeclared ObjectAttribute used in the CatalogData section is added to.	0..1
CategoryName	Name(s) of the category.	0..n
ShortDescription	Short description(s) of the category.	0..n
LongDescription	Long description(s) of the category.	0..n
CategoryAttribute	Attribute list definition.	0..n

ParentCategoryRefList Subelements	Description	Cardinality
CategoryIDRef	Having the same value as the CategoryID of a SchemaCategory implies that SchemaCategory is a proper superclass of this one, and from which this category inherits attributes.	1..n

CategoryAttribute Subelements	Description	Cardinality
AttributeID	Identifier of the attribute. Unique among attributes present in this particular SchemaCategory. If a child SchemaCategory has the same AttributeID as a parent SchemaCategory, the suggested behavior of an xCBL 3.0 catalog processor is to report this as an error.	1..1
AttributeName	Name of the attribute. Multilanguage support is performed by having multiple AttributeName elements, each of which has an xml:lang attribute	0..n
AttributeType	Data type of the attribute.

Has ScalarType enumerated attribute that indicates whether this data value is a String, an Integer, a Numeric value (such as a floating point number), Currency, a Date, or an Enumeration. If an Enumeration, AttributeType has a set of subelements called EnumeratedValue which lists the values of the enumeration.

Also optionally has MAXSIZE attribute which indicates the maximum length of the string. | 1..1 |
| **DefaultUOM** | A UOM (as defined in Product, for example) which is the default unit of measure for this attribute. Defaults to EA (each). It also implies the dimensionality of the attribute (e.g., length, mass, time, etc). | 0..1 |
| **IsRequired** | Indicates that this attribute is required. | 0..1 |

4.4 The CatalogData Element

CatalogData Subelements	Description	Cardinality
Product	Contains all attributes associated with a product.	0..n
Pricing	Contains all prices associated with a particular Product from a particular PriceCatalog.	0..n

Product has an attribute called **SchemaCategoryRef** that is a link (IDREF) to a **SchemaCategory** in the **CatalogSchema**. Since the link is an IDREF, the SchemaCategoryRef attribute will only work if a schema has been supplied with this document. *Use of the SchemaCategoryRef is deprecated – the same functionality is available using the SchemaCategoryRefList subelement (see the next table).*

Product Subelements	Description	Cardinality
Action	The Value attribute indicates the action to be taken: Add, Update, Replace, or Delete. When no Action is present a default is assumed. See the section on catalog update semantics for details.	0..1
ProductID	Unique identifier of the product.

The optional Type attribute indicates whether the ProductID was created by a buyer, a supplier or "other" (such as a marketplace). | 1..1 |

BaseProductNumber	The value to return as the Product identifier. If not supplied, defaults to the value of ProductID. This is only useful to suppliers which use a double or triple of values to uniquely identify a product – the ID, an IDExtension, and a Unit of Measure. Suppliers which have a single unique ID for each and every product they offer can ignore this.	0..1
SchemaCategoryRefList	List of the SchemaCategories of which this Product is an instance (alternative to deprecated use of SchemaCategoryRef XML IDREFS attribute of Product). NOTE: this is a multiple inheritance model. The order of the categories establishes precedence, the list being ordered in descending precedence. This is most important in determing which class an undeclared ObjectAttribute used in the CatalogData section is added to.	0..1
ProductIDExtension	Auxiliary of ProductID	0..1
ExternalItemRef	A reference to a piece of CatalogData in another catalog. Can be either a ItemGUID (see below) or a triple of CatalogID, CatalogProvider, ProductID. *Note!!* Unresolved references do not constitute an error *Note!!* Circular references and recursive references are undefined, and should be considered an error.	0..n
ProductIDStandard	Has an Agency attribute that indicates the standard of ProductID, e.g. EAN, UCC, GTIN .	0..1
ProductName	Name of the product. Has xml:lang attribute for multilanguage support.	0..n
UOM	Represents orderable unit of measure of the product. Contains xCBL element UOM to represent ISO unit of measure code. Describes what unit of measure the LotSize sub-element is expressed in. Defaults to EA (each).	0..1
ComparableUOM	An element of type UOM. The ComparableUOM completely analogous to the notion of a "base UOM" and is included to enable comparisons of this Product's pricings with other (similar) Products	0..1
ComparableUOMConversion Factor	The ComparableUOMConversionFactor is the number of ComparableUOM per UOM.	0..1
Manufacturer	Name of the manufacturer or reference of a manufacturer partner in the ListOfPartners.	0..1
ManuPartNumber	Manufacturer's part number.	0..1
LeadTime	Lead time of the product. Defaults to 1.	0..1
LeadTimeUOM	The time unit of measure for the lead time for the product. Default to DAY if not present. Contains UOM element whose structure is presented in the table in the next section and in the ProductCatalog DTD.	0..1
ValidFrom	Date and time that the product and its price begin to be available. . Needs to be in the format used by all of xCBL: *yyyymmdd* (note the lack of slashes)	0..1
ValidUntil	Date and time that the product and its price stop applying. . Needs to be in the format used by all of xCBL; *yyyymmdd* (note the lack of slashes)	0..1

CountryOfOrigin	Contains Country element with ISO code of country of origin as CountryCoded. See tables below that define Country, UOM, and Currency from the rest of xCBL. Structure is also defined in ProductCatalog DTD.	0..1
MinOrder	Indicates the minimum order. That is, the Minimum number of LotSize x UOM's of ProductID purchasable (e.g., LotSize = 6, UOM = EA, MinOrder = 1, have to buy at least 6 of item). A value of 1 (one) is the default. The ProductPrice MinimumQuantity takes precedence over this value.	0..1
LotSize	Indicates the granularity possible with orders with respect to Product UOM. If Product UOM has a value of EA, and LotSize has a value of 5, then ordering 1 of this Product, is effectively order 5 EA of whatever's offered – essentially, the offering is a 5-pack, ordering one is ordering one 5-pack. Defaults to 1.	0..1
ExternalCategory	Refers to a SchemaCategory present in the document pointed to by the SchemaSource of the CatalogSchema element. With the introduction of the Product sub-element SchemaCategoryRefList, this element is unnecessary, and its use is **deprecated.**	0..n
ShortDescription	Short description(s) of the product.	0..n
LongDescription	Long description(s) of the product.	0..n
CatalogContract	Contract information. The Type attribute indicates whether the Contract is buyer contract, supplier contract or "other".	0..n
ProductPrice	Price(s) of the product.	0..n
ProductVendorData	Vendor data and vendor specified price. If a product is multivendor, price should only appear under ProductVendorData.	0..n
ProductAttachment	Attachments.	0..n
RelatedProduct	ProductID of and relationship to related product(s). The TargetType attribute indicates the relationship of the related product.	0..n
ObjectAttribute	Additional attributes of products not built into the elements described above. NOTE: this is a multiple inheritance model. The order of the categories establishes precedence, the list being ordered in descending precedence. This is most important in determing which class an undeclared ObjectAttribute used here (in the CatalogData section) is added to.	0..n

SchemaCategoryRefList Subelements	Description	Cardinality
CategoryIDRef	Having the same value as the CategoryID of a SchemaCategory implies that Product is an instance of that SchemaCategory.	1..n

ExternalItemRef Subelements	Description	Cardinality
ItemGUID **OR** **CatalogProviderIDRef,** **CatalogIDRef, ProductIDRef**	An ItemGUID is a registered globally unique identifier for an item in a catalog. The alternative is to provide the triple which uniquely identifies a Product: • CatalogProviderID should be a GUID (Commerce One applications expect a TPID, which is a GUID); • CatalogID should be an ID unique to a particular provider; • ProductID must be unique within the catalog.	1..1

CatalogContract Subelements	Description	Cardinality
CatalogContractID	Identifier of the contract.	1..1
CatalogContractItemID	Identifier of this product in the contract	1..1

ProductPrice Subelements	Description	Cardinality
Amount	Decimal currency amount.	1..1
PriceType	Type of the price.	0..1
Currency	ISO currency code.	0..1
UOM	ISO unit of measure code. Defines the unit of measure for the value in PriceBasisQuant sub-element of ProductPrice. Must be *exactly the same as* the UOM sub-element of Product.	0..1
MinimumQuantity	The minimum number of LotSize x UOM's that must be purchased for this pricing to be effective (where UOM is the Product sub-element. If UOM is Liter, and LotSize is 0.5, that would be ½ liters). A value of 0 (zero) is the default and means there is no minimum order. If given, this value takes precedence over the Product MinOrder value.	0..1
ShortDescription	Description of the price.	0..1
ValidFrom	Date that the price begins to be available. . Needs to be in the format used by all of xCBL; *yyyymmdd* (note the lack of slashes)	0..1
ValidUntil	Date that the price stops applying. . Needs to be in the format used by all of xCBL; *yyyymmdd* (note the lack of slashes)	0..1
Buyer	Name of buyer or reference to a buyer partner in ListOfPartners. This must be the value by which a procurement application will identify the buyer to the catalog engine. In the case of Commerce One catalog engines, for example, the buyer's TPID is required. NOTE: the use of this field is deprecated - when ProductPrice is used within a Pricing, the value of Buyer is ignored in CommerceOne applications.	0..1

| PriceBasisQuant | Quantity that is the basis of the price. This enables one to control the level of precision with which the price is expressed.
Must be a number. Amount /PriceBasisQuant = price per 1 UOM. That is, the number of UOM's of the Product the Amount will purchase. It has no direct relationship to LotSize: this doesn't specify the "quanta" in which product can be ordered, just the "quanta" by which it's priced. It defaults to 1 (one). For example, if the price is 5 for $10, this value would be 5, the Amount is 10, and the UOM is EA. | 0..1 |

ProductVendorData Subelements	Description	Cardinality
VendorIDRef	Vendor's ID: a value that can be returned to uniquely identify who the vendor (or supplier) of this Product with this set of prices is. Can be a reference to a member of the ListOfPartners as well. Meant to replace the deprecated ProductVendorData XML attribute PartnerRef.	0..1
VendorPartNumber	Vendor's part number.	0..1
LeadTime	Lead time of the product for this vendor.	0..1
LeadTimeUOM	ISO code. Default to DAY.	0..1
CatalogContract	Contract information.	0..1
MinOrder	Indicates the minimum order That is, the Minimum number of LotSize-UOM's of ProductID purchasable (e.g., LotSize = 6, UOM = EA, MinOrder = 1, have to buy at least 6 of item). A value of 1 (one) is the default. The ProductPrice MinimumQuantity takes precedence over this value..	0..1
ProductPrice	Price(s) of the product for this vendor.	0..n

ProductAttachment Subelements	Description	Cardinality
AttachmentURL	URL of the attachment.	1..1
AttachmentPurpose	Purpose of the attachment.	0..1
AttachmentMIMEType	MIME type of the attachment.	0..1
StortDescription	Short description.	0..n
LongDescription	Long description.	0..n

ObjectAttribute Subelements	Description	Cardinality
AttributeID	Identifier of the attribute.	1..1
AttributeUnit	ISO unit of measure code of the ObjectAttribute. Must be commensurable (of the same *dimension,* so it can be converted and compared with other values) with CategoryAttribute DefaultUOM (if supplied) or with all previous UOM's supplied for ObjectAttributes with this AttributeID. Defaults to EA.	0..1
AttributeValue	Value of the ObjectAttribute. If a schema is present more than one AttributeValue can be listed, each with its own xml:lang attribute.	0..n

Pricing Subelements	Description	Cardinality

ProductIDRef	The ProductID of the Product to which the ProductPrices are for.	1..1
PriceCatalogIDRef	The PriceCatalogID of the PriceCatalog which the ProductPrices are members of.	1..1
ProductPrice	ProductPrice(s)	0..n

5 Elements from Rest of xCBL

These elements are all not part of xCBL 3.0 for catalog content but are presented here for convenience.

The Country element is used in Country of Origin.

Country Subelements	Description	Cardinality
CountryCoded	Enumeration of ISO 3166 country codes: ``` AE AF AG AI AL AM AN AO AQ AR AS AT AU AW AZ BA BB BD BE BF BG BH BI BJ BM BN BO BR BS BT BU BW BY BZ CA CC CD CF CG CH CI CK CL CM CN CO CR CS CU CV CX CY CZ DE DJ DK DM DO DZ EC EE EG EH ER ES ET FI FJ FK FM FO FR GA GB GD GE GF GH GI GL GM GN GP GQ GR GS GT GU GW GY HK HN HR HT HU ID IE IL IN IO IQ IR IS IT JM JO JP KE KG KH KI KM KN KP KR KW KY KZ LA LB LC LI LK LR LS LT LU LV LY MA MC MD MG MH MK ML MM MN MO MP MQ MR MS MT MU MV MW MX MY MZ NA NC NE NF NG NI NL NO NP NR NU NZ OM PA PE PF PG PH PK PL PM PN PR PT PW PY QA RE RO RU RW SA SB SC SD SE SG SH SI SJ SK SL SM SN SO SR ST SV SY SZ TC TD TF TG TH TJ TM TN TO TP TR TT TV TW TZ UA UG UM US UY UZ VA VC VE VG VI VN VU WF WS YE YT YU ZA ZM ZW ```	1..1
CountryCodedOther	If not one of above this contains text representing the country.	0..1

The UOM element is used in several places inside xCBL for catalog content to specify the unit of measure of the product and the lead time unit of measure.

UOM Subelements	Description	Cardinality
UOMCoded	Enumeration of UN ECE Recommendation 20 unit of measure codes.	1..1
UOMCodedOther	If not one of above this contains text representing the country.	0..1

The Currency element is used in several places inside xCBL for catalog content to specify currency that prices are denominated in.

Currency Subelements	Description	Cardinality

| CurrencyCoded | Enumeration of valid ISO 4217 currency codes:
AFA DZD ADF ADP ARP ARS ARA AMD AWF AUD ATS
AZM BSD BHD BDT BBD BYB BEF BZD BMD BTN BOB
BAK BWP BRL BND BGL BIF KHR CAD CVE KYD XOF
XAF XPF CLP CLF CNY COP KMF CRC HRK CUP CYP
CZK DKK DEM BAD DJF DOP XCD ECS EGP SVC ERN
EEK ETB EUR XEU FKP FJD ZAL FIM FRF GMD GEL
GHC GIP XAU GRD GTQ GNF GWP GYD HTG HNL HKD
HUF ISK INR IDR IRA IRR IQD IEP ILS ITL JMD
JPY JOD KZT KES KWD AOR KGS LAK LVL LBP ALL
LSL LRD LYD LTL LUF MOP MKD MGF MWK MYR MVR
MTL MRO MUR MXN MDL MNT MAD MZM BOV MMK NAD
NPR ANG AWG NLG AON TWD ZRN NZD NIO NGN KPW
NOK PKR PAB PGK PYG PEN PHP PLN PLZ PTE GBP
QAR OMR ROL RUR RWF WST STD SAR SBL SCR SLL
SGD SKK SIT SBD SOS ZAR KRW ESP LKR SHP SDA
SDD SDP SRG SZL SEK CHF SYP TJR TZS THB TPE
TOP TTD TND TRL TMM USD AED UGX UAH ECV UYU
USN UZS VUV VEB VND YER YUN ZMK ZW | 1..1 |
| CurrencyCodedOther | If not one of above this contains text representing the country. | 0..1 |

Below is the definition of the Party element. Note that this is a very different definition for Party than in xCBL 2.0.

Party Subelements	Description	Cardinality
PartyID	Contains xCBL 3.0 Identifier element.	1..1
ListOfIdentifier	Contains a list of additional Identifiers for Party. ListOfIdentifier is defined as containing one or more Identifier elements.	0..1
MDFBusiness	Contains a string (#PCDATA)	0..1
NameAddress	Name and address of Party.	0..1
OrderContact	Contains an xCBL 3.0 Contact element	0..1
ReceivingContact	Contains an xCBL 3.0 Contact element	0..1
ShippingContact	Contains an xCBL 3.0 Contact element	0..1
OtherContacts	Contains ListOfContact	0..1
CorrespondenceLanguage	Contains xCBL 3.0 Language element	0..1

NameAddress Subelements	Description	Cardinality
ExternalAddressID	Required address identifier.	0..1
Name1	Required name as string data	1..1
Name2	Optional additional name	0..1
Name3	Optional additional name	0..1
Identifier		0..1
POBox		0..1
Street		0..1
HouseNumber		0..1
StreetSupplement1		0..1
StreetSupplement2		0..1
Building		0..1
Floor		0..1
RoomNumber		0..1
InhouseMail		0..1
Department		0..1
PostalCode		0..1
City		0..1
County		0..1
Region		0..1

District		0..1
Country		0..1
Timezone		0..1

Contact Subelements	Description	Cardinality
ContactID	Agency providing identifier. Structure defined in following table.	0..1
ContactName	Contains string (#PCDATA) acting as identifier	1..1
ContactFunction		0..1
ContactDescription		0..1
ListOfContactNumber	Contains one or more **ContactNumber** elements	0..1

ContactFunction Subelements	Description	Cardinality

| ContactFunctionCoded | Agency providing identifier. Structure defined in following table: `AcceptingContact AcceptingOfficial Accountingcontact AccountsPayableContact AccountsReceivableContact AdditionalInsuranceInformationContact AdministrativeContractingOfficer Administrator Advisor After-HoursContact Agent AlternateContact Applicant AppointmentScheduler Appraiser Attention-ToParty Attorney AuditingContact AuthorizedFinancialContact AuthorizedNegotiator AuthorizedRepresentative AuthorizedSignature AutomatedClearinghouseContact BankingContact BillInquiryContact BoardStaff Broker BrokerContact BusinessUnitManager BuyerNameorDepartment CADAndCAMSpecialist Carrier CaseManager CertificationContact Certifier ChairmanOfTheBoard ChangedBy ChangeOrderApprover ChiefExecutiveOfficer ChiefFinancialOfficer ChiefInformationOfficer ChiefOperatingOfficer CityWorksAuthorityContact Claimant ClaimApprover ClaimRecipient ClaimsContact ClearinghouseContact ClerkOfCourt Co-Borrower Co-Investigator Collector CollegeofEducationAdmissionsOffice ComplianceOfficer ComponentEngineer ComputerSystemsContact ConcurrentEmployerContact ConfirmedWith Consignee Consignor ContainerManager ContractContact CoordinationContact Coordinator CorporatePurchasingAgent CostAndScheduleCoordinator CustomerContactGrantingAppointment CustomerEngineer CustomerMaintenanceManager CustomerRelations DangerousGoodsContact DeliveryContact DeliveryInstructionsContact DentalSchoolAdmissionsOffice DepartmentOrEmployeeToExecuteExportProcedures DepartmentOrEmployeeToExecuteImportProcedures DepartmentOrPersonResponsibleForProcessingPurchaseOrder DesignEngineer Development Director DirectoryAdvertisingContact DivisionDirector DivisionManager EDICoordinator EducationCoordinator ElectricitySupplyContact ElectronicDataInterchangeCoordinator ElectronicSubmissionRecipient EmergencyContact EmergencyContact-Consignee EmergencyContact-MilitaryTrafficManagementCommand EmergencyContact-Shipper EmergencyDangerousGoodsContact EmployerContact Engineer EngineeringContact EnteredBy Estimator EveningProgramsOffice ExecutiveOfficer ExecutiveVice-President Expeditor FinancialAidOffice ForeclosingLenderAdministrativeContact ForwarderContact FunctionalManager GasSupplyContact GeneralContact GoodsReceivingContact GraduateAdmissionsOffice GraduateBusinessOffice GraduateEngineeringOffice GraduateFineArtsOffice GuidanceCounselor HazardousMaterialContact HeadOfUnitForInformationDissemination HeadOfUnitForInformationProduction HeadOfUnitOrComputerDataProcessing HealthMaintenanceOrganizationContact HumanResources InformationContact InsuranceContact InsuredParty InsurerContact Interviewer Investigator InvestmentContact IssuingOfficer JointWorkAgent LaboratoryContact LandregistryContact LawFirm LawSchoolAdmissionsOffice LegalAuditingContact Licensee LocalPurchasingAgent MaintenanceContact Manager Manufacturing ManufacturingPlantContact MarketingDepartment` | 1..1 |

ContactFunctionCodedOther	Contains string (#PCDATA) acting as identifier	0..1

ContactNumber Subelements	Description	Cardinality
ContactNumberValue	.	1..1
ContactNumberTypeCoded	`Other TelephoneNumber FaxNumber EmailAddress MobileNumber TelexNumber PagerNumber HomePage WWW FTP URLOther`	1..1
ContactNumberTypeCodedOther		0..1

Identifier Subelements	Description	Cardinality
Agency	Agency providing identifier. Structure defined in following table.	1..1
Ident	Contains string (#PCDATA) acting as identifier	1..1

Agency Subelements	Description	Cardinality

AgencyCoded	Agency providing identifier. Must be from the following list:	1..1

	`TheMicrocomputerIndustryAssociation ACGIH`
	`ACORD AdvertisingIndustry AECMA AIHA`
	`AirTransportAssociationOfAmerica Alabama`
	`Alaska AluminumAssociation`
	`AmericanApparelManufacturersAssociation`
	`AmericanFurnitureManufacturersAssociation`
	`AmericanIronAndSteelInstitute`
	`AmericanLandTitleAssociation`
	`AmericanMedicalAssociation`
	`AmericanPaperInstitute`
	`AmericanPetroleumInstitute APWAAndOCSI`
	`AmericanSocietyForAutomationInPharmacy`
	`AmericanTextileManufacturersInstitute`
	`AmericanTruckingAssociations`
	`ANSIAccreditedStandardsCommittee Arizona`
	`Arkansas ASME AssignedByBuyerOrBuyersAgent`
	`AssignedByCarrier AssignedByDistributor`
	`AssignedByManufacturer`
	`AssignedByNationalTradeAgency`
	`AssignedByOwnerOfOperation`
	`AssignedByPartyOriginatingTheMessage`
	`AssignedBySellerOrSellersAgent`
	`AssignedByTransportCompany ASTM AT AT-`
	`GeldausgabeautomatenServiceGesellschaftMBH`
	`AT-AustrianCustoms  AT-AustrianPTT  AT-`
	`VerbandOesterreichischerBankenUndBankiers AU-`
	`ACOS AU-ADA AU-AIHW AU-APCA AU-AQIS AU-`
	`AustrianCustomsService AU-DOH AU-HIC AU-LIFA`
	`AU-NCCH AU-ROA AU-SAA AU-Tradegate AWS AYSA`
	`BE BE-BelgianBankersAssociation BE-`
	`BelgianCustoms BE-BelgianMinistryOfFinance`
	`Bellcore BG BIC`
	`BookIndustrySystemsAdvisoryCommittee BR-`
	`BancoCentralDoBrazil BR-`
	`MAAndRAAAndLIMNETAndRINET CA-CPA  CA-CSIO`
	`California CaliforniaLandTitleAssociation`
	`CanadianFreightClassification`
	`CarbideManufacturers CA-`
	`RevenueCanadaCustomsAndExcise CAS CA-SCC  CCC`
	`CEC CECDGXIIID5 CECDGXXI01 CECDGXXIB1`
	`CECDGXXXIV CEFIC CH CH-`
	`AdministrationFederaleDesContributions CH-`
	`AssociationSuisseCodeDesArticles CH-Carbura`
	`CH-CentraleSuissePourLImportationDuCharbon`
	`CH-DirectionGeneraleDesDouanes CH-`
	`DivisionDesImportationsEtExportations CH-`
	`EntrepriseDesPTT CH-`
	`MinistryOfTaxationCentralCustomsAndTaxAdminis`
	`tration CH-`
	`OfficeFiduciaireDesImportateursDeDenreesAlime`
	`ntaires CH-SwissSecuritiesClearingCorporation`
	`CH-TelekursAG CIDX CIECA`
	`CodeAssignedByTheOrganizationThatIsTheUltimat`
	`eDestinationOfTheTransactionSet CO-DIAN`
	`Colorado CommanderRomeAirDevelopmentCenter`
	`ConcordEDIGroup Connecticut`
	`ConventionsOnSADAndTransit`
	`CopperAndBrassFabricatorsCouncil CO-SBDC`
	`CountyDesignatorCode CSI CSIExtended CUSIP CZ`
	`DE DE-ARE DE-BDI DE-BIA  DE-BRD DE-DIN DE-`
	`DTAG DE-EDIBAU DefenseLogisticsAgency DE-GDV`
	`DE-GermanBankersAssociation DE-GermanCustoms`
	`DE-GMBH Delaware DepartmentOfAirForce`
	`DepartmentOfArmy DepartmentOfDefense`
	`DepartmentOfHealthAndHumanServices`
	`DepartmentOfHousingAndUrbanDevelopment`
	`DepartmentOfLabor DepartmentOfTheNavy DE-`
	`SiemensAG DE-VDA DFAS DIN`
	`DirectionGeneraleDesDouanesEtDroitsIndirects`
	`DistrictOfColumbia DK DK-EAN DK-MHA DK-NBH`
	`DK-PBS  DunAndBradstreet EAN Edibuild`
	`EdielNordicForum EDIFICE EDITEUR EDITEX EEG7`
	`EIDX ElectricUtilities`
	`ElectronicIndustriesAssociation EPA ES ESDX`
	`ES-ODETT ES-SpanishCustoms EU-EDIFICAS EU-`
	`ETIS EU-ROFER FASLINC FCC FCI  FDA`	

AgencyCodedOther	In case that an agency not in the enumeration above, any agency name can be placed in this field. Any valid string (#PCDATA) can be accepted.	0..1
AgencyDescription	Description of agency.	0..1
CodeListIdentifierCoded	CodedListIdentifier from xCBL 3.0 enumertion	0..1
CodeListIdentifierCodedOther	If CodeListIdentifier not in above enumeration, value can be placed here.	0..1

6 Catalog Update Semantics

ProductCatalog in xCBL 3.5 attempts to make the update semantics considerably more straight-forward. One of the ways it does is this by separating Pricing information from the Product, making the use of the IsPriceUpdate "flag" unnecessary (and is thus deprecated). In addition, it adds the "Replace" action, which adds the convenience and clarity of combining the "Delete" of an old item and the "Add" of a new, replacement item into one action.

Here are the overall semantics of the four actions now available to the catalog author:

- Add – Add is not applicable to PriceCatalogs (see discussion in Section 6.3), Add the element given, and specified sub-elements, if there's no existing element that matches on it's key. If there is a matching, existing element, the sub-element values given replace the current sub-element if that sub-element is single-valued. If it's multi-valued, then add another value to the existing set of values for that sub-element. If no value is given for the sub-element of an existing element, the previous value (if any) is left intact.
- Update – Similar to an Add, but with two key differences: there must be an existing element that matches on it's key (otherwise it's an error); and existing values are replaced, regardless of whether they're defined as single-valued or multi-valued sub-elements.
  - for the Product sub-element: if no value is given for the sub-element of an existing element, the previous value (if any) is left intact. This facilitates incremental replacements of specific sub-element values in a Product entry, without necessitating repeating the values for other sub-elements which should persist.
  - for the Pricing sub-element: if no ProductPrice is given (the only optional sub-element) it is interpreted to mean that all ProductPrices for that PriceCatalog-Product pair are removed. This facilitates removal of a single product from a PriceCatalog without necessitating repeating the values for all the remaining PriceCatalog-Product pairs, as would be the case if using "Replace" were the only way to effect this change.
- Replace – If there's an existing element that matches on it's key values, delete it and add the values for the current definition as the new value for the element. If there's no existing definition, just add this new value.
- Delete – If there's an existing element that matches on it's key values, delete the existing element and all of it's sub-elements from the catalog. References to the deleted item are not removed. The latter is an important distinction – if a Product is removed, it's pricing persists, and if re-introduced it will suddenly have pricing, though none's apparently been specified.

What follows are discussions of the specific elements that have an Action sub-element associated with them.

6.1 Partner

The key is the value specified in the Partner's Party sub-element's Identifier Ident sub-element.

The extent of the Action is restricted to membership in the list *only*. That is to say, membership in the ListOfPartners has no defined side-effects.

6.2 SupplierAccount

The key is the SupplierAccountID sub-element's value.

A catalog author can use the defined Actions to add a new Supplier Accounts (Add), add new PriceCatalogs to an existing list (for example, a PriceCatalog of limited-time specials)(Add), redefine the set of PriceCatalogs associated with an account (Update), redefine all the information associated with an account (Replace), or to Remove the account (Delete).

6.3   PriceCatalog

The key is the value of the PriceCatalogID sub-element. The update semantics for the immediate sub-elements of PriceCatalog are as for Product. However, the Action also applies to the set of Pricings associated with this PriceCatalog, and those semantics are now discussed.

NOTE!! See the Update discussion in section 6 which has a different interpretation in the Pricing context than in the Product context.

NOTE!! The Add action on PriceCatalogs is the default action, but the semantics for Add for PriceCatalogs is identical to the semantics defined for Update. Effectively, there is no Add for PriceCatalogs, only Update (that is, though Add is still a legal value, its effect is indistinguishable from Update).

The action is interpreted across ALL PricingElements whose PriceCatalogIDRef sub-element's value matches the given PriceCatalogID. In particular:
- a Replace Action on a PriceCatalog is interpreted to Delete *all* matching Pricings (Pricings in the existing catalog which have the same PriceCatalogIDRef value), and replace them with the set included – this could result in the removal of some Pricings for some Products.
- A Delete Action deletes all matching Pricings (Pricings in the existing catalog which have the same PriceCatalogIDRef value).
- An Add or Update Action mean the same thing as they effect Pricings. Namely, for each Pricing element in the given document, the matching, existing Pricing in the catalog is effected (where match is based on *two* key values, not only that of the PriceCatalogIDRef, but also the ProductIDRef), and the set of ProductPrices associated with this Pricing are replaced with the new ones, If there is no matching Pricing element in the existing catalog, the new information is used to add one

This design was arrived at to facilitate PriceCatalog management:
- Only a single PriceCatalog entry in the CatalogHeader sub-element PricingInformation with the Action Value of Delete is necessary to remove all the ProductPrices associated with a PriceCatalog.
- Only a single PriceCatalog entry in the CatalogHeader sub-element PricingInformation with the Action Value of Replace is necessary to remove all the ProductPrices associated with a PriceCatalog and provide a new set (which may remove some Products from that PriceCatalog, if no new prices are specified).
- The default action is to replace the current set of prices for a Product in a PriceCatalog with the new values, making it impossible to supply new prices to implement a price hike, and inadvertently leave the older, lower prices still available.

6.4   Product

The CatalogHeader can be used to control how the contents of the remainder of the catalog will be used:
- IsReplacement is specified to indicate that the entire contents of the Catalog specified in CatalogData are meant to replace all of the Product items in the current version of the Catalog on the Catalog Server with those found in the CatalogData section of the current document. Further, if (and only if) a CatalogSchema element is included, not only are the Product items from the current Catalog removed, but so is any information of classes of Products as specified by SchemaCategory items.
- IsPriceUpdate is specified to indicate that (a) the category information for the Product is ignored and (b) only the ProductPrice information will be changed on the Product's named by the Product elements in the CatalogData section - all other information about the Product is ignored. The result

of Action on ProductPrices depends on the value of IsPriceUpdate. That is, while all other Product elements are ignored when IsPriceUpdate is specified, ProductPrices are not ignored if IsPriceUpdate is not specified - see below for details.

- IsMultiVendor specifies that a catalog is a multi-vendor catalog containing products sold by more than one vendor. If IsMultiVendor is in effect for a catalog, Product elements may contain ProductVendorData elements, but must not include ProductPrice elements directly beneath the Product element. Similarly, if IsMultiVendor is not in effect for a catalog, Product elements must not include ProductVendorData elements.

N.B.: Specifying IsReplacement and IsPriceUpdate in the same document is a contradiction.
N.B.: Specifying IsPriceUpdate in the first ProductCatalog document produced is a contradiction.

Each Product element may each contain an Action element which can have the following values:

- **Add, Delete, Update, Replace**

IsPriceUpdate specified: take no action with respect to any values specified for a Product with this Action subelement value. Implementation is free to ignore that Product entry or to warn that it is taking no action.
IsPriceUpdate not specified: utilize the semantics above.

- **Unspecified**

IsPriceUpdate specified:
- Multivendor catalog –
	If prices exist for specified buyer and vendor, clear-vendor-buyer-prices. Add-vendor-buyer-price
- Not Multivendor catalog –
	If any prices exist for specified buyer and product, clear-buyer-prices. Add –buyer – prices from this document for this product.

	N.B.: only Price subelement values are changed, any other subelement values specified will be ignored.

IsPriceUpdate not specified: if object corresponding to the Product element doesn't exist, same as Add – IsPriceUpdate not specified. If the object corresponding to the Product element does exist, same as Update – PriceUpdate not specified.

6.5 Examples

What follows are some scenarios to clarify various implications of the update semantics

6.5.1   Partner

Catalog File	Intent of catalog document contents
A-Catalog	Define a single partner of relationship buyer, Bobby. Two price catalogs are defined, A-PC and B-PC. A supplier account is established so A-PC is accessible to only Bobby, but B-PC is accessible to any partner with relationship buyer - of which there is only Bobby, in A-Catalog. Hence prices in A-PC and B-PC, are available only to Bobby's buyers.
B-Catalog	Add a single partner of relationship buyer, THC. The set of prices accessible to Bobby remains unchanged, and THC can only access the prices in price catalog B-PC.
C-Catalog	Delete Bobby. Bobby now can only access price catalog A-PC prices, and THC continues to be able to access B-PC prices.
D-Catalog	Add a single partner with no relationship defined (defaults to Supplier, so NOT a buyer), Fred. The addition changes nothing in terms of accessibility - THC has access only to B-PC prices and A-PC prices are still only accessible to Bobby.

The above table outlines a series of documents which are incremental changes to the same catalog.

NOTE: Only Partners of with a PartnerRelationshipCoded value of *Buyer* have any semantics defined, and only when *CatalogAudienceCoded* has the value of *EnumeratedBuyersOnly*.

These are the corresponding documents:

A-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>
 <CatalogID>MyCat-A-LOP</CatalogID>

 <CatalogProvider>
   <Party><PartyID>
            <Identifier>
               <Agency><AgencyCoded>Other</AgencyCoded>
                         <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
               </Agency>
               <Ident>GTC</Ident>
            </Identifier>
 </PartyID></Party></CatalogProvider>

 <ListOfPartners>
  <Partner>
    <Party><PartyID>
            <Identifier>
               <Agency><AgencyCoded>Other</AgencyCoded>
                         <AgencyCodedOther>KRB-Agency</AgencyCodedOther>
               </Agency>
               <Ident>Bobby</Ident>
            </Identifier>
    </PartyID></Party>
   <PartnerRelationship>
     <PartnerRelationshipCoded>Buyer</PartnerRelationshipCoded>
   </PartnerRelationship>
  </Partner>
 </ListOfPartners>

 <CatalogAudience CatalogAudienceCoded="EnumeratedBuyersOnly"/>

 <PricingInformation>
  <PriceCatalog><PriceCatalogID>A-PC</PriceCatalogID></PriceCatalog>
  <PriceCatalog><PriceCatalogID>B-PC</PriceCatalogID></PriceCatalog>
 </PricingInformation>

 <SupplierAccountInformation>
  <SupplierAccount>
   <SupplierAccountID>Bob</SupplierAccountID>
   <BuyerIdentifier>
    <Identifier>
     <Agency><AgencyCoded>GTC-Agency</AgencyCoded></Agency>
     <Ident>Bobby</Ident>
    </Identifier>
   </BuyerIdentifier>
   <PriceCatalogIDRef>A-PC</PriceCatalogIDRef>
  </SupplierAccount>

  <SupplierAccount>
```

A-Catalog.XML

```
   <IsPublicAccount/>
   <PriceCatalogIDRef>B-PC</PriceCatalogIDRef>
  </SupplierAccount>
 </SupplierAccountInformation>

</CatalogHeader>

<CatalogData>

 <Product>
  <ProductID>n-buyer-add-n-1-2</ProductID>
 </Product>

 <Product>
  <ProductID>N-BUYER-ADD-N-1-1</ProductID>
 </Product>

 <Pricing>
  <ProductIDRef>n-buyer-add-n-1-2</ProductIDRef>
  <PriceCatalogIDRef>A-PC</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>66.11</Amount>
  </ProductPrice>
  <ProductPrice>
    <Amount>66.22</Amount>
    <MinimumQuantity>200</MinimumQuantity>
  </ProductPrice>
 </Pricing>

 <Pricing>
  <ProductIDRef>N-BUYER-ADD-N-1-1</ProductIDRef>
  <PriceCatalogIDRef>A-PC</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>66.12</Amount>
  </ProductPrice>
  <ProductPrice>
    <Amount>66.23</Amount>
    <MinimumQuantity>200</MinimumQuantity>
  </ProductPrice>
 </Pricing>

 <Pricing>
  <ProductIDRef>n-buyer-add-n-1-2</ProductIDRef>
  <PriceCatalogIDRef>B-PC</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>76.11</Amount>
  </ProductPrice>
  <ProductPrice>
    <Amount>76.22</Amount>
    <MinimumQuantity>200</MinimumQuantity>
  </ProductPrice>
 </Pricing>

 <Pricing>
  <ProductIDRef>N-BUYER-ADD-N-1-1</ProductIDRef>
```

A-Catalog.XML

```
  <PriceCatalogIDRef>B-PC</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>76.12</Amount>
  </ProductPrice>
  <ProductPrice>
    <Amount>76.23</Amount>
  </ProductPrice>
 </Pricing>

</CatalogData>

</ProductCatalog>
```

B-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>
 <CatalogID>MyCat-A-LOP</CatalogID>

 <CatalogProvider>
   <Party><PartyID>
            <Identifier>
              <Agency><AgencyCoded>Other</AgencyCoded>
                       <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
              </Agency>
              <Ident>GTC</Ident>
            </Identifier>
 </PartyID></Party></CatalogProvider>

 <ListOfPartners>
  <Partner>
   <Party><PartyID>
            <Identifier>
              <Agency><AgencyCoded>Other</AgencyCoded>
                       <AgencyCodedOther>THC-Agency</AgencyCodedOther>
              </Agency>
              <Ident>THC</Ident>
            </Identifier>
   </PartyID></Party>
   <PartnerRelationship>
    <PartnerRelationshipCoded>Buyer</PartnerRelationshipCoded>
   </PartnerRelationship>
  </Partner>
 </ListOfPartners>

</CatalogHeader>

</ProductCatalog>
```

C-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>
 <CatalogID>MyCat-A-LOP</CatalogID>
```

C-Catalog.XML

```xml
<CatalogProvider>
   <Party><PartyID>
            <Identifier>
              <Agency><AgencyCoded>Other</AgencyCoded>
                        <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
              </Agency>
              <Ident>GTC</Ident>
            </Identifier>
</PartyID></Party></CatalogProvider>

<ListOfPartners>
  <Partner>
   <Party><PartyID>
            <Identifier>
              <Agency><AgencyCoded>Other</AgencyCoded>
                        <AgencyCodedOther>THC-Agency</AgencyCodedOther>
              </Agency>
              <Ident>THC</Ident>
            </Identifier>
   </PartyID></Party>
   <PartnerRelationship>
    <PartnerRelationshipCoded>Buyer</PartnerRelationshipCoded>
   </PartnerRelationship>
  </Partner>
</ListOfPartners>

</CatalogHeader>

</ProductCatalog>
```

D-Catalog.XML

```xml
<ProductCatalog>

<CatalogHeader>
 <CatalogID>MyCat-A-LOP</CatalogID>

 <CatalogProvider>
   <Party><PartyID>
            <Identifier>
              <Agency><AgencyCoded>Other</AgencyCoded>
                        <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
              </Agency>
              <Ident>GTC</Ident>
            </Identifier>
</PartyID></Party></CatalogProvider>

<ListOfPartners>

  <!-- NOT a BUYER -->
  <Partner>
   <Party><PartyID>
            <Identifier>
              <Agency><AgencyCoded>Other</AgencyCoded>
                        <AgencyCodedOther>KRB-Agency</AgencyCodedOther>
```

D-Catalog.XML
| ```
                </Agency>
                <Ident>Fred</Ident>
              </Identifier>
            </PartyID>
      </Party>
    </Partner>
  </ListOfPartners>

</CatalogHeader>

</ProductCatalog>
``` |

6.5.2 SupplierAccount

Catalog File	Intent of catalog document contents

A-Catalog	Define an initial catalog document with:
	  3 Partners who are Buyers (B1, B2, B3),
	  5 PriceCatalogs (PC1, PC2, PC3, PC4, PC5),
	  4 SupplierAccounts (SA1, SA2, SA3, SA4),
	  5 Products (Product-1, Product-2, Product-3, Product-4, Product-5)
	  4 Pricings for the pairs (Product-1, PC1), (Product-2, PC2), (Product-3,
	   PC3), (Product-4, PC4), (Product-5, PC5)
	  SupplierAccount/Buyer/PriceCatalog relations of:
	   (SA1, B1, PC1),
	   (SA2, B2, PC2),
	   (SA3, B3, PC3),
	   (SA4, IsPublicAccount, PC4).
	  The result is a catalog with:
	   Product-4 having a price accessible to buyers for anyone;
	   Product-1 having a price accessible to buyers for B1 only;
	   Product-2 having a price accessible to buyers for B2 only;
	   Product-3 having a price accessible to buyers for B3 only;
	   Product-5 having a price which is inaccessible.

B-Catalog	Supply minimal CatalogHeader info to modify catalog defined by A-Catalog
	  document so that:
	   SupplierAccount/Buyer/PriceCatalog relation ID'd as SA4 is deleted.
	  The result is a catalog with:
	   Product-1 having a price accessible to buyers for B1 only;
	   Product-2 having a price accessible to buyers for B2 only;
	   Product-3 having a price accessible to buyers for B3 only;
	   Product-4 having a price which is inaccessible;
	   Product-5 having a price which is inaccessible.

C-Catalog	Supply minimal CatalogHeader info to modify catalog modified by B-Catalog
	  document so that:
	     SupplierAccount/Buyer/PriceCatalog relation ID'd as SA3 has two
	     PriceCatalogs added to it:
	   (SA3, B3, (PC3,PC4,PC5)).
	  The result is a catalog with:
	   Product-1 having a price accessible to buyers for B1 only;
	   Product-2 having a price accessible to buyers for B2 only;
	   Product-3 having a price accessible to buyers for B3 only;
	   Product-4 having a price accessible to buyers for B3 only;
	   Product-5 having a price accessible to buyers for B3 only.

D-Catalog	Supply minimal CatalogHeader info to modify catalog modified by C-Catalog
	  document so that:
	   Using SupplierAccount Action=Update, make SupplierAccount/Buyer/PriceCatalog
	   relation ID'd as SA3 have only one price catalog associated with it, PC5:
	   (SA3, B3, PC5).
	  The result is a catalog with:
	   Product-1 having a price accessible to buyers for B1 only;
	   Product-2 having a price accessible to buyers for B2 only;
	   Product-3 having a price which is inaccessible;
	   Product-4 having a price which is inaccessible;
	   Product-5 having a price accessible to buyers for B3 only.

The above table outlines a series of documents which are incremental changes to the same catalog.

NOTE: SupplierAccounts are the means for expressing the audiences for PriceCatalogs: if a PriceCatalog is referenced by no SupplierAccount, it's prices are unavailable to anyone; if a SupplierAccount is defined as IsPublicAccount, then the associated PriceCatalogs are viewable by anyone (if CatalogAudience value is Public) or by a restricted set of buyers (if CatalogAudience is EnumeratedBuyersOnly).

These are the corresponding documents:

A-Catalog.xml

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-SupplierAccount</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                     <Agency>
                        <AgencyCoded>Other</AgencyCoded>
                        <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                     </Agency>
                     <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

 <ListOfPartners>

  <Partner>
   <Party>
    <PartyID>
     <Identifier>
      <Agency>
       <AgencyCoded>Other</AgencyCoded>
          <AgencyCodedOther>KRB-Agency</AgencyCodedOther>
      </Agency>
      <Ident>B1</Ident>
     </Identifier>
    </PartyID>
   </Party>
   <PartnerRelationship>
    <PartnerRelationshipCoded>Buyer</PartnerRelationshipCoded>
   </PartnerRelationship>
  </Partner>

  <Partner>
   <Party>
    <PartyID>
     <Identifier>
      <Agency>
        <AgencyCoded>Other</AgencyCoded>
        <AgencyCodedOther>KRB-Agency</AgencyCodedOther>
      </Agency>
      <Ident>B2</Ident>
     </Identifier>
    </PartyID>
   </Party>
   <PartnerRelationship>
```

A-Catalog.xml

```
    <PartnerRelationshipCoded>Buyer</PartnerRelationshipCoded>
   </PartnerRelationship>
  </Partner>

  <Partner>
   <Party>
    <PartyID>
     <Identifier>
      <Agency>
        <AgencyCoded>Other</AgencyCoded>
        <AgencyCodedOther>KRB-Agency</AgencyCodedOther>
      </Agency>
      <Ident>B3</Ident>
     </Identifier>
    </PartyID>
   </Party>
   <PartnerRelationship>
    <PartnerRelationshipCoded>Buyer</PartnerRelationshipCoded>
   </PartnerRelationship>
  </Partner>

</ListOfPartners>

<PricingInformation>
 <PriceCatalog>
  <PriceCatalogID>PC1</PriceCatalogID>
 </PriceCatalog>
 <PriceCatalog>
  <PriceCatalogID>PC2</PriceCatalogID>
 </PriceCatalog>
 <PriceCatalog>
  <PriceCatalogID>PC3</PriceCatalogID>
 </PriceCatalog>
 <PriceCatalog>
  <PriceCatalogID>PC4</PriceCatalogID>
 </PriceCatalog>
 <PriceCatalog>
  <PriceCatalogID>PC5</PriceCatalogID>
 </PriceCatalog>
</PricingInformation>

<SupplierAccountInformation>
 <SupplierAccount>
  <SupplierAccountID>SA1</SupplierAccountID>
  <BuyerIdentifier>
   <Identifier>
    <Agency><AgencyCoded>Other</AgencyCoded>
            <AgencyCodedOther>GTC-Agency</AgencyCodedOther></Agency>
    <Ident>B1</Ident>
   </Identifier>
  </BuyerIdentifier>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
 </SupplierAccount>
 <SupplierAccount>
  <SupplierAccountID>SA2</SupplierAccountID>
```

A-Catalog.xml

```
   <BuyerIdentifier>
    <Identifier>
     <Agency><AgencyCoded>Other</AgencyCoded>
             <AgencyCodedOther>GTC-Agency</AgencyCodedOther></Agency>
     <Ident>B2</Ident>
    </Identifier>
   </BuyerIdentifier>
   <PriceCatalogIDRef>PC2</PriceCatalogIDRef>
  </SupplierAccount>
  <SupplierAccount>
   <SupplierAccountID>SA3</SupplierAccountID>
   <BuyerIdentifier>
    <Identifier>
     <Agency><AgencyCoded>Other</AgencyCoded>
             <AgencyCodedOther>GTC-Agency</AgencyCodedOther></Agency>
     <Ident>B3</Ident>
    </Identifier>
   </BuyerIdentifier>
   <PriceCatalogIDRef>PC3</PriceCatalogIDRef>
  </SupplierAccount>
  <SupplierAccount>
   <SupplierAccountID>SA4</SupplierAccountID>
   <IsPublicAccount/>
   <PriceCatalogIDRef>PC4</PriceCatalogIDRef>
  </SupplierAccount>
 </SupplierAccountInformation>

</CatalogHeader>

<!--
<CatalogSchema>
</CatalogSchema>
-->

<CatalogData>

 <Product>
  <ProductID>Product-1</ProductID>
 </Product>

 <Product>
  <ProductID>Product-2</ProductID>
 </Product>

 <Product>
  <ProductID>Product-3</ProductID>
 </Product>

 <Product>
  <ProductID>Product-4</ProductID>
 </Product>

 <Product>
  <ProductID>Product-5</ProductID>
 </Product>
```

A-Catalog.xml

```
 <Pricing>
  <ProductIDRef>Product-1</ProductIDRef>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>11.11</Amount>
  </ProductPrice>
 </Pricing>

 <Pricing>
  <ProductIDRef>Product-2</ProductIDRef>
  <PriceCatalogIDRef>PC2</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>22.22</Amount>
  </ProductPrice>
 </Pricing>

 <Pricing>
  <ProductIDRef>Product-3</ProductIDRef>
  <PriceCatalogIDRef>PC3</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>33.33</Amount>
  </ProductPrice>
 </Pricing>

 <Pricing>
  <ProductIDRef>Product-4</ProductIDRef>
  <PriceCatalogIDRef>PC4</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>44.44</Amount>
  </ProductPrice>
 </Pricing>

 <Pricing>
  <ProductIDRef>Product-5</ProductIDRef>
  <PriceCatalogIDRef>PC5</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>55.55</Amount>
  </ProductPrice>
 </Pricing>

</CatalogData>

</ProductCatalog>
```

B-Catalog.xml

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-SupplierAccount</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                  <Agency>
```

B-Catalog.xml

```
                <AgencyCoded>Other</AgencyCoded>
                <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
               </Agency>
               <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

 <SupplierAccountInformation>
  <SupplierAccount>
   <Action Value="Delete"/>
   <SupplierAccountID>SA4</SupplierAccountID>
  </SupplierAccount>
 </SupplierAccountInformation>

</CatalogHeader>

</ProductCatalog>
```

C-Catalog.xml

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-SupplierAccount</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
               <Agency>
                <AgencyCoded>Other</AgencyCoded>
                <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
               </Agency>
               <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

 <SupplierAccountInformation>
  <SupplierAccount>
   <SupplierAccountID>SA3</SupplierAccountID>
   <BuyerIdentifier>
    <Identifier>
     <Agency><AgencyCoded>Other</AgencyCoded>
            <AgencyCodedOther>GTC-Agency</AgencyCodedOther></Agency>
     <Ident>B3</Ident>
    </Identifier>
   </BuyerIdentifier>
   <PriceCatalogIDRef>PC4</PriceCatalogIDRef>
   <PriceCatalogIDRef>PC5</PriceCatalogIDRef>
  </SupplierAccount>
 </SupplierAccountInformation>

</CatalogHeader>

</ProductCatalog>
```

D-Catalog.xml

D-Catalog.xml

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-SupplierAccount</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                   <Agency>
                     <AgencyCoded>Other</AgencyCoded>
                     <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                   </Agency>
                   <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

 <SupplierAccountInformation>
  <SupplierAccount>
   <Action Value="Update"/>
   <SupplierAccountID>SA3</SupplierAccountID>
   <BuyerIdentifier>
    <Identifier>
     <Agency><AgencyCoded>Other</AgencyCoded>
            <AgencyCodedOther>GTC-Agency</AgencyCodedOther></Agency>
     <Ident>B3</Ident>
    </Identifier>
   </BuyerIdentifier>
   <PriceCatalogIDRef>PC5</PriceCatalogIDRef>
  </SupplierAccount>
 </SupplierAccountInformation>

</CatalogHeader>

</ProductCatalog>
```

6.5.3 PriceCatalog

Catalog File	Intent of catalog document contents
A-Catalog	Define a PriceCatalog "PC1" and put a single price for a single product, "A-Product", in it. Establish a SupplierAccount for Buyers from "B1" with PriceCatalog "PC1". The given single price is only accessible to buyers from B1.
B-Catalog	Without reference to "PC1" in the PricingInformation section, provide a new price for "A-Product" in "PC1". The default action results in only the newly specified price being accessible to only buyers from B1.
C-Catalog	With Action=Add in PricingInformation for "PC1", provide a new price, distinct from those given in A-Catalog and B-Catalog, for "A-Product" in "PC1". Only the newly specified price is accessible only to buyers from B1.
D-Catalog	Without reference to "PC1" in the PricingInformation section, provide a new product "B-Product" and a price for it in "PC1". The default action results in the newly specified price being accessible for B-Product, and the price defined in C-Catalog for A-Product only to buyers from B1
E-Catalog	With Action=Replace in PricingInformation for "PC1", provide a new price, distinct from those given in A-Catalog and B-Catalog, for "A-Product" in "PC1", and no price for B-Product. Only the newly specified price for A-Product is accessible only to buyers from B1, and B-Product is not accessible to any buyer since there's no price for it.
F-Catalog	Without reference to "PC1" in the PricingInformation section, provide a new price for "B-Product" in "PC1". The default action results in the newly specified price being only accessible to buyers from B1.
G-Catalog	Without reference to "PC1" in the PricingInformation section, provide a Pricing for "A-Product" in "PC1" which contains no ProductPrices. This results in the removal of A-Product from the PC1 PriceCatalog, which is to say that only "B-Product" is only accessible to buyers from B1, because there will be no available price for A-Product.
H-Catalog	With Action=Update in PricingInformation for "PC1", provide a new price, distinct from those given in previous catalogs, for "A-Product" in "PC1". The newly specified price being only accessible to buyers from B1 as is the price for B-Product specified in F-Catalog.
I-Catalog	With Action=Delete in PricingInformation for "PC1", no products should be accessible as none have prices defined.
J-Catalog	Add two pricings for identical PriceCatalogs and ProductID's. Only the last one should appear in the catalog.

The above table outlines a series of documents which are incremental changes to the same catalog.

These are the corresponding documents:

A-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-PriceCat</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                 <Agency>
                   <AgencyCoded>Other</AgencyCoded>
                   <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                 </Agency>
                 <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>
```

A-Catalog.XML

```
 <PricingInformation>
  <PriceCatalog>
   <PriceCatalogID>PC1</PriceCatalogID>
  </PriceCatalog>
 </PricingInformation>

 <SupplierAccountInformation>
  <SupplierAccount>
   <SupplierAccountID>Bob</SupplierAccountID>
   <BuyerIdentifier>
    <Identifier>
     <Agency><AgencyCoded>Other</AgencyCoded>
             <AgencyCodedOther>GTC-Agency</AgencyCodedOther></Agency>
     <Ident>B1</Ident>
    </Identifier>
   </BuyerIdentifier>
   <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
  </SupplierAccount>
 </SupplierAccountInformation>

</CatalogHeader>



<CatalogData>

 <Product>
  <ProductID>A-Product</ProductID>
 </Product>

 <Pricing>
  <ProductIDRef>A-Product</ProductIDRef>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>11.11</Amount>
  </ProductPrice>
 </Pricing>

</CatalogData>

</ProductCatalog>
```

B-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-PriceCat</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                   <Agency>
                     <AgencyCoded>Other</AgencyCoded>
                     <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
```

B-Catalog.XML

```
                </Agency>
                <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

</CatalogHeader>

<CatalogData>

 <Pricing>
  <ProductIDRef>A-Product</ProductIDRef>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>22.22</Amount>
  </ProductPrice>
 </Pricing>

</CatalogData>

</ProductCatalog>
```

C-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-PriceCat</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                   <Agency>
                     <AgencyCoded>Other</AgencyCoded>
                     <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                   </Agency>
                   <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

 <PricingInformation>
  <PriceCatalog>
   <Action Value="Add"/>
   <PriceCatalogID>PC1</PriceCatalogID>
  </PriceCatalog>
 </PricingInformation>

</CatalogHeader>

<CatalogData>

 <Pricing>
  <ProductIDRef>A-Product</ProductIDRef>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>33.33</Amount>
  </ProductPrice>
```

C-Catalog.XML

```
 </Pricing>

</CatalogData>

</ProductCatalog>
```

D-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-PriceCat</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                    <Agency>
                       <AgencyCoded>Other</AgencyCoded>
                       <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                    </Agency>
                    <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

</CatalogHeader>

<CatalogData>

 <Product>
  <ProductID>B-Product</ProductID>
 </Product>

 <Pricing>
  <ProductIDRef>B-Product</ProductIDRef>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>111.11</Amount>
  </ProductPrice>
 </Pricing>

</CatalogData>

</ProductCatalog>
```

E-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-PriceCat</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                    <Agency>
                       <AgencyCoded>Other</AgencyCoded>
                       <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
```

E-Catalog.XML

```
          </Agency>
          <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

 <PricingInformation>
  <PriceCatalog>
   <Action Value="Replace"/>
   <PriceCatalogID>PC1</PriceCatalogID>
  </PriceCatalog>
 </PricingInformation>

</CatalogHeader>

<CatalogData>

 <Pricing>
  <ProductIDRef>A-Product</ProductIDRef>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>44.44</Amount>
  </ProductPrice>
 </Pricing>

</CatalogData>

</ProductCatalog>
```

F-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-PriceCat</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                  <Agency>
                    <AgencyCoded>Other</AgencyCoded>
                    <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                  </Agency>
                  <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

</CatalogHeader>

<CatalogData>

 <Pricing>
  <ProductIDRef>B-Product</ProductIDRef>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>222.22</Amount>
  </ProductPrice>
```

F-Catalog.XML

```
 </Pricing>

</CatalogData>

</ProductCatalog>
```

G-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-PriceCat</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                   <Agency>
                     <AgencyCoded>Other</AgencyCoded>
                     <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                   </Agency>
                   <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

</CatalogHeader>

<CatalogData>

 <Pricing>
  <ProductIDRef>A-Product</ProductIDRef>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
 </Pricing>

</CatalogData>

</ProductCatalog>
```

H-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-PriceCat</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                   <Agency>
                     <AgencyCoded>Other</AgencyCoded>
                     <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                   </Agency>
                   <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

 <PricingInformation>
  <PriceCatalog>
```

H-Catalog.XML

```
   <Action Value="Update"/>
   <PriceCatalogID>PC1</PriceCatalogID>
  </PriceCatalog>
 </PricingInformation>

</CatalogHeader>

<CatalogData>

 <Pricing>
  <ProductIDRef>A-Product</ProductIDRef>
  <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
  <ProductPrice>
    <Amount>55.55</Amount>
  </ProductPrice>
 </Pricing>

</CatalogData>


</ProductCatalog>
```

I-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>

 <CatalogID>MyCat-PriceCat</CatalogID>

 <CatalogProvider>
  <Party><PartyID><Identifier>
                    <Agency>
                      <AgencyCoded>Other</AgencyCoded>
                      <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                    </Agency>
                    <Ident>GTC-PriceCat</Ident>
  </Identifier></PartyID></Party>
 </CatalogProvider>

 <PricingInformation>
  <PriceCatalog>
   <Action Value="Delete"/>
   <PriceCatalogID>PC1</PriceCatalogID>
  </PriceCatalog>
 </PricingInformation>

</CatalogHeader>

</ProductCatalog>
```

J-Catalog.XML

```
<ProductCatalog>

<CatalogHeader>
```

J-Catalog.XML

```
<CatalogID>MyCat-PriceCat</CatalogID>

<CatalogProvider>
 <Party><PartyID><Identifier>
                  <Agency>
                    <AgencyCoded>Other</AgencyCoded>
                    <AgencyCodedOther>GTC-Agency</AgencyCodedOther>
                  </Agency>
                  <Ident>GTC-PriceCat</Ident>
 </Identifier></PartyID></Party>
</CatalogProvider>

<PricingInformation>
 <PriceCatalog>
  <PriceCatalogID>PC1</PriceCatalogID>
 </PriceCatalog>
</PricingInformation>

</CatalogHeader>

<CatalogData>

<Pricing>
 <ProductIDRef>A-Product</ProductIDRef>
 <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
 <ProductPrice>
   <Amount>66.66</Amount>
 </ProductPrice>
</Pricing>

<Pricing>
 <ProductIDRef>A-Product</ProductIDRef>
 <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
 <ProductPrice>
   <Amount>77.77</Amount>
 </ProductPrice>
</Pricing>

</CatalogData>

</ProductCatalog>
```

6.5.4 Product

A series of N catalogs to test Product update semantics as expressed via
Product Actions of which there are effectively 4:

- Add
- Update
- Replace
- Delete

There are an additional two which are in fact equivalent to two of the above:

- *'the unspecified Action value for new Products'* is equivalent to 'Add'
- *'the unspecified Action value for existing Products'* is equivalent to 'Update'.

Catalog File	Intent of catalog document contents
A-Catalog	Create a catalog with one product (Product-1) having values for every core attribute and an extended attribute (ObjectAttribute). Where attributes can accept multiple values in order to accept values expressed in more tha on natural language, values are supplied in at least two languages.
B-Catalog	Use Product Action 'Replace' to completely replace all the values for Product-1. Part of this should be to make the languages supported different, so be sure that the non-English language set chosen in A-Catalog only partially intersects language set chosen in this document. Also, completely different ObjectAttributes are defined.
C-Catalog	Use the Product Action 'Delete' to remove the sole Product, but not it's Pricing. Product should be inaccessable since it undefined, even though a pricing exists for it.
D-Catalog	Utilize the default Product Action to re-introduce the part, but a minimal definition (part number and class). Given that the Pricing wasn't removed, the Product should again be accessible.

The previous documents should be sufficient to demonstrate the update semantics associated with 'Replace', 'Delete', and 'the unspecified Action value for new Products' on Product elements. The remainder tests 'the unspecified Action value for existing Product', the 'Add', and 'Update Action values for existing Products'.

For existing values, the update semantics are different depending on whether the sub-element is a "single-value", "multiple-language-value", or "multiple-value". In the following continuation of the scenario: Manufacturer and ManuPartNumber are a "single-value" elements; ShortDescription and LongDescription are "multiple-language-value" elements; and ObjectAttrbute and ProductAttachment are "multiple-value" elements.

It should be noted that the sub-elements of ObjectAttribute are not effected by the update semantics except as the entire ObjectAttribute is effected.

Catalog File	Intent of catalog document contents
E-Catalog	Utilize the default Product Action to update with new values for Manufacturer, ObjectAttribute, ShortDescription, and ProductAttachment.
F-Catalog	Utilize the default Product Action to change the given values for the Manufacturer, ObjectAttribute, ShortDescription, and for ProductAttachment given in E-Catalog.
G-Catalog	Utilize the default Product Action to add new values for ManuPartNumber, and LongDescription without changing existant values for other attributes.
H-Catalog	Utilize the Product Action 'Add' to change the given value for the Manufacturer, to add second values for the ShortDescription and ProductAttachment given in F-Catalog, and to add a new ObjectAttribute. For ShortDescription this means giving a new value with a different xml:lang value as given for this field in F-Catalog. For ObjectAttribute, this means creating a new attribute with a different AttributeID.
I-Catalog	With Product Action 'Update', replace the ObjectAttribute defined and modified in [E,F,G]-Catalog with a new one, and replace all the values for ShortDescription and ProductAttachment given in the previous catalog documents. Note for ShortDescription, ALL the values are replaced regardless of the xml:lang value of the existing entries. Note for ObjectAttribure, ALL the previous ObjectAttributes are replaced with a new one - not just the AttributeValue, the entire ObjectAttribute.

The above table outlines a series of documents which are incremental changes to the same catalog.

These are the corresponding documents:

A-Catalog.XML

```
<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogProvider>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
    <PricingInformation>
      <PriceCatalog>
        <PriceCatalogID>PC1</PriceCatalogID>
      </PriceCatalog>
    </PricingInformation>
    <SupplierAccountInformation>
      <SupplierAccount>
        <IsPublicAccount/>
        <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
      </SupplierAccount>
    </SupplierAccountInformation>
  </CatalogHeader>
  <CatalogSchema>
    <SchemaName>Custom</SchemaName>
    <SchemaCategory>
      <CategoryID>0</CategoryID>
      <CategoryName>Product</CategoryName>
```

```
                                    A-Catalog.XML
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>8</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>0</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>
       Communications, Computer Equipment, Peripherals,
       Components and Supplies
      </CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>51</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>0</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Paper Materials and Products</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>1</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>0</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Unknown</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>110</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>8</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Hardware and accessories</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>441</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>51</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Paper products</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>768</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>110</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Computers</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>2107</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>441</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Printing and writing paper</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>3859</CategoryID>
```

A-Catalog.XML

```
    <ParentCategoryRefList>
     <CategoryIDRef>768</CategoryIDRef>
    </ParentCategoryRefList>
    <CategoryName>Notebook computers</CategoryName>
  </SchemaCategory>
  <SchemaCategory>
    <CategoryID>12758</CategoryID>
    <ParentCategoryRefList>
     <CategoryIDRef>2107</CategoryIDRef>
    </ParentCategoryRefList>
    <CategoryName>Message note pads</CategoryName>
  </SchemaCategory>
</CatalogSchema>
<CatalogData>
  <Product>
    <ProductID>ACPPart4-III/500/12G</ProductID>
    <BaseProductNumber>ACPPart4</BaseProductNumber>
    <SchemaCategoryRefList>
      <CategoryIDRef>3859</CategoryIDRef>
    </SchemaCategoryRefList>
    <ProductIDExtension>III/500/12G</ProductIDExtension>
    <ProductIDStandard>Other</ProductIDStandard>
    <ProductName>
     Fleet M700 Notebook III/500/12G (English)
    </ProductName>
    <ProductName xml:lang='fr'>
     Fleet M700 Cahier III/500/12G (French)
    </ProductName>
    <UOM>
      <UOMCoded>EA</UOMCoded>
    </UOM>
    <ComparableUOM>
     <UOM><UOMCoded>CS</UOMCoded></UOM>
    </ComparableUOM>
    <ComparableUOMConversionFactor>
     1/36
    </ComparableUOMConversionFactor>
    <Manufacturer>Digital</Manufacturer>
    <ManuPartNumber>DEC700</ManuPartNumber>
    <LeadTime>5</LeadTime>
    <LeadTimeUOM><UOM><UOMCoded>WEE</UOMCoded></UOM></LeadTimeUOM>
    <ValidFrom>20011231</ValidFrom>
    <ValidUntil>20101231</ValidUntil>
    <CountryOfOrigin>
     <Country><CountryCoded>USA</CountryCoded></Country>
    </CountryOfOrigin>
    <MinOrder>1</MinOrder>
    <LotSize>1</LotSize>
    <ShortDescription xml:lang='en'>
     Fleet M700 PIII 500 12GB (English)
    </ShortDescription>
    <ShortDescription xml:lang='fr'>
     Fleet M700 PIII 500 12GB (French)
    </ShortDescription>
    <LongDescription xml:lang='en'>
```

```
                              A-Catalog.XML
     Fleet M700 Notebook with Pentium III 500mhz 12GB Harddisk
     (English)
   </LongDescription>
   <LongDescription xml:lang='fr'>
    Fleet M700 Cahier avec Pentium III 500mhz 12GB Disque Dur
    (French)
   </LongDescription>
   <CatalogContract Type='Supplier'>
     <CatalogContractID>DG-2002</CatalogContractID>
     <CatalogContractItemID>0001</CatalogContractItemID>
   </CatalogContract>
   <CatalogContract Type='Buyer'>
     <CatalogContractID>GeneralData-2002</CatalogContractID>
     <CatalogContractItemID>0053</CatalogContractItemID>
   </CatalogContract>
   <ProductAttachment>
     <AttachmentURL>11380.JPG</AttachmentURL>
     <AttachmentPurpose>PicName</AttachmentPurpose>
     <AttachmentMIMEType>image/jpeg</AttachmentMIMEType>
     <ShortDescription xml:lang='en'>
      Picture of the product (English)
     </ShortDescription>
     <ShortDescription xml:lang='fr'>
      Image du produit (French)
     </ShortDescription>
     <LongDescription xml:lang='en'>
      Picture of the product, duh (English)
     </LongDescription>
     <LongDescription xml:lang='fr'>
      Image du produit, doh (French)
     </LongDescription>
   </ProductAttachment>
   <ObjectAttribute>
     <AttributeID>MOSFac COD</AttributeID>
     <AttributeValue>56101708</AttributeValue>
   </ObjectAttribute>
   <ObjectAttribute>
     <AttributeID>Gross Weight</AttributeID>
     <AttributeUnit>
      <UOM><UOMCoded>KGM</UOMCoded></UOM>
     </AttributeUnit>
     <AttributeValue>2.3</AttributeValue>
   </ObjectAttribute>
   <ObjectAttribute>
     <AttributeID>Color</AttributeID>
     <AttributeValue xml:lang='en'>Blue (English)</AttributeValue>
     <AttributeValue xml:lang='fr'>Rouge (French)</AttributeValue>
   </ObjectAttribute>
 </Product>

 <Pricing>
   <ProductIDRef>ACPPart4-III/500/12G</ProductIDRef>
   <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
   <ProductPrice>
     <Amount>200.0000</Amount>
```

A-Catalog.XML

```
          <PriceType>0</PriceType>
          <Currency>
            <CurrencyCoded>USD</CurrencyCoded>
          </Currency>
          <UOM>
            <UOMCoded>EA</UOMCoded>
          </UOM>
        </ProductPrice>
      </Pricing>
    </CatalogData>
</ProductCatalog>
```

B-Catalog.XML

```
<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogProvider>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
  </CatalogHeader>

  <CatalogData>
    <Product>
      <Action Value='Replace'/>
      <ProductID>ACPPart4-III/500/12G</ProductID>
      <BaseProductNumber>ACPPart4-B-Value</BaseProductNumber>
      <SchemaCategoryRefList>
        <CategoryIDRef>12758</CategoryIDRef>
      </SchemaCategoryRefList>
      <ProductIDExtension>III/500/12G B-Value</ProductIDExtension>
      <ProductIDStandard>Other B-Value</ProductIDStandard>
      <ProductName>
       Fleet M700 Notebook III/500/12G B-Value (English)
      </ProductName>
      <ProductName xml:lang='de'>
       Fleet M700 Notizbuch III/500/12G (German)
      </ProductName>
      <UOM>
        <UOMCoded>EC</UOMCoded>
      </UOM>
      <ComparableUOM>
       <UOM><UOMCoded>C4</UOMCoded></UOM></ComparableUOM>
      <ComparableUOMConversionFactor>
       1/36000
      </ComparableUOMConversionFactor>
      <Manufacturer>Digital B-Value</Manufacturer>
```

B-Catalog.XML

```
<ManuPartNumber>DEC700 B-Value</ManuPartNumber>
<LeadTime>50</LeadTime>
<LeadTimeUOM><UOM><UOMCoded>DAY</UOMCoded></UOM></LeadTimeUOM>
<ValidFrom>20020202</ValidFrom>
<ValidUntil>20091231</ValidUntil>
<CountryOfOrigin>
 <Country><CountryCoded>GBR</CountryCoded></Country>
</CountryOfOrigin>
<MinOrder>10</MinOrder>
<LotSize>10</LotSize>
<ShortDescription xml:lang='en'>
 Fleet M700 PIII 500 12GB B-Value (English)
</ShortDescription>
<ShortDescription xml:lang='de'>
 Fleet M700 PIII 500 12GB (German)
</ShortDescription>
<LongDescription xml:lang='en'>
 Fleet M700 Notebook with Pentium III 500mhz 12GB Harddisk
 B-Value (English)
</LongDescription>
<LongDescription xml:lang='de'>
 Fleet M700 Notizbuch mit Pentium III 500mhz 12GB Festplatte
 (German)
</LongDescription>
<CatalogContract Type='Buyer'>
  <CatalogContractID>DG-2002-B-Value</CatalogContractID>
  <CatalogContractItemID>0002</CatalogContractItemID>
</CatalogContract>
<CatalogContract Type='Supplier'>
  <CatalogContractID>GeneralData-2002-B-Value</CatalogContractID>
  <CatalogContractItemID>0064</CatalogContractItemID>
</CatalogContract>
<ProductAttachment>
  <AttachmentURL>11380-B-Value.JPG</AttachmentURL>
  <AttachmentPurpose>PicName</AttachmentPurpose>
  <AttachmentMIMEType>image/jpeg</AttachmentMIMEType>
  <ShortDescription xml:lang='en'>
   Picture of the product B-Value (English)
  </ShortDescription>
  <ShortDescription xml:lang='de'>
   Bildnis des Produkts (German)
  </ShortDescription>
  <LongDescription xml:lang='en'>
   Picture of the product, duh B-Value (English)
  </LongDescription>
  <LongDescription xml:lang='de'>
   Bildnis des Produkts, dah (German)
  </LongDescription>
</ProductAttachment>
<ObjectAttribute>
  <AttributeID>MOSFac COD B-Value</AttributeID>
  <AttributeValue>8901</AttributeValue>
</ObjectAttribute>
<ObjectAttribute>
  <AttributeID>Net Weight</AttributeID>
```

```
                         B-Catalog.XML
       <AttributeUnit>
        <UOM><UOMCoded>GRM</UOMCoded></UOM>
       </AttributeUnit>
       <AttributeValue>2300</AttributeValue>
     </ObjectAttribute>
     <ObjectAttribute>
       <AttributeID>Color B-Value</AttributeID>
       <AttributeValue xml:lang='en'>
        Blue B-Value (English)
       </AttributeValue>
       <AttributeValue xml:lang='de'>
        Rot (German)
       </AttributeValue>
     </ObjectAttribute>
   </Product>

   <Pricing>
     <ProductIDRef>ACPPart4-III/500/12G</ProductIDRef>
     <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
     <ProductPrice>
       <Amount>200.0000</Amount>
       <PriceType>0</PriceType>
       <Currency>
         <CurrencyCoded>USD</CurrencyCoded>
       </Currency>
       <UOM>
         <UOMCoded>EA</UOMCoded>
       </UOM>
     </ProductPrice>
   </Pricing>
 </CatalogData>
</ProductCatalog>
```

```
                         C-Catalog.XML
<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogProvider>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
  </CatalogHeader>

  <CatalogData>
    <Product>
      <Action Value='Delete'/>
      <ProductID>ACPPart4-III/500/12G</ProductID>
```

C-Catalog.XML

```
      </Product>
    </CatalogData>
</ProductCatalog>
```

D-Catalog.XML

```
<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogProvider>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
  </CatalogHeader>

  <CatalogData>
    <Product>
      <ProductID>ACPPart4-III/500/12G</ProductID>
      <SchemaCategoryRefList>
        <CategoryIDRef>2107</CategoryIDRef>
      </SchemaCategoryRefList>
    </Product>
  </CatalogData>
</ProductCatalog>
```

E-Catalog.XML

```
<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogProvider>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
  </CatalogHeader>

  <CatalogData>
    <Product>
      <ProductID>ACPPart4-III/500/12G</ProductID>
      <SchemaCategoryRefList>
        <CategoryIDRef>2107</CategoryIDRef>
```

E-Catalog.XML

```
        </SchemaCategoryRefList>
        <Manufacturer>Digital E-Value</Manufacturer>
        <ShortDescription xml:lang='en'>
         Fleet M700 PIII 500 12GB E-Value (English)
        </ShortDescription>
        <ProductAttachment>
         <AttachmentURL>11380-E-Value.JPG</AttachmentURL>
         <AttachmentPurpose>PicName</AttachmentPurpose>
         <AttachmentMIMEType>image/jpeg</AttachmentMIMEType>
         <ShortDescription xml:lang='en'>
          Picture of the product E-Value (English)
         </ShortDescription>
        </ProductAttachment>
        <ObjectAttribute>
          <AttributeID>MOSFac COD E-Value</AttributeID>
          <AttributeValue>8901</AttributeValue>
        </ObjectAttribute>
      </Product>
  </CatalogData>
</ProductCatalog>
```

F-Catalog.XML

```
<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogProvider>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
  </CatalogHeader>

  <CatalogData>
    <Product>
      <ProductID>ACPPart4-III/500/12G</ProductID>
      <SchemaCategoryRefList>
        <CategoryIDRef>2107</CategoryIDRef>
      </SchemaCategoryRefList>
      <Manufacturer>Digital F-Value</Manufacturer>
      <ShortDescription xml:lang='en'>
       Fleet M700 PIII 500 12GB F-Value (English)
      </ShortDescription>
      <ProductAttachment>
       <AttachmentURL>11380-F-Value.JPG</AttachmentURL>
       <AttachmentPurpose>PicName</AttachmentPurpose>
       <AttachmentMIMEType>image/jpeg</AttachmentMIMEType>
       <ShortDescription xml:lang='en'>
        Picture of the product F-Value (English)
```

F-Catalog.XML

```
      </ShortDescription>
    </ProductAttachment>
    <ObjectAttribute>
      <AttributeID>MOSFac COD E-Value</AttributeID>
      <AttributeValue>8901 F-Value</AttributeValue>
    </ObjectAttribute>
  </Product>
  </CatalogData>
</ProductCatalog>
```

G-Catalog.XML

```
<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogProvider>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
  </CatalogHeader>

  <CatalogData>
    <Product>
      <ProductID>ACPPart4-III/500/12G</ProductID>
      <ManuPartNumber>DEC700</ManuPartNumber>
      <LongDescription xml:lang='en'>
       Fleet M700 Notebook with Pentium III 500mhz 12GB Harddisk
       G-Value(English)
      </LongDescription>
      <LongDescription xml:lang='fr'>
       Fleet M700 Cahier avec Pentium III 500mhz 12GB Disque Dur
       G-Value (French)
      </LongDescription>
    </Product>
  </CatalogData>
</ProductCatalog>
```

H-Catalog.XML

```
<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogProvider>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
```

H-Catalog.XML

```
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
  </CatalogHeader>

  <CatalogData>
    <Product>
      <Action Value='Add'/>
      <ProductID>ACPPart4-III/500/12G</ProductID>
      <Manufacturer>Digital H-Value</Manufacturer>
      <ShortDescription xml:lang='en'>
       Fleet M700 PIII 500 12GB H-Value (English)
      </ShortDescription>
      <ProductAttachment>
       <AttachmentURL>11380-H-Value.JPG</AttachmentURL>
       <AttachmentPurpose>PicName</AttachmentPurpose>
       <AttachmentMIMEType>image/jpeg</AttachmentMIMEType>
       <ShortDescription xml:lang='en'>Picture of the product H-Value
(English)
      </ShortDescription>
      </ProductAttachment>
      <ObjectAttribute>
        <AttributeID>MOSFac COD H-Value</AttributeID>
        <AttributeValue>8901 H-Value</AttributeValue>
      </ObjectAttribute>
    </Product>
  </CatalogData>
</ProductCatalog>
```

I-Catalog.XML

```
<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogProvider>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
  </CatalogHeader>

  <CatalogData>
    <Product>
      <Action Value='Update'/>
      <ProductID>ACPPart4-III/500/12G</ProductID>
      <SchemaCategoryRefList>
        <CategoryIDRef>2107</CategoryIDRef>
```

```
                                  I-Catalog.XML
      </SchemaCategoryRefList>
      <ShortDescription xml:lang='en'>
       Fleet M700 PIII 500 12GB I-Value (English)
      </ShortDescription>
      <ProductAttachment>
       <AttachmentURL>11380-I-Value.JPG</AttachmentURL>
       <AttachmentPurpose>PicName</AttachmentPurpose>
       <AttachmentMIMEType>image/jpeg</AttachmentMIMEType>
       <ShortDescription xml:lang='en'>
        Picture of the product I-Value  (English)
       </ShortDescription>
      </ProductAttachment>
      <ObjectAttribute>
        <AttributeID>MOSFac COD I-Value</AttributeID>
        <AttributeValue>8901 I-Value</AttributeValue>
      </ObjectAttribute>
    </Product>
  </CatalogData>
</ProductCatalog>
```

7   Implementation Considerations

7.1   ObjectAttribute Semantics

ObjectAttributes are an important, distinctive feature of xCBL ProductCatalog: they are an extensible, permitting new, important information about Products to be formally encoded in the catalog; the additional information they provide enable buyers to compare and contrast product offerings as well as to improve the accuracy of searches.

However, ObjectAttributes semantics are, at best, ill-defined. The CommerceOne eMPC implementation has made some implementation decisions based on our interpretation of the ObjectAttribute syntax and the implied semantics, which we believe are the most reasonable that can be made given the current state of the ObjectAttribute specification.

ObjectAttributes are features of classes, enabling individual instances to have particular values for those Attributes. An instance of an ObjectAttribute has information that must be utilized for two purposes: to identify the attribute of the Class(es) of which the current Product is an instance (AttributeID); to associate the given Value and UnitOfMeasure information with this Product (AttributeValue and AttributeUnit).

This is perhaps obvious, but the distinction between "instance" and "class" is important to keep in mind. In particular, since one may instantiate an ObjectAttribute without declaring it as a part of the schema, the instance of the ObjectAttribute may also be an implicit declaration of a new Attribute for the Class(es) of which the Product is an instance.

This "implicit declaration" of new Attributes has consequences: there is implicit information about the "type" of the new attribute derived from the given AttributeValue and AttributeUnit; deletion of the ObjectAttribute which initiated the declaration of the Attribute for the class doesn't imply the Attribute is no longer an attribute of the class, otherwise removal of a value on one Product of a given class(es) would eliminate Attributes of other Products of the same class(es), and hence their associated values for those Attributes.

The "type" information is nominal, especially for Attributes implicitly created by inclusion as a subelement of a Product. However, if an AttributeUnit is given, it is an indication of type: comparable units define an implicit type, for example LBR (pounds), KGM (kilograms), and other units of weight are comparable, and

hence imply the attribute is a "weight". We call these types "dimensions". Clearly, an Attribute with AttributeID of "Size" and AttributeUnit resolving to a "volume" is different from an Attribute with an identical ID (Size), but an AttributeUnit dimension of "area" or "weight".

Additionally, we expect that specifying an AttributeUnit implies that the AttributeValue must be a numeric value. Similarly, if a string is given as the AttributeValue, it would be an error to specify an AttributeUnit.

In the eMPC implementation, we utilize all of this information to determine the identity of the Attribute. That is, we allow there to be distinctly different attributes which have identical values for AttributeID, but vary significantly in their AttributeValue and/or AttributeUnit values:

```
<ObjectAttribute>
 <AttributeID>Weight</AttributeID>
 <AttributeValue>Heavy</AttributeValue>
<ObjectAttribute>

<ObjectAttribute>
 <AttributeID>Weight</AttributeID>
 <AttributeValue>20</AttributeValue>
<ObjectAttribute>
```

_____

```
<ObjectAttribute>
 <AttributeID>Weight</AttributeID>
 <!-- GM is the iso code for "gram per square meter" -->
 <AttributeUnit><UOM><UOMCoded>RP</UOMCoded></UOM></AttributeUnit>
 <AttributeValue>100</AttributeValue>
<ObjectAttribute>
```

_____

```
<ObjectAttribute>
 <AttributeID>Weight</AttributeID>
 <!-- KGM is the iso code for "kilogram" -->
 <AttributeUnit><UOM><UOMCoded>KGM</UOMCoded></UOM></AttributeUnit>
 <AttributeValue>100</AttributeValue>
<ObjectAttribute>

<ObjectAttribute>
 <AttributeID>Weight</AttributeID>
 <!-- LBR is the iso code for "pound" measure of weight -->
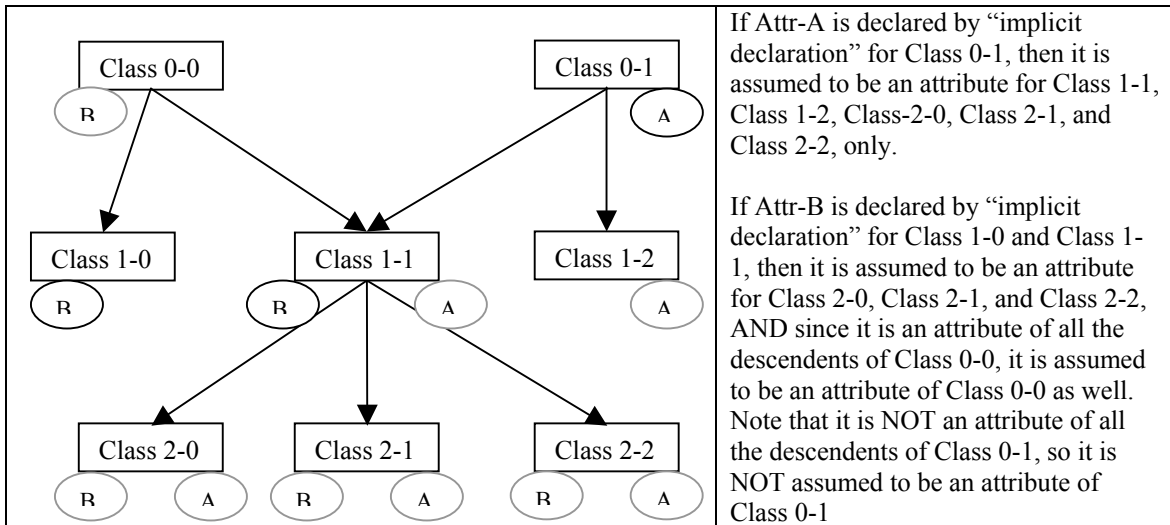 <AttributeUnit><UOM><UOMCoded>LBR</UOMCoded></UOM></AttributeUnit>
 <AttributeValue>100</AttributeValue>
<ObjectAttribute>
```

The above are resolved into three different attributes in eMPC: the first is an uninterpretable string value (uninterpretable computationally, not to a buyer); the second is also uninterpretable as it has no AttributeUnit specified, and hence isn't comparable with anything else; the third is a "pressure" measure, but is also used to quantify the "weight" of a paper stock in the metric system; the fourth is clearly a weight as is the fifth and they are comparable even though the UOM specified is different. Hence, the first two resolve into the "Weight-no-dimension" Attribute, the third is the "Weight-pressure-dimension" Attribute, and the fourth and fifth are "Weight-weight-dimension" Attribute.

Finally, the "implicit declaration" of new Attributes must be associated with at least one class. eMPC is conservative, and only associates the new attribute with the direct classes of the instance in which it appears and on all of the(ir) descendants.

Class 0-0 (R) → Class 1-0 (R), Class 1-1 (R)(A) Class 0-1 (A) → Class 1-1, Class 1-2 (A) Class 1-1 → Class 2-0 (R)(A), Class 2-1 (R)(A), Class 2-2 (R)(A)	If Attr-A is declared by "implicit declaration" for Class 0-1, then it is assumed to be an attribute for Class 1-1, Class 1-2, Class-2-0, Class 2-1, and Class 2-2, only. If Attr-B is declared by "implicit declaration" for Class 1-0 and Class 1-1, then it is assumed to be an attribute for Class 2-0, Class 2-1, and Class 2-2, AND since it is an attribute of all the descendents of Class 0-0, it is assumed to be an attribute of Class 0-0 as well. Note that it is NOT an attribute of all the descendents of Class 0-1, so it is NOT assumed to be an attribute of Class 0-1

However, should the same Attribute be instantiated in an instance of a different class (or classes), this information may be used to infer that the Attribute is inherited by these classes from their "least common ancestor", and hence the Attribute is associated with the ancestral class. This is ONLY done if ALL of the descendents of a class already have that Attribute.

In summary, when undeclared ObjectAttributes are utilized, the class they're associated with is determined by the direct class(es) of the Product in which they're instantiated. The identity of the Attribute is determined by the AttributeID as well as inferred from information included in the AttributeUnit and AttibuteValue fields.

7.2   Multiple Inheritance and "Multiple Taxonomies"

xCBL ProductCatalog supports multiple inheritance, both for the category and for an instance. XCBL ProductCatalog only permits one CatalogSchema to be defined, but this Schema can reflect "multiple taxonomies" as the following discussion shall illustrate.

For the purposes of this document, a *taxonomy* is meta-data which can be represented as a directed graph of labelled nodes. The edges of the graph are meant to represent an ancestor-descendent relationship, with *inheritance* (inheritance is discussed in more detail below). Each labelled node represents a classification that can be applied to any number of instances. For an instance to be classified as "belonging" to a classification means that it shares a certain collection of attributes so classified. Values for those attributes are comparable, though the values themselves may be different. For example, all lawn mowers have blades of some length, but the lengths themselves vary.

Typically, "multiple taxonomies" refers to there being a collection of "independent" taxonomies, where independent means there are no shared nodes between the graphs. It is also usually a property of taxonomies when discussed in the context of "multiple taxonomies" that each taxonomy has one and only one root.

xCBL ProductCatalog supports *multiple inheritance*, and it is important that this be well-understood as well. Basically, multiple inheritance means:

- that any node in a directed graph can have more than one direct ancestor, or "parent";
- that *all* of the attributes of all of the "parents" are conferred upon the child.

xCBL ProductCatalog only permits one CatalogSchema to be defined, but this Schema can reflect "multiple taxonomies" as defined above:

```
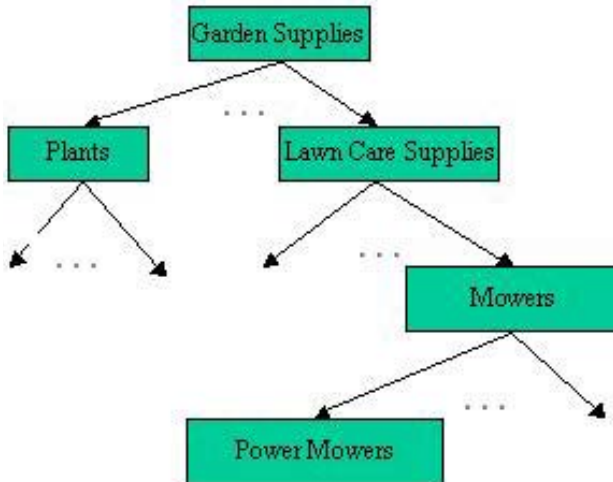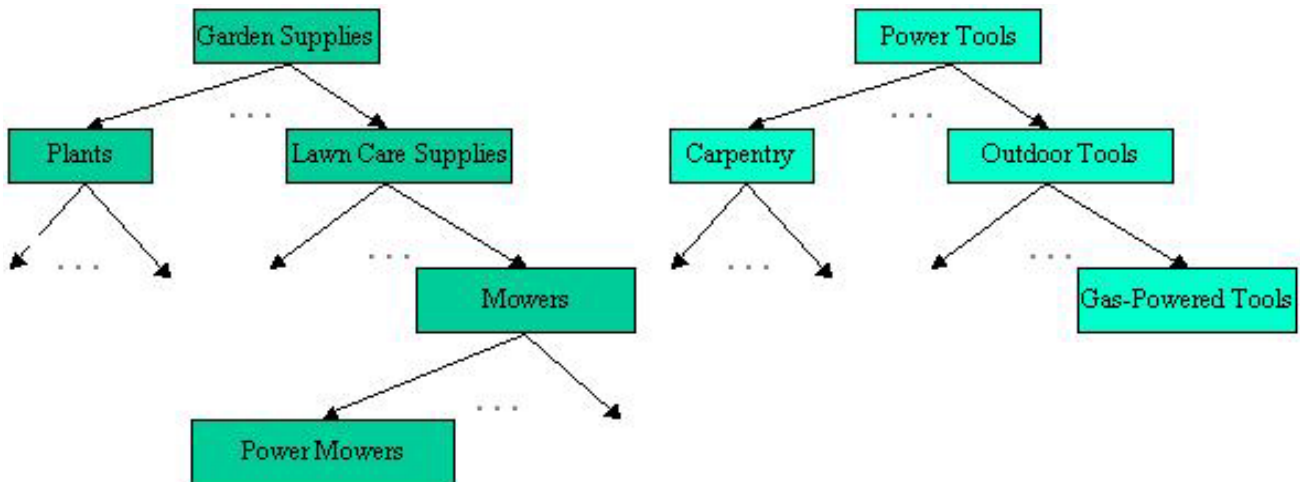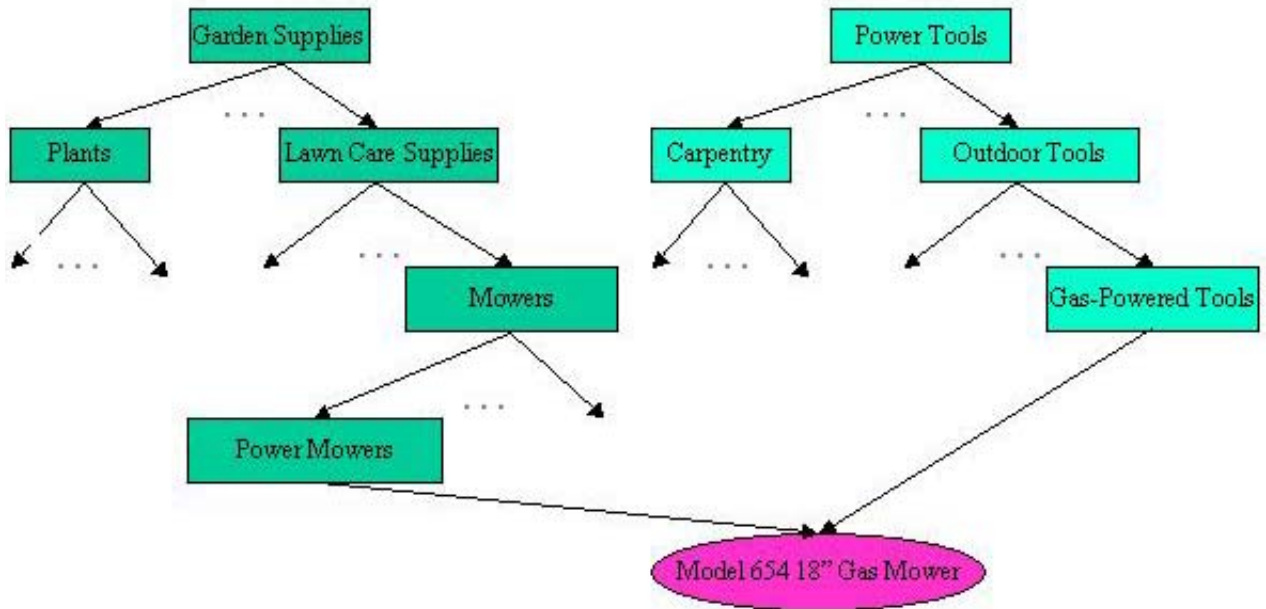<ProductCatalog>
  <CatalogHeader>
    <CatalogID>ACPsupplier4</CatalogID>
    <CatalogDate>20020408</CatalogDate>
    <CatalogProvider ProviderID='ACPsupplier4'>
      <Party>
        <PartyID>
          <Identifier>
            <Agency>
              <AgencyCoded>AssignedByMarketPlace</AgencyCoded>
            </Agency>
            <Ident>ACPsupplier4</Ident>
          </Identifier>
        </PartyID>
      </Party>
    </CatalogProvider>
    <PricingInformation>
      <PriceCatalog>
        <PriceCatalogID>PC1</PriceCatalogID>
      </PriceCatalog>
    </PricingInformation>
    <SupplierAccountInformation>
      <SupplierAccount>
        <SupplierAccountID></SupplierAccountID>
        <IsPublicAccount/>
        <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
      </SupplierAccount>
    </SupplierAccountInformation>
  </CatalogHeader>
  <CatalogSchema>
    <SchemaName>Custom</SchemaName>
```

```
    <!-- TOP Category for GARDEN SUPPLIES taxonomy -->
    <SchemaCategory>
      <CategoryID>A</CategoryID>
      <CategoryName>Garden Supplies</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>1A</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>A</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Lawn Care Supplies</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>10A</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>1A</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Mowers</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>100A</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>10A</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Power Mowers</CategoryName>
    </SchemaCategory>
    <!-- TOP Category for POWER TOOLS taxonomy -->
    <SchemaCategory>
      <CategoryID>B</CategoryID>
      <CategoryName>Power Tools</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>1B</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>B</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Outdoor Tools</CategoryName>
    </SchemaCategory>
    <SchemaCategory>
      <CategoryID>10B</CategoryID>
      <ParentCategoryRefList>
       <CategoryIDRef>1B</CategoryIDRef>
      </ParentCategoryRefList>
      <CategoryName>Gas Powered Tools</CategoryName>
    </SchemaCategory>
  </CatalogSchema>
  <CatalogData>
    <Product>
      <ProductID>M-654-18-G</ProductID>
      <SchemaCategoryRefList>
        <CategoryIDRef>100A</CategoryIDRef>
        <CategoryIDRef>10B</CategoryIDRef>
      </SchemaCategoryRefList>
      <ProductName>Model 654 18" Gas Powered Lawnmower</ProductName>
    </Product>
```

```
     <Pricing>
       <ProductIDRef>M-654-18-G</ProductIDRef>
       <PriceCatalogIDRef>PC1</PriceCatalogIDRef>
       <ProductPrice>
         <Amount>200.0000</Amount>
       </ProductPrice>
     </Pricing>
   </CatalogData>
</ProductCatalog>
```

To properly reflect the desired result of there being "multiple taxonomies", it is important that the individual taxonomies remain independent - that is, they must not share any CategoryID values, which is to say the names of the nodes must be unique to each taxonomy. Note that the CategoryID must be unique, however the CategoryName need not be. The CategoryName is solely used for display and the taxonomic information needed to distinguish to similarly named Categories is available in the context in which the CategoryName is displayed.

8 Formal Schema Definitions

8.1 Document Type Definition

The DTD schema for ProductCatalog can be found at http://www.xcbl.org/xcbl35/dtd/schemas.html

8.2 XDR Schema

The XDR schema for ProductCatalog can be found at http://www.xcbl.org/xcbl35/xdr/schemas.html

8.3 XSD Schema

The XSD Schema can be found at http://www.xcbl.org/xcbl35/xsd/schemas.html