
ICE: Information and Content Exchange Protocol

Basic ICE Specification

Version 2.0

2004 08 01

This version

<http://www.icestandard.org/Spec/SPEC-ICE-2.0Basic.pdf>

Latest version

<http://www.icestandard.org/Spec/SPEC-ICE2.0d.pdf>

Previous version

<http://www.icestandard.org/Spec/SPEC-ICE1.1.htm>

Editors:

Jay Brodsky, Tribune Media Services
Marco Carrer, Oracle Corporation
Bruce Hunt, Adobe Systems, Inc.
Dianne Kennedy, IDEAlliance
Daniel Koger, Independent Consultant
Richard Martin, Active Data Exchange
Laird Popkin, Warner Music Group
Adam Souzis, Independent Consultant

Copyright (c) International Digital Enterprise Alliance, Inc. [IDEAlliance] (1998, 1999, 2001, 2001, 2003, 2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to IDEAlliance, except as needed for the purpose of developing IDEAlliance specifications, in which case the procedures for copyrights defined in the IDEAlliance Intellectual Property Policy document must be followed, or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by IDEAlliance or its successors or assigns.

NO WARRANTY, EXPRESSED OR IMPLIED, IS MADE REGARDING THE ACCURACY, ADEQUACY, COMPLETENESS, LEGALITY, RELIABILITY OR USEFULNESS OF ANY INFORMATION CONTAINED IN THIS DOCUMENT OR IN ANY SPECIFICATION OR OTHER PRODUCT OR SERVICE PRODUCED OR SPONSORED BY IDEALLIANCE. THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN AND INCLUDED IN ANY SPECIFICATION OR OTHER PRODUCT OR SERVICE OF IDEALLIANCE IS PROVIDED ON AN "AS IS" BASIS. IDEALLIANCE DISCLAIMS ALL WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY ACTUAL OR ASSERTED WARRANTY OF NON-INFRINGEMENT OF PROPRIETARY RIGHTS, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. NEITHER IDEALLIANCE NOR ITS CONTRIBUTORS SHALL BE HELD LIABLE FOR ANY IMPROPER OR INCORRECT USE OF INFORMATION. NEITHER IDEALLIANCE NOR ITS CONTRIBUTORS ASSUME ANY RESPONSIBILITY FOR ANYONE'S USE OF INFORMATION PROVIDED BY IDEALLIANCE. IN NO EVENT SHALL IDEALLIANCE OR ITS CONTRIBUTORS BE LIABLE TO ANYONE FOR DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, COMPENSATORY DAMAGES, LOST PROFITS, LOST DATA OR ANY FORM OF SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL OR PUNITIVE DAMAGES OF ANY KIND WHETHER BASED ON BREACH OF CONTRACT OR WARRANTY, TORT, PRODUCT LIABILITY OR OTHERWISE.

IDEAlliance takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available. IDEAlliance does not represent that it has made any effort to identify any such rights. Information on IDEAlliance's procedures with respect to rights in IDEAlliance specifications can be found at the IDEAlliance website. Copies of claims of rights made available for publication, assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the President of IDEAlliance.

IDEAlliance requests interested parties to disclose any copyrights, trademarks, service marks, patents, patent applications, or other proprietary or intellectual property rights which may cover technology that may be required to implement this specification. Please address the information to the President of IDEAlliance.

Status of this Document

This document is an approved IDEAlliance Specification. It represents a significant step towards a stable specification suitable for widespread dissemination and implementation. It has been reviewed and approved by the ICE Authoring Group of IDEAlliance.

ICE 2.0 is the first major revision of the ICE Specification. As such, ICE 2.0 is not a compatible update to the ICE 1.0 specification. This update is a response to the implementation experience that has been gained over the past four years as well as the advancement in technology and W3C Recommendations. It differs from the ICE 1.0 and ICE 1.1 specifications in that it is specifically designed to support a Web Services model for syndication, has been modularized, incorporates XML Namespaces, and moves from an XML DTD to XML Schema.

As of this publication, the ICE Specification has been organized into a set of documents. This is one document in a set of documents (ICE Primer: Introduction and Overview, ICE Cookbook, Basic ICE Specification, Full ICE Specification, ICE Schemas and Scripts, and Guidelines to Extending the ICE Protocol) intended to jointly replace ICE 1.1. It has been developed by the IDEAlliance ICE Authoring Group. New documents may be added to this set over time.

The ICE Authoring Group and [IDEAlliance](#) recommend that implementations be updated to conform to the new ICE 2.0 Specification. The new specification embraces the latest Web technologies and W3C Recommendations. It provides added functionality that greatly enhances the usability of the protocol in a very wide range of syndication applications and can provide a substantial foundation for delivering syndication solutions in a Web Services environment.

Abstract

This document describes the Information and Content Exchange protocol for use by content syndicators and their subscribers. The ICE protocol defines the roles and responsibilities of Syndicators and Subscribers, defines the format and method of content exchange, and provides support for management and control of syndication relationships. We expect ICE to be useful in automating content exchange and reuse, both in traditional publishing contexts and in business-to-business relationships where the exchange eBusiness content must be reliably automated.

Table of Contents

Status of this Document	i
Abstract	i
1. Basic ICE Overview	3
2. A Basic ICE Scenario	3
2.1 Syndicator and Subscriber Set up a Business Agreement	3
2.2 Syndicator Makes “Catalog” Available	4
2.3 Subscriber “Gets” ICE Catalog	5
2.4 Subscriber “Gets” Content	5
3. Transport and Messaging	5
4. Catalogs and Subscription Management	6
4.1 Offers	6
4.1.1 Description	7
4.1.2 Delivery Policy	7
4.1.2.1 Delivery Rule	8
4.1.2.2 Transport	8
4.1.3 Example Basic ICE Offer	9
5. Packages and Delivery	10
5.1 Basic ICE Package Attributes	10
5.2 Package Elements	10
5.2.1 Group	11
5.2.2 Basic ICE Metadata	12
5.2.3 Add	12
5.2.4 Item	13
5.2.5 Item-Ref	13
5.2.4 Basic ICE Reference	13

1. Basic ICE Overview

Due to the nature of the content syndication business, it is important for ICE to support Subscriber implementations of varying levels of sophistication. In the most general case, a Subscriber is a sophisticated server implementation capable of not only sending ICE requests, but also receiving communications initiated by the Syndicator at any time, such as the "push" of new content. A Full ICE Subscriber has an ICE server running at all times. ICE also supports the concept of a *Basic ICE* implementation. This is an implementation where the Subscriber can initiate communicates (e.g. polling for updates) but does not have a persistent server available to receive messages. It is expected that a Basic ICE Subscriber is run on demand, either by a user or by an automated script. Thus, in a Basic ICE implementation, communication is out-of-band.

The Basic ICE level of conformance provides for very simple syndication functionality. In fact, all that Basic ICE enables is for the Syndicator to post messages to a URL where the Subscriber can "get" them. Basic ICE does not allow for subscription management capabilities. The Syndicator sends no messages to the Subscriber in Basic ICE. Basic ICE has no requirement for the Subscriber to establish a "listener" for push messages.

2. A Basic ICE Scenario

Let's look at a step-by-step example of a simple transaction between a Syndicator and a Subscriber. See Figure 2.1.

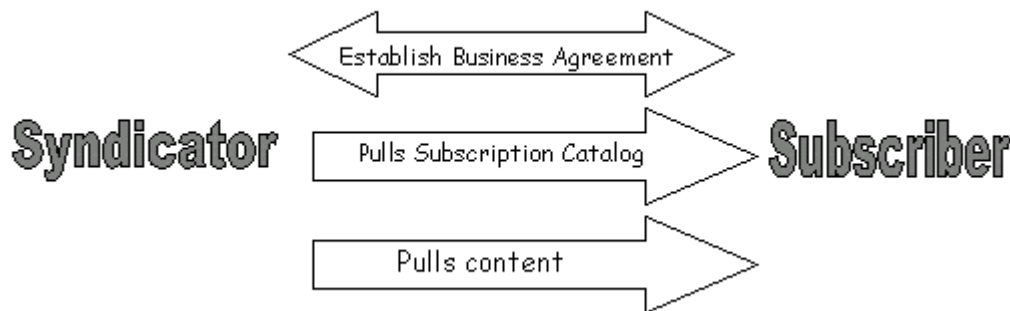


Figure 2.1 A Basic ICE Scenario

2.1 Syndicator and Subscriber Set up a Business Agreement

Syndication relationships begin with a business agreement. The business agreement negotiation happens *outside* ICE and can involve person-to-person discussion, legal review, and contracts. Because Basic ICE has no subscription management features,

these parameters may be discussed as part of the business agreement. We believe Basic ICE will most often be used in the scenario where the Syndicator is making content freely available to anyone. Hence subscription management features are not required.

2.2 Syndicator Makes “Catalog” Available

In ICE Version 1.0, the ICE protocol defined a set of messages to deliver a catalog of offers from the Syndicator to the Subscriber. In ICE 2.0, these messages were replaced by using the Version 2.0 content-syndication mechanism directly to deliver a package of offers where the `subscription-id="1"`. This ICE package should be placed at `http://<server-url>/get-package/1`.

The following example shows a package with a Basic ICE offer provided by a Syndicator.

```
<icedel:package
  xmlns:icedel="http://icestandard.org/ICE/V20/delivery"
  fullupdate="true"
  package-id="1"
  subscription-id="1">
  <icedel:add>
<icedel:metadata
  item-type="http://icestandard.org/ICE/V20/item-type/offer"
  content-type="text/xml"/>
<icedel:item>
  <icesub:offer
    xmlns:icesub="http://icestandard.org/ICE/V20/subscribe"
    offer-id="offID2"
    name="offName2">
    <icesub:description>
      headlines
    </icesub:description>
    <icesub:delivery-policy>
    <icesub:delivery-rule>
      <icesub:transport>
        <icesub:delivery-endpoint
          url=http://www.iceserver.com/gp/08292BC/>
        </icesub:delivery-endpoint>
      </icesub:transport>
    </icesub:delivery-policy>
    </icesub:offer>
  </icedel:item>
</icedel:add>
</icedel:package>
```

Note: In this offer everything is left to default including the mode on the delivery rule. ICE 2.0 was designed so that all defaults support Basic ICE. The transfer protocol will be “http:get” and the packaging will be “ice”. The pull will be made from the same location from which the catalog was pulled.

2.3 Subscriber “Gets” ICE Catalog

In ICE 1.0, messages were provided for retrieving the ICE catalog. In Basic ICE 2.0, the catalog is retrieved using the same mechanism as is used to retrieve other content. By convention, if the `subscription-id="1"` the package is made up of ICE offers.

2.4 Subscriber “Gets” Content

To retrieve content in Basic ICE, the Subscriber does an `HTTP:GET` on the URL specified in the offer that is selected by the Subscriber.

Basic ICE does not enforce complex delivery rules. All content is *pulled from* the Syndicator in Basic ICE. The Subscriber pulls ICE content encoded in an ICE/SOAP format. This is simply an XML file where SOAP and ICE are used to wrap the content and ICE status codes that may be sent by the Syndicator. The Subscriber can treat this package as it would treat any XML-encoded file. It is not necessary that the Subscriber have SOAP capabilities to receive content or ICE status codes.

An example of what the Subscriber gets from the Syndicator’s URL follows.

```
<?xml version="1.0" ?>
<env:Envelope
xmlns:env='http://www.w3.org/2002/12/soapenvelope'>
  <env:Header>
<icemes:header
  xmlns:icemes="http://icestandard.org/ICE/2002/message"
timestamp="2003-03-03T00:00:00Z"
message-id="m0056">
  <icemes:sender name="mycompany"
    role="http://icestandard.org/ICE/2002/role/syndicator"
    sender-id="http://www.xxyz.org"/>
</icemes:header>
  </env:Header>
  <env:Body>
<icedel:package
  xmlns:icedel="http://icestandard.org/ICE/2002/delivery"
    fullupdate="true"
    package-id="12"
    subscription-id="3">
  <icedel:add>
<icedel:item-ref>
  <icedel:reference
    url="http://mysite.com/text.htm"/>
</icedel:item-ref>
  </icedel:add>
  </icedel:package>
  </env:Body>
</env:Envelope>
```

3. Transport and Messaging

Two entities are involved in ICE transport and messaging. The Syndicator produces content that is pulled by Subscribers. The philosophy behind Basic ICE is to enable a very simple form of syndication that does not require sophisticated processing by either the Syndicator or the Subscriber. In Basic ICE, all messages/packages from the Syndicator are accessible from a URL and the Subscriber uses an `HTTP:GET` to retrieve messages. This implies that the Subscriber does not have to have a SOAP-enabled server to receive “push” content from the Syndicator. Rather the Subscriber always pulls from the Syndicator. Because there are no end points for either the Syndicator or the Subscriber, a WSDL script for the Basic ICE does not exist.

4. Catalogs and Subscription Management

Basic ICE was designed for publishing information that is available to any interested party. Basic ICE does not support subscription management. So, with Basic ICE, there are no `<icesub:subscribe` or `<icesub:subscription` elements. The Subscriber cannot use `<cancel>`. Catalogs of syndication offers, that were uniquely identified by an element type in ICE 1.0, are simply a special type of `<icedel:package>` that contains offers (the ICE catalog) and by convention is identified by the `subscription-id="1"`.

4.1 Offers

The structure of an offer for Basic ICE is shown in Figure 4.1. Most of the attributes on offer and most of the elements that make up offer are only used in Full ICE and are not shown here. This Chapter will only document those that apply to Basic ICE.

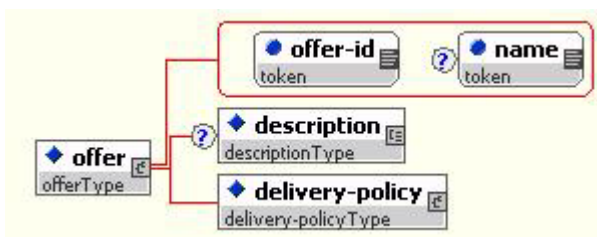


Figure 4.1 Basic ICE Offer Structure

ICE was written so that all defaults are set to support basic ICE. Hence offers in Basic ICE are themselves very basic.

An `<icesub:offer>` has the following attributes that will be used for Basic ICE:

- **offer-id**
Required. This is a string. It is an identifier that **MUST** be unique across all catalog offers from a Syndicator. Its function is to clearly identify this offer from all other catalog offers made by a Syndicator.
- **name**
Optional. This is a string. It is a name that may be used to distinguish subscriptions and offers from other subscriptions or offers. Its intended use is to provide a readable short description of the offer such as, "Julia Child's Contemporary French Cooking Column".

An offer is made up of a several elements that will be used for Basic ICE. These include `<icesub:description>` and `<icesub:delivery-policy>`.

4.1.1 Description

This element is a text field and facilitates the entry of a description of the offer. This simple element is shown in Figure 4.2. Note that the `xml:lang=` attribute on `<icesub:text>` enables the specification of language for the text field within `<icesub:description>`.

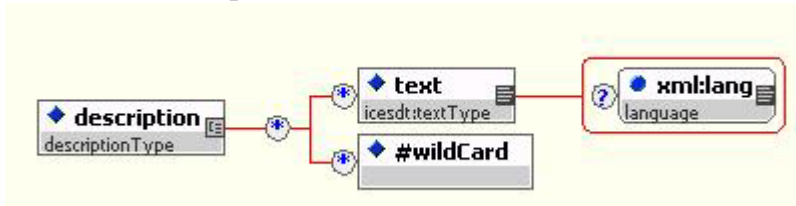


Figure 4.2 Description element structure

The description is useful in Basic ICE because it can help Subscribers understand whether they want to pull the content of the subscription.

4.1.2 Delivery Policy

Each subscription offer has one delivery-policy that in turn contains one or more delivery rules. Only certain elements and attributes of a delivery policy make sense for Basic ICE. These structures are shown in the reduced diagram of `<icesub:delivery-policy>`. See Figure 4.3.

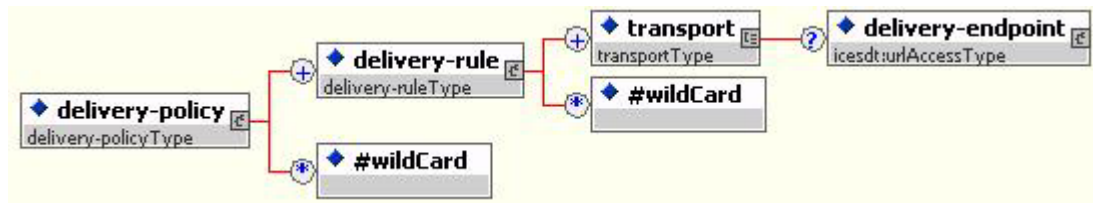


Figure 4.3 Delivery-policy Structure

4.1.2.1 Delivery Rule

Each `<icesub:delivery-policy>` is made up of one or more delivery rules. Only certain elements and attributes of a delivery rule make sense for Basic ICE. These structures are shown in the reduced diagram of `<icesub:delivery-rule>` in the Figure 4.4. Each delivery rule in Basic ICE is made up of one or more `<icesub:transport>` elements.

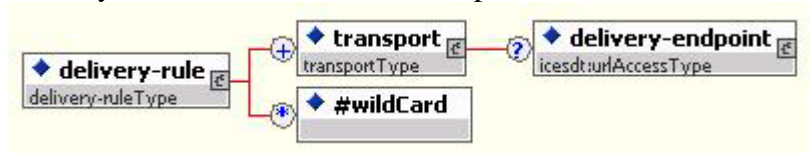


Figure 4.4 Basic ICE Delivery-rule Structure

4.1.2.2 Transport

The `<icesub:delivery-rule>` is made up of one or more `<icesub:transport>`. In Basic ICE, transports are specified when the Syndicator makes an offer. This element provides a mechanism for the Syndicator to indicate the delivery transports for the `<icesub:offer>`. You can see the makeup of a Basic ICE `<icesub:transport>` in the Figure 4.5:

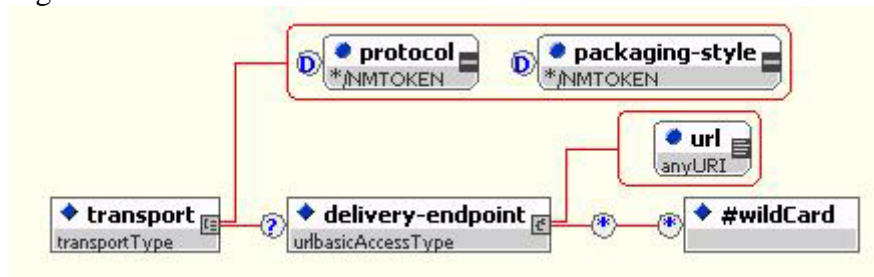


Figure 4.5 Basic ICE Transport Structure

The `<icesub:transport>` is made up of an optional delivery endpoint. See Figure 4.5. In Basic ICE, when the Syndicator is offering content in “pull” mode, a delivery endpoint for that pull must be specified by using `<icesub:delivery-endpoint>`.

The `<icesub:delivery-endpoint>` has 1 attribute that makes sense for Basic ICE. These include:

- **url**
Required. This attribute specifies the URL for push delivery. The datatype is `anyURI`.

4.1.3 Example Basic ICE Offer

Basic ICE is a simple “pull” of content by the Subscriber. The following example shows a Basic ICE offer, using the built-in defaults for Basic ICE. Syndicator transports were left to default. Delivery settings are not provided because the assumption is that the protocol will be HTTP:GET and the packaging will be “ice” and The pull will be made from the URL specified by the endpoint.

```
<icedel:package
  xmlns:icedel="http://icestandard.org/ICE/V20/delivery"
  package-id="1"
  subscription-id="1">
  <icedel:add>
    <icedel:metadata item-
type="http://icestandard.org/ICE/V20/item-type/offer"
content-type="text/xml"/>
    <icedel:item>
      <icesub:offer
xmlns:icesub="http://icestandard.org/ICE/V20/subscribe"
name="offName2"
offered="off2">
        <icesub:description>
          new stories headlines and abstracts
        </icesub:description>
        <icesub:delivery-policy>
          <icesub:delivery-rule>
            <icesub:transport>
              <icesub:delivery-endpoint
url="http://www.iceserver.com/gp/08292BC"/>
            </transport>
          </icesub:delivery-rule>
        </icesub:delivery-policy>
      </icesub:offer>
    </icedel:item>
  </icedel:add>
</icedel:package>
```

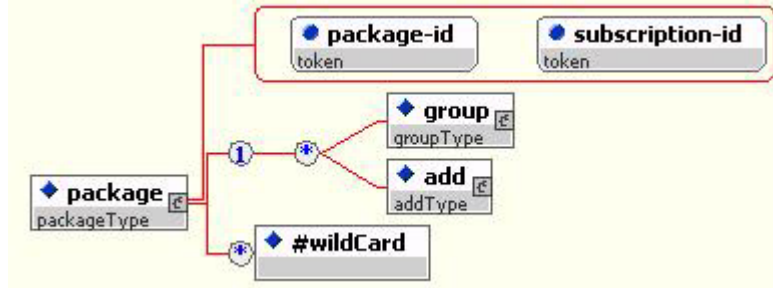
Change this example to have the URL to pull from

NOTE: You can recognize that this package contains a subscription offer because the subscription-id is set to “1”.

5. Packages and Delivery

Basic ICE only supports the delivery of packages containing full updates of content. In Basic ICE, the delivery is simply the act of the Syndicator placing the content in a SOAP/ICE XML document at the URL specified in the offer. .

The ICE package is made up of three elements. Even though the ICE package is made up of three elements (group, add, remove), only two of these (group and add) make sense for Basic ICE. In Basic ICE it is reasonable that the Syndicator delivers a group of items or adds a single item to the subscription content. Since each package contains a full update in Basic ICE, removing items will never be used. The ICE package, reduced for Basic



ICE is shown in Figure 5.1.

Figure 5.1 Basic ICE Package Structure

5.1 Basic ICE Package Attributes

There are several attributes on package that are meaningful for Basic ICE:

- **package-id**
Required. Identifies the package within the scope of a subscription. The Syndicator assigns the `package-id`.
- **subscription-id**
Required. In Basic ICE, the subscription-id is the unique id of the content feed and is used by all subscribers. The Syndicator assigns the `subscription-id`.

5.2 Package Elements

The ICE package is made up of 3 elements. An ICE `<icedel:package>` describes a set of content operations: additions, removals, and a group of additions and/or removals. See Figure 5.2. The remove operation is specified using the `<icedel:remove-item>` element is not used in Basic ICE because this functionality facilitates delivery of incremental updates. The content additions contain the content that needs to be added or updated and are specified using the `<icedel:item>` and `<icedel:item-ref>` elements. The `<icedel:group>` element allows the Syndicator to associate the content specified using the `<icedel:item>` elements together. For example, in the syndication of restaurant reviews, each review may consist of different types of content such as an HTML file and

two graphic files. These three files could be contained within three `<icedel:item` elements and grouped together in an ICE `<icedel:group` as a single restaurant review. Likewise, unrelated content can be specified in a `<icedel:package` by just using the `<icedel:add` and then `<item` elements without an intervening `<icedel:group`. The `<icedel:item` and `<icedel:item-ref` elements distinguish themselves by the way they contain the content. The `<icedel:item` element is used to contain content directly in the delivered content. The `<icedel:item-ref` element is used to distribute an indirect reference to the actual content. Note that the `#wildcard` allows for insertion of content from any namespace.

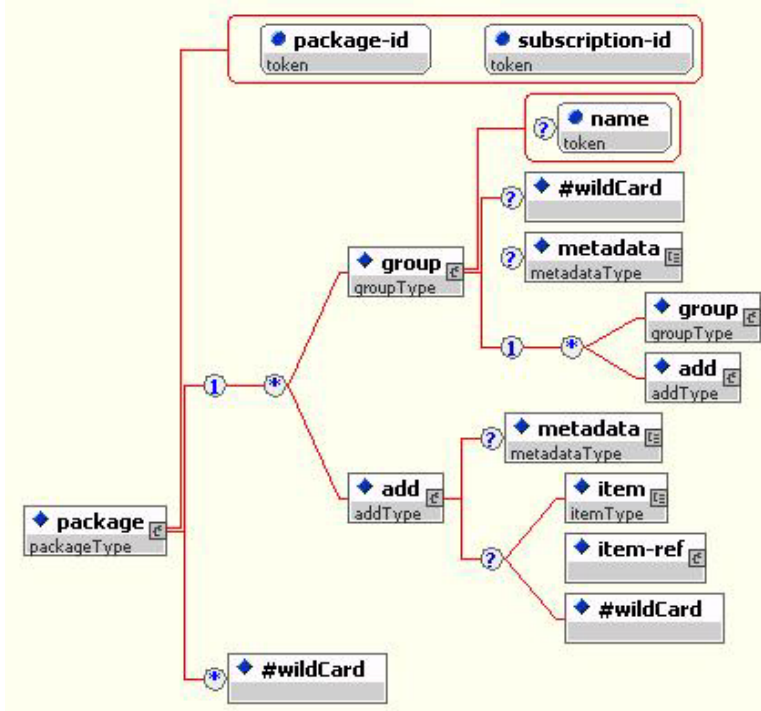


Figure 5.2 Package Elements for Basic ICE

5.2.1 Group

The `<icedel:group` is a container element that can be used to group content items being added or removed. It also enables the attachment of metadata to a group of content items.

Attributes on `<icedel:group` include:

- name**
Optional. This attribute specifies a name for the item group that can be used to identify that group within a package.

5.2.2 Basic ICE Metadata

The `<icedel:metadata` element enables the entry of metadata on `<icedel:group` and `<icedel:add` by using its attributes and a description field. See Figure 5.3.



Figure 5.3 Basic ICE Metadata

Because the attributes on the metadata element are set to support Basic ICE, only the content-type and item-type attributes may be used.

- **content-type**
Optional. This attribute enables the specification of the type of content such as “news.”
- **item-type**
Optional. This attribute specifies a URI that identifies what type of item this is. For example the value of item-type may be <http://icestandard.org/ICE/V20/item-type/offer>.

Note: For Basic ICE, the requirement is that content will be used in its entirety and that the content may not be edited.

5.2.3 Add

The `<icedel:add` element is used to add new content according to the delivery policy of the subscription. It enables the attachment of metadata to the content being added. The `<icedel:add` enables content to be directly included in the message by using the `<icedel:item` element, an indirect reference to content using `<icedel:item-ref` mechanism.

The structure of `<icedel:add` is shown in Figure 5.4.

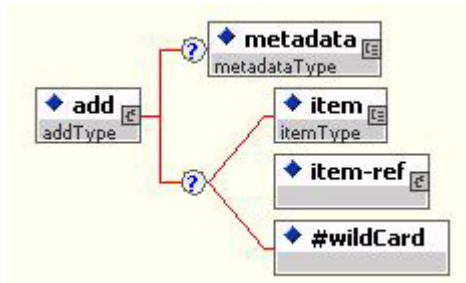


Figure 5.4 Add Element Structure

5.2.4 Item

The `<icedel:item` element directly carries content from the Syndicator to the Subscriber. The `<icedel:item` can carry the ICE element `<icesub:offer`.

5.2.5 Item-Ref

The `<icedel:item-ref` element references Syndicator content. The `<icedel:item-ref` structure is shown in Figure 5.5. It is made up of a single `<icedel:reference` element. This means that for each reference, an `<item-ref` element must be used.



Figure 5.5 Item-ref Structure

The `<item-ref` element has two attributes:

retrieve-after

Optional. This attribute specifies a time after which the item can be retrieved. It is specified in the `icesdt:dateTime` format.

name

Optional. This attribute specifies the item name that can be used as a transient identifier within a group or add.

5.2.4 Basic ICE Reference

The `<icedel:reference` element is used to reference the content of the `<icedel:item-ref` element. The reference element is empty (with the exception of any wildcard content). The attributes carry the information for this element. See Figure 5.6.

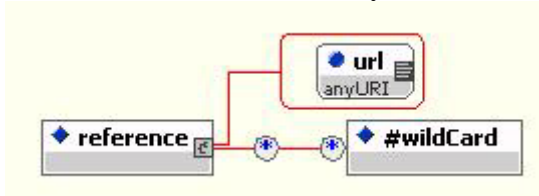


Figure 5.6 Basic ICE Item-ref Reference Structure

The `<icedel:reference` element has four attributes:

url

Required. This attribute specifies the URL from which the content can be retrieved.