

---

# ICE: Information and Content Exchange Protocol

## Schemas and Scripts

Version 2.0

2004 08 01

**This version**

<http://www.icestandard.org/Spec/SPEC-ICE-2.0Schemas.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. Overview of ICE Schemas.....	3
2. ICE Simple Datatype Schema.....	4
3. ICE Message Schema .....	9
4. ICE Delivery Schema .....	12
5. ICE Subscribe Schema.....	18
6. Overview of ICE WSDL Samples .....	26
7. Sample Full ICE Syndicator WSDL Script.....	27
8. Sample Full ICE Subscriber WSDL Scripts .....	34

# 1. Overview of ICE Schemas

The ICE 2.0 Specification is made up of four (4) W3C Schema Definitions (XSD). Together these schemas define the datatypes and messages that make up the ICE Protocol. ICE XML Schemas can be found on the ICE Standard website:

<http://www.icestandard.org/Spec/V20/schema/icesimpledatatypes20040801.xsd>

<http://www.icestandard.org/Spec/V20/schema/ice-message20040801d>

<http://www.icestandard.org/Spec/V20/schema/ice-subscribe20040801d>

<http://www.icestandard.org/Spec/V20/schema/ice-delivery20040801d>

## 2. ICE Simple Datatype Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://icestandard.org/ICE/V20/simpledatatypes"
  xmlns:icesdt="http://icestandard.org/ICE/V20/simpledatatypes"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:annotation>
    <xs:documentation>
      We define our own schema for the XML namespace because the
      canonical one available at http://www.w3.org/XML/1998/namespace
      doesn't include xml:base and is not in sync with the Recommended
      version of W3C Schema 1.0
    </xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="xml.xsd"/>

  <!--
  | attlist.genericmetadata
  -->
  <xs:attributeGroup name="attlist.genericmetadata">
    <xs:annotation>
      <xs:documentation>
        changes from 1.1: moved content-transfer-encoding to item only
      </xs:documentation>
    </xs:annotation>
    <xs:attribute name="atomic-use"
      type="xs:boolean" default="false"/>
    <xs:attribute name="editable" type="xs:boolean"
      default="false"/>
    <xs:attribute name="ip-status" type="xs:token"/>
    <xs:attribute name="license" type="xs:token"/>
    <xs:attribute name="rights-holder" type="xs:token"/>
    <xs:attribute name="show-credit" type="xs:boolean"/>
    <xs:attribute name="item-type" type="icesdt:item-types"
      default="http://icestandard.org/ICE/V20/item-type/undefined"/>
  </xs:attributeGroup>

  <!--
  | ice-compliance
  | Basic ICE compliance:
  |   http://icestandard.org/ICE/V20/syndicator/basic
  | Full ICE compliance:
  |   http://icestandard.org/ICE/V20/syndicator/full
  -->
  <xs:simpleType name="compliance-types">
    <xs:annotation>
```

```
<xs:documentation>
  a URI that names an ICE compliance level.
</xs:documentation>
</xs:annotation>
<xs:union memberTypes="xs:anyURI">
<xs:simpleType>
  <xs:restriction base="xs:anyURI">
    <xs:enumeration
      value="http://icestandard.org/ICE/V20/syndicator/basic"/>
    <xs:enumeration
      value="http://icestandard.org/ICE/V20/syndicator/full"/>
    <xs:enumeration
      value="http://icestandard.org/ICE/V20/itemtype/undefined"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
</xs:union>
</xs:simpleType>

<!--
  |
  | item-types (feature)
  |
-->
<xs:simpleType name="item-types">
  <xs:annotation>
    <xs:documentation>
      a URI that defines the content of an item.
    </xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="xs:anyURI">
  <xs:simpleType>
    <xs:restriction base="xs:anyURI">
      <xs:enumeration
        value="http://icestandard.org/ICE/V20/item-type/undefined"/>
      <xs:enumeration
        value="http://icestandard.org/ICE/V20/item-type/offer"/>
      <xs:enumeration
        value="http://icestandard.org/ICE/V20/item-type/rss0.91"/>
      <xs:enumeration
        value="http://icestandard.org/ICE/V20/item-type/rss0.92"/>
      <xs:enumeration
        value="http://icestandard.org/ICE/V20/item-type/rss1.0"/>
      <xs:enumeration
        value="http://icestandard.org/ICE/V20/item-type/rss2.0"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType>
    <xs:restriction base="xs:anyURI"/>
  </xs:simpleType>
</xs:union>
</xs:simpleType>
```

```
<!--
  |
  | dateTime
  |
-->
<xs:simpleType name="dateTime">
  <xs:annotation>
    <xs:documentation>
      the pattern here expresses the restriction that
      datetimes in ICE must be in the UTC time zone
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:dateTime">
    <xs:pattern value=".*Z"/>
  </xs:restriction>
</xs:simpleType>
```

```
<!--
  |
  | time
  |
-->
<xs:simpleType name="time">
  <xs:annotation>
    <xs:documentation>
      the pattern here expresses the restriction that
      times in ICE must be in the UTC time zone
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:time">
    <xs:pattern value=".*Z"/>
  </xs:restriction>
</xs:simpleType>
```

```
<!--
  |
  | duration
  |
-->
<xs:simpleType name="duration">
  <xs:annotation>
    <xs:documentation>
      the pattern here expresses the restriction that
      durations in ICE must only include seconds
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:duration">
    <xs:pattern value="PT\d*\.\?\d*S"/>
  </xs:restriction>
</xs:simpleType>
```



```
<!--
  |
  | parameters
  |
-->
<xs:element name="parameters">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
</xs:element>

<!--
  |
  | package-sequence-stateType
  |
-->
<xs:simpleType name="package-sequence-stateType">
  <xs:union memberTypes="xs:token">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="ICE-INITIAL"/>
        <xs:enumeration value="ICE-ANY"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:token"/>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>

<!--
  |
  | textType
  |
-->
<xs:complexType name="textType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute ref="xml:lang"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

```
<!--
|
| urlAccessType
|
-->
<xs:complexType name="urlAccessType">
  <xs:annotation>
    <xs:documentation>
      changes from 1.1:
      * removed the (FIXED) ice-element attribute
      * added link-only attribute
      * remove access-window
    </xs:documentation>
  </xs:annotation>
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="url" use="required" type="xs:anyURI"/>
  <xs:attribute name="username" type="xs:token"/>
  <xs:attribute name="password" type="xs:token"/>
  <xs:attribute name="authentication-scheme">
    <xs:simpleType>
      <xs:union memberTypes="xs:string">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="basic"/>
            <xs:enumeration value="digest"/>
          </xs:restriction>
        </xs:simpleType>
        <xs:simpleType>
          <xs:restriction base="xs:string"/>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

</xs:schema>
```

## 3. ICE Message Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
  targetNamespace = "http://icestandard.org/ICE/V20/message"
  xmlns:icemsg = "http://icestandard.org/ICE/V20/message"
  xmlns:xs = "http://www.w3.org/2001/XMLSchema"
  xmlns:icesdt = "http://icestandard.org/ICE/V20/simpledatatypes"
  elementFormDefault = "qualified">

  <import namespace =
    "http://icestandard.org/ICE/V20/simpledatatypes"
    schemaLocation = "ice-simpledatatypes.xsd"/>

  <!--
  |
  | header
  |
  -->
  <element name = "header">
    <complexType>
      <sequence>
        <element name = "sender" type = "icemsg:senderType"/>
        <element name = "receiver"
          type = "icemsg:receiverType" minOccurs = "0"/>
        <element ref = "icemsg:user-agent" minOccurs = "0"/>
        <any namespace = "##other" processContents = "lax"
          minOccurs = "0" maxOccurs = "unbounded"/>
      </sequence>
      <attribute name = "timestamp" use = "required"
        type = "icesdt:dateTime"/>
      <attribute name = "message-id" use = "required"
        type = "token"/>
      <attribute name = "response-to"
        type = "token"/>
      <anyAttribute namespace = "##other" processContents = "lax"/>
    </complexType>
  </element>

  <!--
  |
  | receiverType
  |
  -->

  <complexType name = "receiverType">
    <attribute name = "name" use = "required" type = "string"/>
    <attribute name = "receiver-id" use = "required" type = "anyURI"/>
    <attribute name = "role">
      <simpleType>
        <restriction base = "NMTOKENS">
          <enumeration value = "subscriber"/>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>
</schema>
```

```

        <enumeration value = "syndicator"/>
    </restriction>
</simpleType>
</attribute>
<attribute name = "compliance-level" default = "basic"
    type = "icesdt:compliance-types"/>
<anyAttribute namespace = "##other" processContents = "lax"/>
</complexType>

<!--
|
| senderType
|
-->

<complexType name = "senderType">
    <attribute name = "name" use = "required" type = "string"/>
    <attribute name = "role">
        <simpleType>
            <restriction base = "NMTOKENS">
                <enumeration value = "subscriber"/>
                <enumeration value = "syndicator"/>
            </restriction>
        </simpleType>
    </attribute>
    <attribute name = "sender-id" use = "required" type = "anyURI"/>
    <attribute name = "location" type = "anyURI"/>
    <attribute name = "compliance-level" default = "basic" type =
"icesdt:compliance-types"/>
    <anyAttribute namespace = "##other" processContents = "lax"/>
</complexType>

<!--
|
| user-agent
|
-->

    <element name = "user-agent" type = "icesdt:textType"/>

<!--
| ping request message
-->

<element name = "ping">
    <complexType/>
</element>

<!--
|
| OK simple confirmation response message
|
-->

<element name = "OK">
    <complexType/>

```

```
</element>

<!--
  |
  |   Status-code
  |
  -->

<element name = "status-code">
  <complexType>
    <sequence>
      <any namespace = "##other" processContents = "lax" minOccurs =
"0" maxOccurs = "unbounded"/>
    </sequence>
    <attribute name = "code" use = "required"
      type = "positiveInteger"/>
    <attribute name = "message-id" use = "required" type = "token"/>
    <attribute name = "subscription-id" use = "required"
      type = "token"/>
    <attribute name = "location" type = "anyURI"/>
    <attribute name = "duration"
      type = "icesdt:duration"/>
    <anyAttribute namespace = "##other" processContents = "lax"/>
  </complexType>
</element>
</schema>
```

## 4. ICE Delivery Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
  targetNamespace = "http://icestandard.org/ICE/V20/delivery"
  xmlns:icedel = "http://icestandard.org/ICE/V20/delivery"
  xmlns:xs = "http://www.w3.org/2001/XMLSchema"
  xmlns:icesdt = "http://icestandard.org/ICE/V20/simplifiedatatypes"
  elementFormDefault = "qualified">
<import namespace = http://icestandard.org/ICE/V20/simplifiedatatypes
  schemaLocation = "ice-simplifiedatatypes.xsd"/>
<import namespace = http://www.w3.org/XML/1998/namespace
  schemaLocation = "xml.xsd"/>

<attributeGroup name = "attlist.item">
  <attribute name = "content-filename" type = "xs:token"/>
  <attribute name = "content-type" default = "application/octet-stream"
    type = "xs:token"/>
</attributeGroup>

<group name = "cm.package">
  <choice>
    <choice minOccurs = "0" maxOccurs = "unbounded">
      <element name = "group" type = "icedel:groupType"/>
      <element name = "add" type = "icedel:addType"/>
      <element ref = "icedel:remove-item"/>
    </choice>
  </choice>
</group>

<!--
  | get-packages      |
-->

<element name = "get-packages">
  <complexType>
    <sequence>
      <element ref = "icedel:get-package" maxOccurs = "unbounded"/>
      <any namespace = "##other" processContents = "lax"
        minOccurs = "0" maxOccurs = "unbounded"/>
    </sequence>
    <anyAttribute namespace = "##other" processContents = "lax"/>
  </complexType>
</element>

<!--
  |
  | get-package
  |
-->

<element name = "get-package" type = "icedel:get-packageType"/>
```

```
<!--  
  |  
  | get-packageType  
  |  
-->
```

```
<complexType name = "get-packageType">  
  <sequence>  
    <element ref = "icesdt:parameters" minOccurs = "0"/>  
  </sequence>  
  <attribute name = "current-state" type = "xs:token"/>  
  <attribute name = "subscription-id" use = "required" type =  
"xs:token"/>  
  <anyAttribute namespace = "##other" processContents = "lax"/>  
</complexType>
```

```
<!--  
  |  
  | packages  
  |  
-->
```

```
<element name = "packages">  
  <complexType>  
    <sequence minOccurs = "0" maxOccurs = "unbounded">  
      <element ref = "icedel:package"/>  
      <any namespace = "##other" processContents = "lax" minOccurs =  
"0" maxOccurs = "unbounded"/>  
    </sequence>  
    <anyAttribute namespace = "##other" processContents = "lax"/>  
  </complexType>  
</element>
```

```
<!--  
  |  
  | package  
  |  
-->
```

```
<element name = "package" type = "icedel:packageType"/>
```

```
<!--  
  |  
  | packageType  
  |  
-->
```

```
<complexType name = "packageType">  
  <sequence>  
    <group ref = "icedel:cm.package"/>  
    <any namespace = "##other" processContents = "lax" minOccurs = "0"  
maxOccurs = "unbounded"/>  
  </sequence>  
  <attribute name = "package-id" use = "required" type = "xs:token"/>  
  <attribute name = "subscription-id" use = "required" type =  
"xs:token"/>
```

```

    <attribute name = "fullupdate" default = "true" type = "xs:boolean"/>
    <attribute name = "confirmation" default = "false" type =
"xs:boolean"/>
    <attribute name = "new-state" default = "ICE-ANY"
    type ="icesdt:package-sequence-stateType"/>
<attribute name = "old-state" default = "ICE-ANY" type =
"icesdt:package-sequence-stateType"/>
    <anyAttribute namespace = "##other" processContents = "lax"/>
</complexType>

```

```

<!--
|
|  groupType
|
-->

```

```

<complexType name = "groupType">
  <sequence>
    <any namespace = "##other" processContents = "lax"
    minOccurs = "0"/>
    <element name = "metadata" type = "icedel:metadataType"
    minOccurs = "0"/>
    <group ref = "icedel:cm.package"/>
  </sequence>
  <attribute name = "name" type = "xs:token"/>
  <attribute name = "subscription-element-id" type = "xs:string"/>
</complexType>

```

```

<!--
|
|  addType
|  todo: item-link??
|
-->

```

```

<complexType name = "addType">
  <sequence>
    <element name = "metadata" type = "icedel:metadataType"
    minOccurs = "0"/>
    <choice minOccurs = "0">
      <element name = "item" type = "icedel:itemType"/>
      <element name = "item-ref">
        <complexType>
          <sequence>
            <element name = "reference"
            type = "icesdt:urlAccessType"/>
            <any namespace = "##other"
            processContents = "lax" minOccurs = "0"
            maxOccurs = "unbounded"/>
          </sequence>
          <attribute name = "retrieve-after"
            type = "icesdt:dateTime"/>
          <attribute name = "name" type = "xs:token"/>
        </complexType>
      </element>
    </choice>
    <any namespace = "##other" processContents = "lax"/>
  </sequence>
</complexType>

```



```

        </choice>
    </sequence>
    <attribute name = "subscription-element-id" type = "xs:token"/>
    <attribute name = "is-new" type = "xs:boolean"/>
    <attribute name = "activation" type = "icesdt:dateTime"/>
    <attribute name = "expiration" type = "icesdt:dateTime"/>
    <anyAttribute namespace = "##other" processContents = "lax"/>
</complexType>

<!--
    |
    | metadataType
    |
-->

<complexType name = "metadataType" mixed = "true">
    <sequence minOccurs = "0" maxOccurs = "unbounded">
        <element name = "description" type = "icesdt:textType" minOccurs =
"0"/>
        <any namespace = "##local ##other " processContents = "lax"
minOccurs = "0" maxOccurs = "unbounded"/>
    </sequence>
    <attributeGroup ref = "icedel:attlist.item"/>
    <attributeGroup ref = "icesdt:attlist.genericmetadata"/>
    <anyAttribute namespace = "##other" processContents = "lax"/>
</complexType>

<!--
    |
    | itemType
    |
-->

<complexType name = "itemType" mixed = "true">
    <sequence>
        <any namespace = "##other" processContents = "lax"
minOccurs = "0" maxOccurs = "unbounded"/>
    </sequence>
    <attribute name = "content-transfer-encoding"
default = "x-native-xml">
        <simpleType>
            <restriction base = "NMTOKEN">
                <enumeration value = "base64"/>
                <enumeration value = "x-native-xml"/>
            </restriction>
        </simpleType>
    </attribute>
    <attribute name = "name" type = "xs:token"/>
</complexType>

<!--
    |
    | attlist.item
    |
-->

```

```
<!--
  |
  | remove-item
  |
-->

<element name = "remove-item" type = "icedel:removeType"/>

<!--
  |
  | removeType
  |
-->

<complexType name = "removeType">
  <sequence>
    <any namespace = "##other" processContents = "lax" minOccurs =
"0"/>
  </sequence>
  <attribute name = "subscription-element-id" use = "required" type =
"xs:token"/>
  <anyAttribute namespace = "##other" processContents = "strict"/>
</complexType>

<!--
  |
  | package-confirmations
  |
-->

<element name = "package-confirmations" type = "icedel:package-
confirmationsType"/>

<!--
  |
  | package-confirmationsType
  |
-->

<complexType name = "package-confirmationsType">
  <sequence maxOccurs = "unbounded">
    <choice>
      <element name = "confirmation" type = "icedel:confirmationType"/>
      <any namespace = "##other" processContents = "lax"/>
    </choice>
  </sequence>
</complexType>

<!--
  |
  | confirmationType
  |
-->

<complexType name = "confirmationType">
  <sequence>
```

```
<any namespace = "##other" processContents = "lax"
    minOccurs = "0" maxOccurs = "unbounded"/>
</sequence>
<attribute name = "confirmed" use = "required" type = "xs:boolean"/>
<attribute name = "package-id" use = "required" type = "xs:token"/>
<attribute name = "processing-completed" use = "required">
  <simpleType>
    <restriction base = "NMTOKEN">
      <enumeration value = "received"/>
      <enumeration value = "processed"/>
    </restriction>
  </simpleType>
</attribute>
<anyAttribute namespace = "##other" processContents = "lax"/>
</complexType>
</schema>
```

## 5. ICE Subscribe Schema

```

<?xml version = "1.0" encoding = "UTF-8"?>
<xs:schema xmlns = "http://icestandard.org/ICE/V20/subscribe"
  targetNamespace = "http://icestandard.org/ICE/V20/subscribe"
  xmlns:xs = "http://www.w3.org/2001/XMLSchema"
  xmlns:icesdt = "http://icestandard.org/ICE/V20/simplicatedatatypes"
  elementFormDefault = "qualified">
  <xs:import namespace =
"http://icestandard.org/ICE/V20/simplicatedatatypes"
  schemaLocation = "..\xml schemas\ice-simplicatedatatypes.xsd"/>

<!--
  |
  | offer
  |
  -->

<xs:element name = "offer" type = "offerType"/>

<!--
  |
  | offerType
  |
  -->

<xs:complexType name = "offerType">
  <xs:sequence>
    <xs:element name = "content-metadata"
      type = "content-metadataType" minOccurs = "0"/>
    <xs:element name = "offer-metadata" type = "offer-metadataType"
      minOccurs = "0"/>
    <xs:element name = "description" type = "descriptionType"
      minOccurs = "0"/>
    <xs:element name = "delivery-policy"
      type = "delivery-policyType"/>
    <xs:element name = "business-term" type = "business-termType"
      minOccurs = "0" maxOccurs = "unbounded"/>
    <xs:element name = "required-extension" minOccurs = "0"
      maxOccurs = "unbounded">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base = "required-extensionType">
          <xs:attribute name = "extension-type" use = "required"
            type = "xs:anyURI"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:attribute name = "offer-id" use = "required" type = "xs:token"/>
<xs:attribute name = "name" type = "xs:token"/>
<xs:attribute name = "valid-after" type = "xs:dateTime"/>

```

```
<xs:attribute name = "expiration-date" type = "xs:dateTime"/>
<xs:attribute name = "full-ice" default = "false"
  type = "xs:boolean"/>
<xs:anyAttribute namespace = "##other" processContents = "lax"/>
</xs:complexType>
```

```
<!--
  |
  | delivery-policyType
  |
-->
```

```
<xs:complexType name = "delivery-policyType">
  <xs:sequence>
    <xs:element name = "delivery-rule" type = "delivery-ruleType"
      maxOccurs = "unbounded"/>
    <xs:any namespace = "##other" processContents = "lax"
      minOccurs = "0" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:attribute name = "quantity" type = "xs:integer"/>
  <xs:attribute name = "startdate" type = "icesdt:dateTime"/>
  <xs:attribute name = "stopdate" type = "icesdt:dateTime"/>
  <xs:attribute name = "expiration-priority" default = "first">
    <xs:simpleType>
      <xs:restriction base = "xs:NMTOKEN">
        <xs:enumeration value = "first"/>
        <xs:enumeration value = "time"/>
        <xs:enumeration value = "quantity"/>
        <xs:enumeration value = "last"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:anyAttribute namespace = "##other" processContents = "lax"/>
</xs:complexType>
```

```
<!--
  |
  | delivery-ruleType
  |
-->
```

```
<xs:complexType name = "delivery-ruleType">
  <xs:sequence>
    <xs:element name = "transport" maxOccurs = "unbounded"
      minOccurs = "1" type = "transportType"/>
    <xs:any namespace = "##local ##other" processContents = "lax"
      minOccurs = "0" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:attribute name = "mode" default = "pull">
    <xs:simpleType>
      <xs:restriction base = "xs:NMTOKEN">
        <xs:enumeration value = "pull"/>
        <xs:enumeration value = "push"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
```

```

<xs:attribute name = "monthday" type = "xs:NMTOKENS"/>
<xs:attribute name = "weekday" type = "xs:NMTOKENS"/>
<xs:attribute name = "starttime" type = "icesdt:time"/>
<xs:attribute name = "duration" type = "icesdt:duration"/>
<xs:attribute name = "min-num-updates" type = "xs:integer"/>
<xs:attribute name = "max-num-updates" type = "xs:integer"/>
<xs:attribute name = "incremental-update" type = "xs:boolean"
  default = "false"/>
<xs:attribute name = "confirmation" default = "false"
  type = "xs:boolean"/>
<xs:attribute name = "required" type = "xs:boolean" default =
"true"/>
<xs:anyAttribute namespace = "##other" processContents = "lax"/>
</xs:complexType>

```

```

<!--
  |
  | transportType
  |
-->

```

```

<xs:complexType name = "transportType" mixed = "true">
  <xs:sequence>
    <xs:element name = "delivery-endpoint"
      type = "icesdt:urlAccessType" minOccurs = "0"
      maxOccurs = "1" />
  </xs:sequence>
  <xs:attribute name = "protocol" default = "http:get">
    <xs:simpleType>
      <xs:restriction base = "xs:NMTOKEN">
        <xs:enumeration value = "http:get"/>
        <xs:enumeration value = "ftp"/>
        <xs:enumeration value = "mailto"/>
        <xs:enumeration value = "soap"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name = "packaging-style" default = "ice">
    <xs:simpleType>
      <xs:restriction base = "xs:NMTOKEN">
        <xs:enumeration value = "ice"/>
        <xs:enumeration value = "raw"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

```

```

<!--
  |
  | content-metadataType
  |
-->

```

```

<xs:complexType name = "content-metadataType" mixed = "true">
  <xs:sequence>
    <xs:element name = "text" type = "icesdt:textType"
      minOccurs = "0" maxOccurs = "unbounded"/>
    <xs:any namespace = "##local ##other" processContents = "lax"
      minOccurs = "0" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref = "icesdt:attlist.genericmetadata"/>
</xs:complexType>

<!--
  |
  | offer-metadataType
  |
-->

<xs:complexType name = "offer-metadataType" mixed = "true">
  <xs:sequence>
    <xs:element name = "text" type = "icesdt:textType" minOccurs = "0"
      maxOccurs = "unbounded"/>
    <xs:any namespace = "##local ##other " processContents = "lax"
      minOccurs = "0" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace = "##other" processContents = "lax"/>
</xs:complexType>

<!--
  |
  | descriptionType
  |
-->

<xs:complexType name = "descriptionType" mixed = "true">
  <xs:sequence minOccurs = "0" maxOccurs = "unbounded">
    <xs:element name = "text" type = "icesdt:textType" minOccurs = "0"
      maxOccurs = "unbounded"/>
    <xs:any namespace = "##local ##other " processContents = "lax"
      minOccurs = "0" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace = "##other" processContents = "lax"/>
</xs:complexType>

<!--
  |
  | business-termType
  |
-->

<xs:complexType name = "business-termType" mixed = "true">
  <xs:sequence>
    <xs:element name = "text" type = "icesdt:textType" minOccurs = "0"
      maxOccurs = "unbounded"/>
    <xs:any namespace = "##other" processContents = "lax" minOccurs =
  "0" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:attribute name = "type" use = "required">

```

```

    <xs:simpleType>
      <xs:restriction base = "xs:NMTOKEN">
        <xs:enumeration value = "credit"/>
        <xs:enumeration value = "licensing"/>
        <xs:enumeration value = "payment"/>
        <xs:enumeration value = "reporting"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name = "url" type = "xs:anyURI"/>
  <xs:attribute name = "name" type = "xs:token"/>
  <xs:attribute name = "usage-required" type = "xs:boolean"/>
  <xs:attribute name = "business-term-id" type = "xs:string"/>
  <xs:anyAttribute namespace = "##other" processContents = "lax"/>
</xs:complexType>

<!--
  |
  | required-extensionType
  |
-->

<xs:complexType name = "required-extensionType" mixed = "true">
  <xs:sequence minOccurs = "0" maxOccurs = "unbounded">
    <xs:element name = "text" type = "icesdt:textType" minOccurs = "0"
      maxOccurs = "unbounded"/>
    <xs:any namespace = "##local ##other " processContents = "lax"
      minOccurs = "0" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace = "##other" processContents = "lax"/>
</xs:complexType>

<!--
  |
  | offer-metadataType
  |
-->

<xs:element name = "subscribe">
  <!--<xs:any namespace = "##other" processContents = "lax"
    minOccurs = "0" maxOccurs = "unbounded"/> -->
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "offer" type = "offerType" minOccurs = "0"/>
      <xs:element ref = "icesdt:parameters" minOccurs = "0"/>
    </xs:sequence>
    <xs:attribute name = "subscription-name" type = "xs:token"/>
    <xs:attribute name = "offer-id" type = "xs:token"/>
    <xs:anyAttribute namespace = "##other" processContents = "lax"/>
  </xs:complexType>
</xs:element>

<!--
  |
  | subscription
  |
-->

```



```
-->

<xs:element name = "subscription" type = "subscriptionType"/>

<!--
  |
  | subscriptionType
  |
-->

<xs:complexType name = "subscriptionType">
  <xs:sequence>
    <xs:element name = "offer" type = "offerType"/>
    <xs:any namespace = "##other" processContents = "lax"
      minOccurs = "0" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:attribute name = "subscription-id" use = "required"
    type = "xs:token"/>
  <xs:attribute name = "subscription-name" type = "xs:token"/>
  <xs:attribute name = "current-state" type = "icesdt:package-sequence-
stateType"/>
  <xs:attribute name = "quantity-remaining" type = "xs:integer"/>
  <xs:anyAttribute namespace = "##other" processContents = "lax"/>
</xs:complexType>

<!--
  | subscribe-fault
  | response message returned in the soap fault details
  | if a subscribe/change-subscription request fails
-->

<xs:element name = "subscription-fault">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref = "offer" minOccurs = "1"/>
      <xs:any namespace = "##other" processContents = "lax" minOccurs =
"0" maxOccurs = "unbounded"/>
    </xs:sequence>
    <xs:attribute name = "code" use = "required" type =
"xs:positiveInteger"/>
    <xs:anyAttribute namespace = "##other" processContents = "lax"/>
  </xs:complexType>
</xs:element>

<!--
  |
  | cancel
  |
-->

<xs:element name = "cancel">
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "reason" type = "icesdt:textType"
        minOccurs = "0" maxOccurs = "unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```

        <xs:any namespace = "##other" processContents = "lax"
            minOccurs = "0" maxOccurs = "unbounded"/>
    </xs:sequence>
    <xs:attribute name = "subscription-id" use = "required" type =
"xs:token"/>
    <xs:anyAttribute namespace = "##other" processContents = "lax"/>
</xs:complexType>
</xs:element>

```

```

<!--
    |
    | cancellation
    |
-->

```

```

<xs:element name = "cancellation">
    <xs:complexType>
        <xs:sequence>
            <xs:any namespace = "##other" processContents = "lax"
                minOccurs = "0" maxOccurs = "unbounded"/>
        </xs:sequence>
        <xs:attribute name = "cancellation-id" use = "required"
            type = "xs:token"/>
        <xs:attribute name = "subscription-id" use = "required"
            type = "xs:token"/>
        <xs:anyAttribute namespace = "##other" processContents = "lax"/>
    </xs:complexType>
</xs:element>

```

```

<!--
    |
    | get-status
    |
-->

```

```

<xs:element name = "get-status">
    <xs:complexType>
        <xs:sequence>
            <xs:any namespace = "##other" processContents = "lax"
                minOccurs = "0" maxOccurs = "unbounded"/>
        </xs:sequence>
        <xs:attribute name = "subscription-id" type = "xs:token"/>
        <xs:anyAttribute namespace = "##other" processContents = "lax"/>
    </xs:complexType>
</xs:element>

```

```

<!--
    |
    | status
    |
-->

```

```

<xs:element name = "status" type = "statusType"/>

```

```

<!--
    |

```

```
| statusType  
|  
-->
```

```
<xs:complexType name = "statusType">  
  <xs:sequence>  
    <xs:element name = "subscription" type = "subscriptionType"  
      minOccurs = "0" maxOccurs = "unbounded"/>  
    <xs:any namespace = "##other" processContents = "lax"  
      minOccurs = "0" maxOccurs = "unbounded"/>  
  </xs:sequence>  
</xs:complexType>  
</xs:schema>
```

## 6. Overview of ICE WSDL Samples

In addition to the schema definitions for ICE 2.0, Sample WSDL1.2 scripts are provided as a reference. These scripts are non-normative and are simply an example of WSDL scripts to define operations and bindings Full ICE.

<http://www.icestandard.org/Spec/V20/wsd1/ice-subscriber-full120040801.wsd1>

<http://www.icestandard.org/Spec/v20/wsd1/ice-syndicator-full120040801.wsd1>

## 7. Sample Full ICE Syndicator WSDL Script

A Full ICE Syndicator WSDL script defines all operations, port types, bindings and services from the Syndicator point of view. Each input to the Syndicator and expected output for every operation, as well as the fault is specified in the WSDL script. See Figure 7.1.

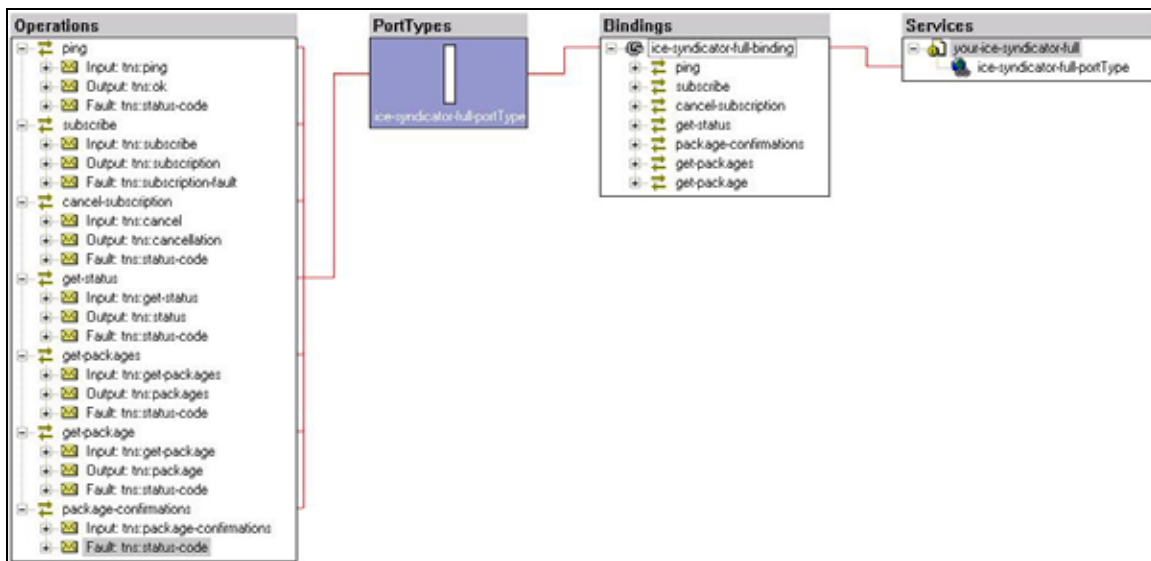


Figure 7.1 Diagram of the Full ICE Syndicator WSDL Script

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="ice-syndicator-full"
targetNamespace="http://icestandard.org/ICE/V20/syndicator/full"
xmlns:tns="http://icestandard.org/ICE/V20/syndicator/full"
xmlns:icemsg="http://icestandard.org/ICE/V20/message"
xmlns:icesub="http://icestandard.org/ICE/V20/subscribe"
xmlns:icedel="http://icestandard.org/ICE/V20/delivery"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/">

  <!--
  | Import schema definitions
  --> <import namespace="http://icestandard.org/ICE/V20/message"
    location="http://www.idealliance.org/ICE/ice-message.xsd"/>

  <import namespace="http://icestandard.org/ICE/V20/delivery"
    location="http://www.idealliance.org/ICE/ice-delivery.xsd"/>

  <import namespace="http://icestandard.org/ICE/V20/subscribe"
    location="http://www.idealliance.org/ICE/ice-subscribe.xsd"/>

  <!-- ping messages -->
```

```
<message name="ping">
  <part name="pingReq" element="icemsg:ping"/>
</message>
<message name="ok">
  <part name="okResp" element="icemsg:OK"/>
</message>

<!-- subscribe messages -->
<message name="subscribe">
  <part name="subscribeReq" element="icesub:offer"/>
</message>
<message name="subscription">
  <part name="subscriptionResp" element="icesub:subscription"/>
</message>

<!-- cancel messages -->
<message name="cancel">
  <part name="cancelReq" element="icesub:cancel"/>
</message>
<message name="cancellation">
  <part name="cancellationResp" element="icesub:cancellation"/>
</message>

<!-- get-status messages -->
<message name="get-status">
  <part name="getStatusReq" element="icesub:get-status"/>
</message>
<message name="status">
  <part name="statusResp" element="icesub:status"/>
</message>

<!-- get-package messages -->
<message name="get-packages">
  <part name="getPackagesReq" element="icedel:get-packages"/>
</message>
<message name="packages">
  <part name="packagesResp" element="icedel:packages"/>
</message>
<message name="get-package">
  <part name="getPackageReq" element="icedel:get-package"/>
</message>
<message name="package">
  <part name="packageResp" element="icedel:package"/>
</message>

<!-- status and fault messages -->
<message name="status-code">
  <part name="statusResp" element="icemsg:status-code"/>
</message>

<message name="subscription-fault">
  <part name="subscriptionFaultResp" element="icesub:subscription-
fault"/>
</message>
```

```
<message name="package-confirmations">
  <part name="packageConfirmationReq" element="icedel:package-
confirmations"/>
</message>

<message name="header">
  <part name="header" element="icemsg:header"/>
</message>

<!-- portType definition -->
<portType name="ice-syndicator-full-portType">

  <operation name="ping">
    <input message="tns:ping" name="ping"/>
    <output message="tns:ok" name="ok"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

  <operation name="subscribe">
    <input message="tns:subscribe" name="subscribe"/>
    <output message="tns:subscription" name="subscription"/>
    <fault message="tns:subscription-fault" name="subscription-
fault"/>
  </operation>

  <operation name="cancel-subscription">
    <input message="tns:cancel" name="cancel"/>
    <output message="tns:cancellation" name="cancellation"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

  <operation name="get-status">
    <input message="tns:get-status" name="get-status"/>
    <output message="tns:status" name="status"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

  <!-- delivery operations -->
  <operation name="get-packages">
    <input message="tns:get-packages" name="get-packages"/>
    <output message="tns:packages" name="packages"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

  <operation name="get-package">
    <input message="tns:get-package" name="get-package"/>
    <output message="tns:package" name="package"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

  <!-- package confirmations operations -->
  <operation name="package-confirmations">
    <input message="tns:package-confirmations" name="package-
confirmations"/>
  </operation>
</portType>
```

```
        <output message="tns:ok"      name="ok" />
        <fault  message="tns:status-code"      name="status-code" />
    </operation>

</portType>

<!--
    SOAP Binding
-->
<binding name="ice-syndicator-full-binding" type="tns:ice-syndicator-
full-portType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />

    <!--
        ping
    -->
    <operation name="ping">
        <soap:operation />
        <input name="ping">
            <soap:body use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full" />
            <soap:header message="header" part="header" use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full" />
        </input>
        <output name="ok">
            <soap:body use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full" />
            <soap:header message="header" part="header" use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full" />
            <soap:headerfault message="fault" use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full" />
        </output>
        <fault name="status-code">
            <soap:fault name="status-code" use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full" />
        </fault>
    </operation>

    <!--
        subscribe
    -->
    <operation name="subscribe">
        <soap:operation />
        <input name="subscribe">
            <soap:body use="literal"

namespace="http://icestandard.org/ICE/V20/syndicator/full" />
            <soap:header message="header" part="header" use="literal"

namespace="http://icestandard.org/ICE/V20/syndicator/full" />
        </input>
        <output name="subscription">
```



```
        <soap:body use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        <soap:header message="header" part="header" use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        <soap:headerfault message="fault" use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        </output>
        <fault name="subscription-fault">
            <soap:fault name="subscription-fault" use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        </fault>
    </operation>

    <!--
        cancel
    -->
    <operation name="cancel-subscription">
        <soap:operation/>
        <input name="cancel">
            <soap:body use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
            <soap:header message="header" part="header" use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        </input>
        <output name="cancellation">
            <soap:body use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
            <soap:header message="header" part="header" use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
            <soap:headerfault message="fault" use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        </output>
        <fault name="status-code">
            <soap:fault name="status-code" use="literal"
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        </fault>
    </operation>

    <!--
        get-status
    -->
    <operation name="get-status">
        <soap:operation/>
        <input name="get-status">
            <soap:body use="literal"
```

```
namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  <soap:header message="header" part="header" use="literal"

namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </input>
  <output name="status">
    <soap:body use="literal"

namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  <soap:header message="header" part="header" use="literal"

namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  <soap:headerfault message="fault" use="literal"

namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </output>
  <fault name="status-code">
    <soap:fault name="status-code" use="literal"

namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </fault>
</operation>

<!--
  package-confirmations
-->
<operation name="package-confirmations">
  <soap:operation/>
  <input name="package-confirmations">
    <soap:body use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
    <soap:header message="header" part="header" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </input>
  <output name="ok">
    <soap:body use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
    <soap:header message="header" part="header" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
    <soap:headerfault message="fault" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </output>
  <fault name="status-code">
    <soap:fault name="status-code" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </fault>
</operation>

<!--
  get-packages
-->
<operation name="get-packages">
  <soap:operation/>
  <input name="get-packages">
```

```
<soap:body use="literal"
  namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
<soap:header message="header" part="header" use="literal"
  namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
</input>
<output name="packages">
  <soap:body use="literal"
    namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  <soap:header message="header" part="header" use="literal"
    namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  <soap:headerfault message="fault" use="literal"
    namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
</output>
<fault name="status-code">
  <soap:fault name="status-code" use="literal"
    namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
</fault>
</operation>
<!--
  get-package
-->
<operation name="get-package">
  <soap:operation/>
  <input name="get-package">
    <soap:body use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
    <soap:header message="header" part="header" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </input>
  <output name="package">
    <soap:body use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
    <soap:header message="header" part="header" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
    <soap:headerfault message="fault" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </output>
  <fault name="status-code">
    <soap:fault name="status-code" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </fault>
</operation>

</binding>

<!-- Sample Service -->
<service name="your-ice-syndicator-full">
  <port name="ice-syndicator-full-portType" binding="tns:ice-
syndicator-full-binding">
    <soap:address location="http://your-ice-server.com/soap-ice"/>
  </port>
</service>
</definitions>
```

## 8. Sample Full ICE Subscriber WSDL Scripts

A Full ICE Subscriber WSDL script defines all operations, port types, bindings and services from the Subscriber point of view. Each input to the Subscriber and expected output for every operation, as well as the fault is specified in the WSDL script. See Figure 8.1.

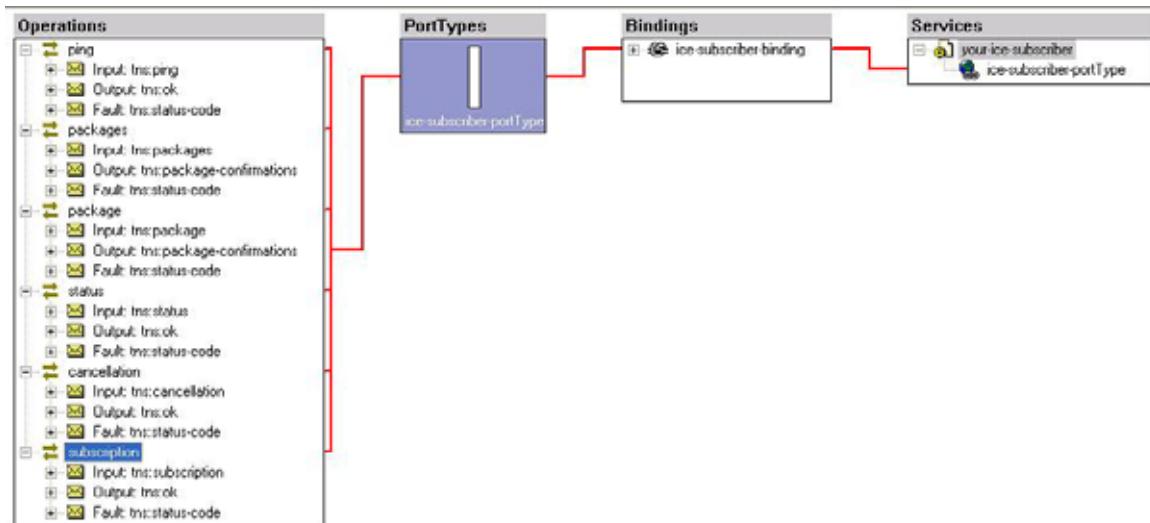


Figure 8.1 Diagram of the Full ICE Subscriber WSDL script.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="ice-subscriber"
  targetNamespace="http://icestandard.org/ICE/V20/wsd1/subscriber"
  xmlns:tns="http://icestandard.org/ICE/V20/wsd1/subscriber"
  xmlns:icemsg="http://icestandard.org/ICE/V20/message"
  xmlns:icedel="http://icestandard.org/ICE/V20/delivery"
  xmlns:icesub="http://icestandard.org/ICE/V20/subscribe"
  xmlns:soap="http://schemas.xmlsoap.org/wsd1/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsd1/">

  <!-- import schema definition -->
  <import namespace="http://icestandard.org/ICE/V20/message"
    location="http://www.idealliance.org/ICE/ice-message.xsd"/>

  <import namespace="http://icestandard.org/ICE/V20/delivery"
    location="http://www.idealliance.org/ICE/ice-delivery.xsd"/>

  <import namespace="http://icestandard.org/ICE/V20/subscribe"
    location="http://www.idealliance.org/ICE/ice-subscribe.xsd"/>

  <!-- messages -->
  <!-- ping messages -->
  <message name="ping">
```

```
    <part name="pingReq" element="icemsg:ping"/>
</message>
<message name="ok">
    <part name="okResp" element="icemsg:OK"/>
</message>

<!-- subscribe messages -->
<message name="subscribe">
    <part name="subscribeReq" element="icesub:offer"/>
</message>

<message name="subscription">
    <part name="subscriptionResp" element="icesub:subscription"/>
</message>

<!-- cancel messages -->
<message name="cancel">
    <part name="cancelReq" element="icesub:cancel"/>
</message>
<message name="cancellation">
    <part name="cancellationResp" element="icesub:cancellation"/>
</message>

<!-- get-status messages -->
<message name="get-status">
    <part name="getStatusReq" element="icesub:get-status"/>
</message>
<message name="status">
    <part name="statusResp" element="icesub:status"/>
</message>

<!-- get-package messages -->
<message name="get-packages">
    <part name="getPackagesReq" element="icedel:get-packages"/>
</message>
<message name="packages">
    <part name="packagesResp" element="icedel:packages"/>
</message>
<message name="get-package">
    <part name="getPackageReq" element="icedel:get-package"/>
</message>
<message name="package">
    <part name="packageResp" element="icedel:package"/>
</message>

<!-- status and fault messages -->
<message name="status-code">
    <part name="statusResp" element="icemsg:status-code"/>
</message>

<message name="subscription-fault">
    <part name="subscriptionFaultResp" element="icesub:subscription-
fault"/>
</message>

<message name="package-confirmations">
```

```
    <part name="packageConfirmationResp" element="icedel:package-
confirmations"/>
  </message>

<!-- ice header -->

  <message name="header">
    <part name="header" element="icemsg:header"/>
  </message>

  <!-- portType definition -->
  <portType name="ice-subscriber-portType">
<!-- Subscriber input/output operations -->
<!-- ping operation -->
  <operation name="ping">
    <input message="tns:ping" name="ping"/>
    <output message="tns:ok" name="ok"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

<!-- package operations -->
  <operation name="packages">
    <input message="tns:packages" name="packages"/>
    <output message="tns:package-confirmations" name="package-
confirmations"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

  <operation name="package">
    <input message="tns:package" name="package"/>
    <output message="tns:package-confirmations" name="package-
confirmations"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

<!-- status operation -->
  <operation name="status">
    <input message="tns:status" name="status"/>
    <output message="tns:ok" name="ok"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

<!-- subscription operations -->

  <operation name="cancellation">
    <input message="tns:cancellation" name="cancellation"/>
    <output message="tns:ok" name="ok"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>

  <operation name="subscription">
    <input message="tns:subscription" name="subscription"/>
    <output message="tns:ok" name="ok"/>
    <fault message="tns:status-code" name="status-code"/>
  </operation>
```

```
</portType>

<!--
    SOAP Binding
-->
<binding name="ice-subscriber-binding" type="tns:ice-subscriber-
portType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>

    <!--
        ping
    -->
    <operation name="ping">
        <soap:operation/>
        <input name="ping">
            <soap:body use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
            <soap:header message="header" part="header" use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        </input>
        <output name="ok">
            <soap:body use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
            <soap:header message="header" part="header" use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
            <soap:headerfault message="fault" use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        </output>
        <fault name="status-code">
            <soap:fault name="status-code" use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        </fault>
    </operation>

    <!--
        package
    -->
    <operation name="package">
        <soap:operation/>
        <input name="package">
            <soap:body use="literal"
namespace="http://icestandard.org/ICE/V20/wsdل/subscriber"/>
            <soap:header message="header" part="header" use="literal"
                namespace="http://icestandard.org/ICE/V20/wsdل/subscriber"/>
        </input>
        <output name="package-confirmations">
            <soap:body use="literal"
                namespace="http://icestandard.org/ICE/V20/wsdل/subscriber"/>
            <soap:header message="header" part="header" use="literal"
                namespace="http://icestandard.org/ICE/V20/wsdل/subscriber"/>
            <soap:headerfault message="fault" use="literal"
                namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
        </output>
        <fault name="status-code">
```

```
        <soap:fault name="status-code " use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
      </fault>
    </operation>

    <!--
      packages
    -->
    <operation name="packages">
      <soap:operation/>
      <input name="packages">
        <soap:body use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
        <soap:header message="header" part="header" use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
      </input>
      <output name="package-confirmations">
        <soap:body use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
        <soap:header message="header" part="header" use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
        <soap:headerfault message="fault" use="literal"
          namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
      </output>
      <fault name="status-code">
        <soap:fault name="status-code " use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
      </fault>
    </operation>

    <!--
      cancel
    -->
    <operation name="cancellation">
      <soap:operation/>
      <input name="cancellation">
        <soap:body use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
        <soap:header message="header" part="header" use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
      </input>
      <output name="ok">
        <soap:body use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
        <soap:header message="header" part="header" use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
        <soap:headerfault message="fault" use="literal"
          namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
      </output>
      <fault name="status-code">
        <soap:fault name="status-code" use="literal"
          namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
      </fault>
    </operation>

    <!--
```



```
        subscription
    -->
<operation name="subscription">
  <soap:operation/>
  <input name="subscription">
    <soap:body use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
    <soap:header message="header" part="header" use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
  </input>
  <output name="ok">
    <soap:body use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
    <soap:header message="header" part="header" use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
    <soap:headerfault message="fault" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </output>
  <fault name="status-code">
    <soap:fault name="status-code" use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
  </fault>
</operation>
<!--
        status
    -->
<operation name="status">
  <soap:operation/>
  <input name="status">
    <soap:body use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
    <soap:header message="header" part="header" use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
  </input>
  <output name="ok">
    <soap:body use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
    <soap:header message="header" part="header" use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
    <soap:headerfault message="fault" use="literal"
      namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  </output>
  <soap:headerfault message="fault" use="literal"
    namespace="http://icestandard.org/ICE/V20/syndicator/full"/>
  <fault name="status-code">
    <soap:fault name="status-code" use="literal"
      namespace="http://icestandard.org/ICE/V20/wsdl/subscriber"/>
  </fault>
</operation>
</binding>
```

```
<!-- Sample Service -->
<service name="your-ice-subscriber">
  <port name="ice-subscriber-portType" binding="tns:ice-subscriber-
binding">
    <soap:address location="http://your-ice-server.com/soap-ice"/>
  </port>
</service>
</definitions>
```