        Lightweight Directory Access Protocol (LDAP) Schema for
  Universal Description, Discovery, and Integration version 3 (UDDIv3)

Status of This Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

Copyright Notice

Abstract

   This document defines the Lightweight Directory Access Protocol
   (LDAPv3) schema for representing Universal Description, Discovery,
   and Integration (UDDI) data types in an LDAP directory. It defines
   the LDAP object class and attribute definitions and containment rules
   to model UDDI entities, defined in the UDDI version 3 information
   model, in an LDAPv3-compliant directory.

Table of Contents

1. Introduction

   This document defines the Lightweight Directory Access Protocol
   [LDAPv3] schema elements to represent the core data structures
   identified in the Universal Description, Discovery, and Integration
   version 3 [UDDIv3] information model. This includes a
   businessEntity, a businessService, a bindingTemplate, a tModel, a
   publisherAssertion, and a Subscription. Portions of [UDDIv3] are
   repeated here for clarity.

2. Conventions Used in This Document

   The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119 [RFC2119].

   All schema definitions are provided using [RFC2252] descriptions, and
   are line-wrapped for readability only.

3. Representation of UDDI Data Structures

   The information that makes up a registration in a UDDI registry
   consists of these data structure types. This division by information
   type provides simple partitions to assist in the rapid location and
   understanding of the different information that makes up a
   registration.

   The individual instance data managed by a UDDI registry is sensitive
   to the parent/child relationships found in the schema. A
   businessEntity object contains one or more unique businessService
   objects. Similarly, individual businessService objects contain
   specific instances of bindingTemplate, which in turn contains
   information that includes pointers to specific instances of tModel
   objects.

   It is important to note that no single instance of a core schema type
   is ever "contained" by more than one parent instance. This means
   that only one specific businessEntity object (identified by its
   unique key value) will ever contain or be used to express information
   about a specific instance of a businessService object (also
   identified by its own unique key value).

3.1. businessEntity

   The businessEntity object represents all known information about a
   business or entity that publishes descriptive information about the
   entity as well as the services that it offers. The businessEntity is
   the top-level container that accommodates holding descriptive

information about a business or entity. Service descriptions and
technical information are expressed within a businessEntity by a
containment relationship.

3.1.1. Representation in the Directory

A businessEntity is represented in the directory by the attributes
uddiBusinessKey, uddiAuthorizedName, uddiOperator, uddiDiscoveryURLs,
uddiName, uddiDescription, uddiIdentifierBag, uddiCategoryBag, and
uddiv3DigitalSignature, along with corresponding v3 keys viz.
uddiv3BusinessKey, as defined in Section 4. A businessEntity may
contain zero or more instances of uddiContact and
uddiBusinessService.

A mandatory attribute, uddiBusinessKey, contains the unique
identifier for a given instance of a businessEntity.

businessEntity's definition is given in Section 5.

3.2. businessService

The businessService instances represent a logical business service.
Each businessService object is the logical child of a single
businessEntity object. Each businessService element contains
descriptive information in business terms outlining the type of
technical services found within each businessService instance.

In some cases, businesses would like to share or reuse services,
e.g., when a large enterprise publishes separate businessEntity
structures. This can be established by using the businessService
instance as a projection to an already published businessService.

3.2.1. Representation in the Directory

A businessService is represented in the directory by the attributes
uddiBusinessKey, uddiServiceKey, uddiName, uddiDescription,
uddiCategoryBag, uddiIsProjection, and uddiv3DigitalSignature, along
with corresponding v3 keys viz. uddiv3BusinessKey, and
uddiv3ServiceKey, as defined in Section 4. A businessService may
contain zero or more instances of uddiBindingTemplate.

The mandatory attribute, uddiServiceKey, contains the unique
identifier for a given instance of a businessService.

businessService's definition is given in Section 5.

3.3. bindingTemplate

   Technical descriptions of Web services are accommodated via
   individual contained instances of bindingTemplate objects. These
   instances provide support for determining a technical entry point or
   optionally support remotely hosted services, as well as a lightweight
   facility for describing unique technical characteristics of a given
   implementation. Support for technology and application specific
   parameters and settings files are also supported.

   Since UDDI's main purpose is to enable description and discovery of
   Web service information, it is the bindingTemplate that provides the
   most interesting technical data. With UDDIv3, bindingTemplates also
   can have categorization information.

   Each bindingTemplate instance has a single logical businessService
   parent, which in turn has a single logical businessEntity parent.

3.3.1. Representation in the Directory

   A bindingTemplate is represented in the directory by the attributes
   uddiBindingKey, uddiServiceKey, uddiDescription, uddiAccessPoint,
   uddiHostingRedirector, uddiCategoryBag, and uddiv3DigitalSignature,
   along with corresponding v3 keys viz. uddiv3ServiceKey and
   uddiv3BindingKey, as defined in Section 4. A bindingTemplate may
   contain zero or more instances of uddiTModelInstanceDetails.

   The mandatory attribute, uddiBindingKey, contains the unique
   identifier for a given instance of a bindingTemplate.

   BindingTemplate's definition is given in Section 5.

3.4. tModel

   The tModel object takes the form of keyed metadata (data about data).
   In a general sense, the purpose of a tModel within the UDDI registry
   is to provide a reference system based on abstraction. Thus, the
   kind of data that a tModel represents is pretty nebulous. In other
   words, a tModel registration can define just about anything, but in
   the current revision, two conventions have been applied for using
   tModels: as sources for determining compatibility and as keyed
   namespace references.

   The information that makes up a tModel is quite simple. There are a
   key, a name, an optional description, and a Uniform Resource Locator
   [URL] that points somewhere--presumably somewhere where the curious
   can go to find out more about the actual concept represented by the
   metadata in the tModel itself.

3.4.1. Representation in the Directory

   A tModel is represented in the directory by the attributes
   uddiTModelKey, uddiAuthorizedName, uddiOperator, uddiName,
   uddiDescription, uddiOverviewDescription, uddiOverviewURL,
   uddiIdentifierBag, uddiCategoryBag, uddiIsHidden, and
   uddiv3DigitalSignature, along with the corresponding v3 key viz.
   uddiv3tModelKey, as defined in Section 4. A tModel may also contain
   a uddiHidden to logically delete a tModel.

   A mandatory attribute, uddiTModelKey, contains the unique identifier
   for a given instance of a tModel.

   tModel's definition is given in Section 5.

3.5. publisherAssertion

   Many businesses, such as large enterprises or marketplaces, are not
   effectively represented by a single businessEntity, since their
   description and discovery are likely to be diverse. As a
   consequence, several businessEntity instances can be published,
   representing individual subsidiaries of a large enterprise or
   individual participants of a marketplace. Nevertheless, they still
   represent a more or less coupled community and would like to make
   some of their relationships visible in their UDDI registrations.

3.5.1. Representation in the Directory

   A publisherAssertion is represented in the directory by the
   attributes uddiFromKey, uddiToKey, uddiKeyedReference, and uddiUUID,
   and uddiv3DigitalSignature, as defined in Section 5.

   A mandatory attribute, uddiUUID, contains the unique identifier for a
   given instance of a publisherAssertion.

   publisherAssertion's definition is given in Section 5.

3.6. Operational Information:

   With UDDIv3, the operational information associated with the core
   UDDI data structures is maintained in a separate OperationalInfo
   structure, so that the digital signature specified by the publisher
   remains valid.

   The operationalInfo structure is used to convey the operational
   information for the UDDIv3 core data structures, that is, the
   businessEntity, businessService, bindingTemplate, and tModel

structures. UDDIv3 OperationalInfo consists of 5 elements: created,
Modified, modifiedIncludingChildren, nodeId, and authorizedName.

Depending on the specific UDDIv3 core data structure, the
operationalInformation is represented in the directory as a
combination of implicit LDAP Standard Operational attributes:
createTimestamp and modifyTimestamp, and the following explicit
attributes: uddiAuthorizedName, uddiv3EntityCreationTime,
uddiv3EntityModificationTime, and uddiv3NodeId.

4. Attribute Type Definitions

   The OIDs for the attribute types in this document have been
   registered by the IANA.

4.1. uddiBusinessKey

   This is used in uddiBusinessEntity and uddiBusinessService.

   The uddiBusinessKey is the unique identifier for a given instance of
   a uddiBusinessEntity. The attribute is optional for businessService
   instances contained within a fully expressed parent that already
   contains a businessKey value.

   If the businessService instance is rendered into the Extensible
   Markup Language [XML] and has no containing parent that has within
   its data a businessKey, the value of the businessKey that is the
   parent of the businessService is required to be provided. This
   behavior supports the ability to browse through the parent-child
   relationships given any of the core elements as a starting point.
   The businessKey may differ from the publishing businessEntity's
   businessKey to allow service projections.

       (1.3.6.1.1.10.4.1 NAME 'uddiBusinessKey'
         DESC 'businessEntity unique identifier'
         EQUALITY caseIgnoreMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
         SINGLE-VALUE
)

4.2. uddiAuthorizedName

   The uddiAuthorizedName is the recorded name of the individual who
   published the uddiBusinessEntity or uddiTModel data. This data is
   generated by the controlling operator and should not be supplied
   within save_business operations.

   With UDDIv3, this attribute is part of the "operationalInformation"

metadata associated with core data structures.

```
( 1.3.6.1.1.10.4.2 NAME 'uddiAuthorizedName'
  DESC 'businessEntity publisher name'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
  SINGLE-VALUE
)
```

4.3. uddiOperator

   The uddiOperator is the certified name of the UDDI registry site
   operator that manages the master copy of the uddiBusinessEntity or
   uddiTModel. The controlling operator records this data at the time
   data is saved. This data is generated and should not be supplied
   within save_business or save_tModel operations.

   With UDDIv3, this field is no longer used -- it is replaced by the
   nodeId (uddiv3NodeId) attribute that is part of the
   "operationalInformation" metadata.

```
( 1.3.6.1.1.10.4.3 NAME 'uddiOperator'
  DESC 'registry site operator of businessEntitys master copy'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
)
```

4.4. uddiName

   This is used in uddiBusinessEntity, uddiBusinessService, and
   uddiTModel.

   These are the human-readable names recorded for the
   uddiBusinessEntity, uddiBusinessService, or uddiTModel, adorned with
   a unique xml:lang value to signify the language that they are
   expressed in. Name search is provided via find_business,
   find_service, or find_tModel calls.

   The publishing of several names, e.g., for romanization purposes, is
   supported. In order to signify the language that the names are
   expressed in, they carry unique xml:lang values. Not more than one
   name element may omit specifying its language. Names passed in this
   way will be assigned the default language code of the registering
   party. This default language code is established at the time that
   publishing credentials are established with an individual Operator

Site. If no default language is provisioned at the time a publisher
signs up, the operator can adopt an appropriate default language
code.

With UDDIv3, multiple values with the same language code are
permitted.

```
( 1.3.6.1.1.10.4.4 NAME 'uddiName'
  DESC 'human readable name'
  EQUALITY caseIgnoreMatch
  ORDERING caseIgnoreOrderingMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)
```

The xml:lang value precedes the name value, with the "#" character
used as the separator.

4.5. uddiDescription

The uddiDescription is an optional repeating element of one or more
descriptions. One description is allowed per national language code
supplied. With UDDIv3, there is no restriction on the number of
descriptions or on what xml:lang value that they may have.

```
( 1.3.6.1.1.10.4.5 NAME 'uddiDescription'
  DESC 'short description'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)
```

The xml:lang value precedes the name value, with the "#" character
used as the separator.

4.6. uddiDiscoveryURLs

This is a list of Uniform Resource Locators (URLs) that point to
alternate, file-based service discovery mechanisms. Each recorded
uddiBusinessEntity structure is automatically assigned a URL that
returns the individual uddiBusinessEntity structure. A URL search is
provided via find_business call.

The uddiDiscoveryURLs attribute is used to hold pointers to URL-
addressable discovery documents. The expected retrieval mechanism
for URLs referenced in the data within this structure is via the
Hypertext Transfer Protocol [HTTP] HTTP-GET operation. The expected
return document is not defined. Rather, a framework for establishing
conventions is provided, and two such conventions are defined within

UDDI behaviors. It is hoped that other conventions come about and
use this structure to accommodate alternate means of discovery. With
UDDIv3, a new convention is defined with useType as "homepage".
Further, a UDDIv3 server need not generate/add a discoveryURL itself,
since this can invalidate the digital signature of signed the
Business Entity saved by publishers.

```
( 1.3.6.1.1.10.4.6 NAME 'uddiDiscoveryURLs'
  DESC 'URL to retrieve a businessEntity instance'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)
```

The useType value precedes the URL value, with the "#" character used
as the separator.

4.7. uddiUseType

The uddiUseType is used to describe the type of contact or address in
freeform text. Suggested examples for contact include "technical
questions", "technical contact", "establish account", "sales
contact", etc. Suggested examples for address include
"headquarters", "sales office", "billing department", etc.

```
( 1.3.6.1.1.10.4.7 NAME 'uddiUseType'
  DESC 'name of convention the referenced document follows'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
)
```

4.8. uddiPersonName

The uddiPersonName should list the name of the person or name of the
job role that will be available behind the contact. Examples of
roles include "administrator" or "webmaster".

```
( 1.3.6.1.1.10.4.8 NAME 'uddiPersonName'
  DESC 'name of person or job role available for contact'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
)
```

With UDDIv3, uddiPersonName becomes multi-valued and each name can
have an xml:lang attribute. The xml:lang value precedes the name
value with the "#" character used as the separator.

4.9. uddiPhone

   This is used to hold telephone numbers for the contact. This element
   can be adorned with an optional uddiUseType attribute for descriptive
   purposes. If more than one phone element is saved, uddiUseType
   attributes are required on each.

        (1.3.6.1.1.10.4.9 NAME 'uddiPhone'
          DESC 'telephone number for contact'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)

   The useType precedes the telephone number by a separating '#' (e.g.,
   "Work Number#123 456-7890") .

4.10. uddiEMail

   This is used to hold email addresses for the contact. This element
   can be adorned with an optional uddiUseType attribute for descriptive
   purposes. If more than one email element is saved, uddiUseType
   attributes are required on each.

        (1.3.6.1.1.10.4.10 NAME 'uddiEMail'
          DESC 'e-mail address for contact'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)

   The useType precedes the email address by a separating '#' (e.g.,
   "President of the United States #president@whitehouse.gov").

4.11. uddiSortCode

   The uddiSortCode is used to drive the behavior of external display
   mechanisms that sort addresses. The suggested values for
   uddiSortCode include numeric ordering values (e.g., 1, 2, 3),
   alphabetic character ordering values (e.g., a, b, c), or the first n
   positions of relevant data within the address.

        (1.3.6.1.1.10.4.11 NAME 'uddiSortCode'
          DESC 'specifies an external display mechanism'
          EQUALITY caseIgnoreMatch
          ORDERING caseIgnoreOrderingMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

   With UDDIv3, the sortCode attribute is deprecated because of the
   guarantee of preserving the document Order.

4.12. uddiTModelKey

   The uddiTModelKey is the unique identifier for a given instance of an
   uddiTModel.

   It is also used in a KeyedReference and in Address structures. When
   used with a keyed reference, this is the unique key to identify a
   value set and implies that the keyName keyValue pair in a
   uddiIdentifier or uddiCategory Bag are to be interpreted by the value
   set referenced by the tModelKey.

   When used with Addressline elements, it implies that the keyName
   keyValue pair given by subsequent uddiAddressLine elements are to be
   interpreted by the address structure associated with the tModel that
   is referenced.

        (1.3.6.1.1.10.4.12 NAME 'uddiTModelKey'
          DESC 'tModel unique identifier'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.13. uddiAddressLine

   The uddiAddressLine contains the actual address in freeform text. If
   the address element contains a uddiTModelKey, these uddiAddressLine
   elements are to be adorned, each with an optional keyName keyValue
   attribute pair. Together with the uddiTModelKey, keyName and
   keyValue qualify the uddiAddressLine in order to describe its
   meaning.

   The uddiAddressLine elements contain string data with a line length
   limit of 80 character positions. Each uddiAddressLine element can be
   adorned with two optional descriptive attributes, keyName and
   keyValue. Both attributes must be present in each address line if a
   uddiTModelKey is assigned to the address structure. By doing this,
   the otherwise arbitrary use of address lines becomes structured.
   Together with the address' uddiTModelKey, keyName and keyValue
   virtually build a uddiKeyedReference that represents an address line
   qualifier, given by the referenced uddiTModel.

   When no uddiTModelKey is provided for the address structure, the
   keyName and keyValue attributes can be used without restrictions, for
   example, to provide descriptive information for each uddiAddressLine

by using the keyName attribute. Since both the keyName and the
keyValue attributes are optional, address line order is significant
and will always be returned by the UDDI-compliant registry in the
order originally provided during a call to save_business.

```
( 1.3.6.1.1.10.4.13 NAME 'uddiAddressLine'
  DESC 'address'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)
```

The keyName, keyValue, and addressData of this attribute are
separated by "#" (e.g., "#"<keyName>"#"<keyValue>"#"<addressData>).
The addressData is the only required portion of the attribute.

4.14. uddiIdentifierBag

The uddiIdentifierBag element allows uddiBusinessEntity or uddiTModel
structures to include information about common forms of
identification such as D-U-N-S_ numbers, tax identifiers, etc. This
data can be used to signify the identity of the uddiBusinessEntity or
can be used to signify the identity of the publishing party.
Including data of this sort is optional, but when used greatly
enhances the search behaviors exposed via the find_xx messages
defined in the UDDI Version 2.0 API Specification [UDDIapi]. For a
full description of the structures involved in establishing an
identity, see UDDI Version 2.0 Data Structure Specification -
Appendix A: Using Identifiers [UDDIdsr].

```
( 1.3.6.1.1.10.4.14  NAME 'uddiIdentifierBag'
  DESC 'identification information'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)
```

The tModel, keyName, and keyValue of this attribute are separated by
"#" (e.g., <tModel>"#"<keyName>"#"<keyValue>). The keyValue is the
only required portion of the attribute.

4.15. uddiCategoryBag

The uddiCategoryBag element allows uddiBusinessEntity,
uddiBusinessService, and uddiTModel structures to be categorized
according to any of several available taxonomy-based classification
schemes. Operator Sites automatically provide validated
categorization support for three taxonomies that cover industry codes
(via NAICS), product and service classifications (via UNSPC), and
geography (via ISO 3166). Including data of this sort is optional,

but when used, it greatly enhances the search behaviors exposed by
the find_xx messages defined in the UDDI Version 2.0 API
Specification [UDDIapi]. For a full description of structures
involved in establishing categorization information, see UDDI Version
2.03 Data Structure Specification--Appendix B: Using Categorization
[UDDIdsr].

```
    ( 1.3.6.1.1.10.4.15 NAME 'uddiCategoryBag'
      DESC 'categorization information'
      EQUALITY caseIgnoreMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
    )
```

The tModel, keyName, and keyValue of this attribute are separated by
"#" (e.g., <tModel>"#"<keyName>"#"<keyValue>). The keyValue is the
only required portion of the attribute.

With UDDIv3, uddiBindingTemplates also supports the uddiCategoryBag
element and they can also be categorized according to any of several
available taxonomy-based classification schemes.

4.16. uddiKeyedReference

The uddiKeyedReference is a general-purpose attribute for a name-
value pair, with an additional reference to a tModel.

```
    ( 1.3.6.1.1.10.4.16 NAME 'uddiKeyedReference'
      DESC 'categorization information'
      EQUALITY caseIgnoreMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
    )
```

The tModel, keyName, and keyValue of this attribute are separated by
"#" (e.g., <tModel>"#"<keyName>"#"<keyValue>). The keyValue is the
only required portion of the attribute. With UDDIv3, the tModelKey
also becomes a mandatory part of the attribute.

Also, UDDIv3 defines KeyedReferenceGroups for CategoryBags. A
keyedReferenceGroup contains a tModelKey and a simple list of
KeyedReference structures. The uddiKeyedReference attribute will
support KeyedReferenceGroups by suffixing the tModelKey for
KeyedReferenceGroup to each of the keyedReference values associated
with the group.

For example, to represent a keyedReference group containing a list of
2 keyed references, the attribute will hold the following 2 strings
as its values:

```
        tModelKey1#KeyName1#KeyValue1#KeyedReferenceGroup1_tModelKey
        tModelKey2#KeyName2#KeyValue2#KeyedReferenceGroup1_tModelKey
```

4.17. uddiServiceKey

   This is the unique key for a given uddiBusinessService. When saving
   a new uddiBusinessService structure, pass an empty uddiServiceKey
   value. This signifies that a UUID value is to be generated. To
   update an existing uddiBusinessService structure, pass the UUID value
   that corresponds to the existing service. If a uddiServiceKey is
   received via an inquiry operation, the key values may not be blank.
   When saving a new or updated service projection, pass the
   uddiServiceKey of the referenced uddiBusinessService structure.

   This attribute is optional when the uddiBindingTemplate data is
   contained within a fully expressed parent that already contains a
   uddiServiceKey value. If the uddiBindingTemplate data is rendered
   into XML and has no containing parent that has within its data a
   uddiServiceKey, the value of the uddiServiceKey that is the ultimate
   containing parent of the uddiBindingTemplate is required to be
   provided. This behavior supports the ability to browse through the
   parent-child relationships given any of the core elements as a
   starting point.

        (1.3.6.1.1.10.4.17 NAME 'uddiServiceKey'
          DESC 'businessService unique identifier'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.18. uddiBindingKey

   This is the unique key for a given uddiBindingTemplate. When saving
   a new uddiBindingTemplate structure, pass an empty uddiBindingKey
   value. This signifies that a UUID value is to be generated. To
   update an existing uddiBindingTemplate, pass the UUID value that
   corresponds to the existing uddiBindingTemplate instance. If a
   uddiBindingKey is received via an inquiry operation, the key values
   may not be blank.

        (1.3.6.1.1.10.4.18 NAME 'uddiBindingKey'
          DESC 'bindingTemplate unique identifier'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)
```

4.19.  uddiAccessPoint

   The uddiAccessPoint element is an attribute-qualified pointer to a
   service entry point.  The notion of service at the metadata level
   seen here is fairly abstract and many types of entry points are
   accommodated.  A single attribute is provided named URLType.

   Required attribute-qualified element8: This element is a text field
   that is used to convey the entry point address suitable for calling a
   particular Web service.  This may be a URL, an electronic mail
   address, or even a telephone number.  No assumptions about the type
   of data in this field can be made without first understanding the
   technical requirements associated with the Web service.

      ( 1.3.6.1.1.10.4.19 NAME 'uddiAccessPoint'
        DESC 'entry point address to call a web service'
        EQUALITY caseIgnoreMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE
      )

   The URLType value precedes the accessPoint value by a separating '#'.

   With UDDIv3, the "URLType" attribute is replaced by a "UseType"
   attribute.  Using this UseType attribute, the accessPoint attribute
   can model a hostingRedirector or support indirection to indicate that
   the accesspoint is specified within a remotely hosted WSDL document.

   For a UDDIv3 registry that needs to support UDDIv2 clients, the
   attribute must allow the representation of URLType and UseType values
   independently.

   The UDDIv3 spec specifies the following logic for mapping values
   between URLType and UseType: If an entity is saved with the v3
   namespace and a v2 inquiry is made, the URLType will be returned as
   "other".  In the case when a v3 inquiry is made on an entity
   published with the v2 namespace, the v3 useType attribute will be
   returned as "endPoint".

   For implementations that need to explicitly model both forms, the
   recommended format is as follows: v2URLType#v3UseType#Address

4.20.  uddiHostingRedirector

   The uddiHostingRedirector element is used to designate that a
   uddiBindingTemplate entry is a pointer to a different
   uddiBindingTemplate entry.  The value in providing this facility is
   seen when a business or entity wants to expose a service description

(e.g., advertise that it has a service available that suits a
specific purpose) that is actually a service described in a separate
uddiBindingTemplate record.  This might occur when a service is
remotely hosted (hence the name of this element), or when many
service descriptions could benefit from a single service description.

The uddiHostingRedirector element has a single attribute and no
element content.  The attribute is a uddiBindingKey value that is
suitable within the same UDDI registry instance for querying and
obtaining the uddiBindingDetail data that is to be used.

More on the uddiHostingRedirector can be found in the appendices for
the UDDI Version 2.0 API Specification [UDDIapi].

Required element if uddiAccessPoint is not provided: This element is
adorned with a uddiBindingKey attribute, giving the redirected
reference to a different uddiBindingTemplate.  If you query a
uddiBindingTemplate and find a uddiHostingRedirector value, you
should retrieve that uddiBindingTemplate and use it in place of the
one containing the uddiHostingRedirector data.

```
( 1.3.6.1.1.10.4.20 NAME 'uddiHostingRedirector'
  DESC 'designates a pointer to another bindingTemplate'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
)
```

With UDDIv3, the hostingRedirector is a deprecated element, since its
functionality is now covered by the accessPoint. For backward-
compatibility, it can still be used, but it is not recommended.

4.21. uddiInstanceDescription

This is an optional repeating element. This is one or more
language-qualified text descriptions that designate what role a
uddiTModel reference plays in the overall service description.

```
( 1.3.6.1.1.10.4.21 NAME 'uddiInstanceDescription'
  DESC 'instance details description'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)
```

The xml:lang value precedes the name value, with the "#" character
used as the separator.

4.22. uddiInstanceParms

   The uddiInstanceParms is an optional element of the uddiInstance. It
   is used to contain settings parameters or a URL reference to a file
   that contains settings or parameters required to use a specific facet
   of a uddiBindingTemplate description. If used to house the
   parameters themselves, the suggested content is a namespace-qualified
   XML string using a namespace outside of the UDDI schema. If used to
   house a URL pointer to a file, the suggested format is a URL that is
   suitable for retrieving the settings or parameters via HTTP-GET.

        (1.3.6.1.1.10.4.22 NAME 'uddiInstanceParms'
          DESC 'URL reference to required settings'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.23. uddiOverviewDescription

   This is an optional repeating element. This language-qualified
   string is intended to hold a short descriptive overview of how a
   particular uddiTModel is to be used.

        (1.3.6.1.1.10.4.23 NAME 'uddiOverviewDescription'
          DESC 'outlines tModel usage'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)

   The xml:lang value precedes the name value, with the "#" character
   used as the separator.

4.24. uddiOverviewURL

   This is an optional element. This string data element is to be used
   to hold a URL reference to a long form of an overview document that
   covers the way a particular uddiTModel specific reference is used as
   a component of an overall Web service description. The recommended
   format for the overviewURL is a URI that is suitable for retrieving
   the actual overview document with an HTTP-GET operation, for example,
   via a Web browser.

       (1.3.6.1.1.10.4.24 NAME 'uddiOverviewURL'
         DESC 'URL reference to overview document'
         EQUALITY caseIgnoreMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
         SINGLE-VALUE
)

   With UDDIv3, uddiOverviewURL becomes multi-valued to allow the
   representation of multiple OverviewDocs within a single
   InstanceDetail element.

   Modeling multiple OverviewDocs within an InstanceDetail element:

   In UDDIv3, the InstanceDetails element in TmodelInstanceInfo can have
   multiple OverviewDoc's. In UDDIv2, we could have only 1 OverviewDoc.
   To retain the grouping between a set of overviewDescriptions and
   overviewURL, we can make both OverviewDoc and OverviewURL multi-
   valued, and have a "group ID" Prefix to each value (to group
   OverviewDescriptions and OverviewURL).

   An example is shown below:

       Overview Description   OverviewURL
       1#xml:lang#overviewDescription1   1#UseType#overviewURL
       1#xml:lang#overviewDescription2   2#UseType#overviewURL
       1#xml:lang#overviewDescription3   4#UseType#overviewURL
       3#xml:lang#overviewDescription1
       3#xml:lang#overviewDescription2
       4#xml:lang#overviewDescription1

   This implies that OverviewDoc1 has 3 overview descriptions and an
   overviewURL. OverviewDoc2 has only an overviewURL. OverviewDoc3 has
   only 2 overviewDescriptions. OverviewDoc4 also has 1 overview
   description and an overviewURL.

4.25. uddiFromKey

   The uddiFromKey is a required element. This is the unique key
   reference to the first uddiBusinessEntity for which the assertion is
   made.

        (1.3.6.1.1.10.4.25 NAME 'uddiFromKey'
          DESC 'unique businessEntity key reference'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.26. uddiToKey

   The uddiToKey is a required element. This is the unique key
   reference to the second uddiBusinessEntity for which the assertion is
   made.

        (1.3.6.1.1.10.4.26 NAME 'uddiToKey'
          DESC 'unique businessEntity key reference'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.27. uddiUUID

   The uddiUUID is a required element. This is to ensure unique
   identification of uddiContact, uddiAddress, and
   uddiPublisherAssertion objects.

        (1.3.6.1.1.10.4.27 NAME 'uddiUUID'
          DESC 'unique attribute'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

   With UDDIv3, this attribute will also be used for unique
   identification of Subscription-feature-related entities.

4.28. uddiIsHidden

   This is used to provide functionality for the delete_tModel
   operation. Logical deletion hides the deleted tModels from
   find_tModel result sets but does not physically delete it.

       (1.3.6.1.1.10.4.28 NAME 'uddiIsHidden'
         DESC 'isHidden attribute'
         EQUALITY booleanMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
         SINGLE-VALUE
)

   In case of UDDIv3, this attribute will represent the "deleted"
   attribute value.

4.29. uddiIsProjection

   This is used to identify a Business Service that has a Service
   Projection.

       (1.3.6.1.1.10.4.29 NAME 'uddiIsProjection'
         DESC 'isServiceProjection attribute'
         EQUALITY booleanMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
         SINGLE-VALUE
)

4.30. uddiLang

   This is used to model the xml:lang value for the Address structure in
   UDDIv3.

       (1.3.6.1.1.10.4.30 NAME 'uddiLang'
         DESC 'xml:lang value in v3 Address structure'
         EQUALITY caseIgnoreMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
         SINGLE-VALUE
)

   The following are attribute definitions to model new elements/fields
   in UDDIv3 information model. These attribute definitions have the
   "uddiv3" prefix to indicate that these attributes represent UDDI
   information model elements unique to UDDIv3.

4.31. uddiv3BusinessKey

   This is the unique UDDIv3 identifier for a given instance of
   uddiBusinessEntity. It is used in uddiBusinessEntity and
   uddiBusinessService.

   A uddiBusinessEntity will include the uddiBusinessKey (the v2 form)
   for unique identification by UDDIv2 clients. The uddiBusinessKey
   (36-char) will also be the LDAP naming attribute for the
   uddiBusinessEntity. The uddiBusinessEntity entry MAY also include
   the uddiv3BusinessKey, the explicit v3 form key, which can be 255
   characters long.

        (1.3.6.1.1.10.4.31 NAME 'uddiv3BusinessKey'
          DESC 'UDDIv3 businessEntity unique identifier'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.32. uddiv3ServiceKey

   This is the unique UDDIv3 identifier for a given instance of
   uddiBusinessService. It is used in uddiBusinessService and
   uddiBindingTemplate.

   A uddiBusinessService will include the uddiServiceKey (the v2 form)
   for unique identification by UDDIv2 clients. The uddiServiceKey
   (36-char) will also be the LDAP naming attribute for the
   uddiBusinessService entry. The uddiBusinessService entry MAY also
   include the uddiv3ServiceKey, the explicit v3 form key, which can be
   255 characters long.

        (1.3.6.1.1.10.4.32 NAME 'uddiv3ServiceKey'
          DESC 'UDDIv3 businessService unique identifier'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.33. uddiv3BindingKey

   This is the unique UDDIv3 identifier for a given instance of
   uddiBindingTemplate.

   A uddiBindingTemplate will include the uddiBindingKey (the v2 form)
   for unique identification by UDDIv2 clients. The uddiBindingKey
   (36-char) will also be the LDAP naming attribute for the

uddiBindingTemplate entry. The uddiBindingTemplate entry MAY also
include the uddiv3BindingKey, the explicit v3 form key, which can be
255 characters long.

```
( 1.3.6.1.1.10.4.33 NAME 'uddiv3BindingKey'
  DESC 'UDDIv3 BindingTemplate unique identifier'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
)
```

4.34. uddiv3TModelKey

This is the unique UDDIv3 identifier for a given instance of a
uddiTModel.

A uddiTModel will include the uddiTModelKey (the v2 form) for unique
identification by UDDIv2 clients. The uddiTModelKey (41-char) will
also be the LDAP naming attribute for the uddiTModel entry. The
uddiTModel entry MAY also include the uddiv3TModelKey, the explicit
v3 form key, which can be 255 characters long.

```
( 1.3.6.1.1.10.4.34 NAME 'uddiv3TModelKey'
  DESC 'UDDIv3 TModel unique identifier'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
)
```

The tModelKey is also used in a KeyedReference and in Address
structures. In all instances where a tModelKey is used as a
reference to tModel, the v3 form of the tModel key (viz.
uddiv3TModelKey) will be the form used, since using the v2 form key
will require translating it to the v3 key by the UDDI Server, which
may invalidate the digital signature of the entity.

4.35. uddiv3DigitalSignature

The UDDIv3 v3 schema supports the signing of the following UDDI
elements using "XML-Signature Syntax and Processing" (see
http://www.w3.org/TR/xmldsig-core/).

```
..businessEntity
..businessService
..bindingTemplate
..tModel
..publisherAssertion
```

This uddiv3DigitalSignature attribute holds the digital signature for
the corresponding UDDI entity.

```
( 1.3.6.1.1.10.4.35 NAME 'uddiv3DigitalSignature'
  DESC 'UDDIv3 entity digital signature'
  EQUALITY caseExactMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  )
```

A Signature element SHOULD be generated according to the required
steps of "Core Generation" in XML-Signature Syntax and Processing.
The signature should be calculated on the top-level element that will
be stored by the registry as a result of the Publication API call.
This element, referred to as the data object in the XML-Signature and
Syntax specification, is the businessEntity element for save_business
API calls, the businessService element for save_service API calls,
the bindingTemplate for save_binding API calls, the tModel for
save_tModel API calls, and the publisherAssertion for
set_publisherAssertions and add_publisherAssertion API calls.

The signature should be generated on the elements before they are
added to the body of an API call. Also, according to the signature
generation, all children of the element being signed are included in
the generation of the signature unless first excluded by application
of a transform. Due to the containment of service projections as
businessService elements within a businessEntity element, this also
means that changes to the projected service will render a signature
of the businessEntity containing the projection invalid, unless a
businessService element representing a service projection is excluded
using a transform.

Due to the location of the sequence of Signature elements within an
element that is to be signed, the signature is "enveloped". As a
result of the enveloping of the signature, it is necessary to apply
at least one transformation on the signed entity to exclude the
signature or signature(s). The transformation selected by a
publisher or the XML-Signature tool is specified in a Transform
element inside the Signature element.

4.36. uddiv3NodeId

   This attribute contains the Node Identity for a UDDIv3 node.

      (1.3.6.1.1.10.4.36 NAME 'uddiv3NodeId'
        DESC 'UDDIv3 Node Identifier'
        EQUALITY caseIgnoreMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE
)

4.37. uddiv3EntityModificationTime

   This attribute is used to maintain the last modification time for a
   UDDI entity. It is needed in the context of maintaining the
   modifiedIncludingChildren element. When a child entity (e.g.,
   uddiBindingTemplate) is updated, the parent entity (e.g.,
   uddiBusinessService) LDAP timestamp also gets updated. The
   uddiv3EntityModificationTime attribute saves the last modification
   time of the parent entity (uddiBusinessService in this case).

      (1.3.6.1.1.10.4.37 NAME 'uddiv3EntityModificationTime'
        DESC 'UDDIv3 Last Modified Time for Entity'
        EQUALITY generalizedTimeMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
        SINGLE-VALUE
)

   The following attribute definitions define attributes related to the
   modeling of UDDIv3 subscription-related entities in the LDAP
   directory.

   Subscription provides clients, known as subscribers, with the ability
   to register their interest in receiving information concerning
   changes made in a UDDI registry. These changes can be scoped based
   on preferences provided with the request. The uddiv3Subscription
   object class is used to model registered UDDIv3 subscriptions.

4.38. uddiv3SubscriptionKey

   This is the unique UDDIv3 identifier for a given instance of a
   uddiv3Subscription entity.

        (1.3.6.1.1.10.4.38 NAME 'uddiv3SubscriptionKey'
          DESC 'UDDIv3 Subscription unique identifier'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.39. uddiv3SubscriptionFilter

   This attribute contains the UDDIv3 Subscription Filter, specified as
   part of the save_subscription API, i.e., the Inquiry API specified as
   filtering criteria with a registered subscription. The filtering
   criteria limits the scope of a subscription to a subset of registry
   records. The get_xx and find_xx APIs are all valid choices for use
   as a subscriptionFilter. Only one of these can be chosen for each
   subscription.

        (1.3.6.1.1.10.4.39 NAME 'uddiv3SubscriptionFilter'
          DESC 'UDDIv3 Subscription Filter'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.40. uddiv3NotificationInterval

   This attribute contains the Notification Interval string. It is of
   the type xsd:duration and specifies how often Asynchronous change
   notifications are to be provided to a subscriber.

        (1.3.6.1.1.10.4.40 NAME 'uddiv3NotificationInterval'
          DESC 'UDDIv3 Notification Interval'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.41. uddiv3MaxEntities

   This attribute contains the maximum number of entities to be returned
   as part of a subscription notification. It is an integer and
   specifies the maximum number of entities in a notification returned
   to a subscription listener.

        (1.3.6.1.1.10.4.41 NAME 'uddiv3MaxEntities'
          DESC 'UDDIv3 Subscription maxEntities field'
          EQUALITY integerMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
          SINGLE-VALUE
)

4.42. uddiv3ExpiresAfter

   This attribute specifies the Expiry Time associated with a
   subscription. It is of the XML Schema type xsd:dateTime.

        (1.3.6.1.1.10.4.42 NAME 'uddiv3ExpiresAfter'
          DESC 'UDDIv3 Subscription ExpiresAfter field'
          EQUALITY generalizedTimeMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
          SINGLE-VALUE
)

4.43. uddiv3BriefResponse

   This attribute is a Boolean flag for Brief Response associated with a
   subscription entity. It controls the level of detail returned to a
   subscription listener. The default is "false" when omitted. When
   set to "true", it indicates that the subscription results are to be
   returned to the subscriber in the form of a keyBag, listing all of
   the entities that matched the subscriptionFilter.

        (1.3.6.1.1.10.4.43 NAME 'uddiv3BriefResponse'
          DESC 'UDDIv3 Subscription ExpiresAfter field'
          EQUALITY booleanMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
          SINGLE-VALUE
)

4.44. uddiv3EntityKey

   This is the unique UDDIv3 identifier for a given instance of a core
   UDDI data structure that is to be logged as an Obituary entry
   uddiv3EntityObituary. When a core UDDIv3 Entity is deleted and there
   is an active subscription registered against this UDDI Entity, an
   Obituary entry is created, in which the v3 key of the deleted entry
   is logged as part of the uddiv3EntityKey attribute.

        (1.3.6.1.1.10.4.44 NAME 'uddiv3EntityKey'
          DESC 'UDDIv3 Entity unique identifier'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
          SINGLE-VALUE
)

4.45. uddiv3EntityCreationTime

   This attribute is used to log the original Creation Time for a UDDI
   Entity that is deleted in the uddiv3EntityObituary entry.

   It is also used in uddiBusinessService and uddiBindingTemplate. A
   Move BS operation needs to delete and recreate BT sub-tree due to
   lack of support for moving a sub-tree in many LDAPv3 servers. This
   attribute is used to save the original creation time of the BT during
   a Move BS.

        (1.3.6.1.1.10.4.45 NAME 'uddiv3EntityCreationTime'
          DESC 'UDDIv3 Entity Creation Time'
          EQUALITY generalizedTimeMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
          SINGLE-VALUE
)

4.46. uddiv3EntityDeletionTime

   This attribute is used to log the entity deletion time for a UDDI
   Entity that is deleted in the uddiv3EntityObituary entry.

        (1.3.6.1.1.10.4.46 NAME 'uddiv3EntityDeletionTime'
          DESC 'UDDIv3 Entity Deletion Time'
          EQUALITY generalizedTimeMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
          SINGLE-VALUE
)

5. Object Class Definitions

   The OIDs for the object classes in this document have been registered
   by the IANA.

5.1. uddiBusinessEntity

   This structural object class represents a businessEntity.

        (1.3.6.1.1.10.6.1 NAME 'uddiBusinessEntity'
          SUP top
          STRUCTURAL
          MUST (uddiBusinessKey $
                 uddiName)
          MAY (uddiAuthorizedName $
                uddiOperator $
                uddiDiscoveryURLs $
                uddiDescription $
                uddiIdentifierBag $
                uddiCategoryBag $
                uddiv3BusinessKey $
                uddiv3DigitalSignature $
                uddiv3EntityModificationTime $
                uddiv3NodeId)
)

5.2. uddiContact

   This structural object class represents a contact. It is contained
   by a uddiBusinessEntity.

        (1.3.6.1.1.10.6.2 NAME 'uddiContact'
          SUP top
          STRUCTURAL
          MUST (uddiPersonName $
                 uddiUUID)
          MAY (uddiUseType $
                uddiDescription $
                uddiPhone $
                uddiEMail)
)

5.3. uddiAddress

   This structural object class represents an address. It is contained
   by a uddiContact.

       (1.3.6.1.1.10.6.3 NAME 'uddiAddress'
         SUP top
         STRUCTURAL
         MUST (uddiUUID)
         MAY (uddiUseType $
               uddiSortCode $
               uddiTModelKey $
               uddiv3TmodelKey $
               uddiAddressLine $
               uddiLang)
)

5.4. uddiBusinessService

   This structural object class represents a businessService.

       (1.3.6.1.1.10.6.4 NAME 'uddiBusinessService'
         SUP top
         STRUCTURAL
         MUST (uddiServiceKey)
         MAY (uddiName $
            uddiBusinessKey $
               uddiDescription $
               uddiCategoryBag $
               uddiIsProjection $
               uddiv3ServiceKey $
               uddiv3BusinessKey $
               uddiv3DigitalSignature $
               uddiv3EntityCreationTime $
               uddiv3EntityModificationTime $
               uddiv3NodeId)
)

5.5. uddiBindingTemplate

   This structural object class represents a bindingTemplate.

      (1.3.6.1.1.10.6.5 NAME 'uddiBindingTemplate'
        SUP top
        STRUCTURAL
        MUST (uddiBindingKey)
        MAY (uddiServiceKey $
              uddiDescription $
              uddiAccessPoint $
              uddiHostingRedirector $
              uddiCategoryBag $
              uddiv3BindingKey $
              uddiv3ServiceKey $
              uddiv3DigitalSignature $
              uddiv3EntityCreationTime $
              uddiv3NodeId)
)

5.6. uddiTModelInstanceInfo

   This structural object class represents a tModelInstanceInfo. It is
   contained by a uddiBindingTemplate.

      (1.3.6.1.1.10.6.6 NAME 'uddiTModelInstanceInfo'
        SUP top
        STRUCTURAL
        MUST (uddiTModelKey)
        MAY (uddiDescription $
              uddiInstanceDescription $
              uddiInstanceParms $
              uddiOverviewDescription $
              uddiOverviewURL $
              uddiv3TmodelKey)
)

5.7. uddiTModel

   This structural object class represents a tModel.

       (1.3.6.1.1.10.6.7 NAME 'uddiTModel'
         SUP top
         STRUCTURAL
         MUST (uddiTModelKey $
                 uddiName)
         MAY (uddiAuthorizedName $
               uddiOperator $
               uddiDescription $
               uddiOverviewDescription $
               uddiOverviewURL $
               uddiIdentifierBag $
               uddiCategoryBag $
               uddiIsHidden
               uddiv3TModelKey $
               uddiv3DigitalSignature $
               uddiv3NodeId)
)

5.8. uddiPublisherAssertion

   This structural object class represents a publisherAssertion.

       (1.3.6.1.1.10.6.8 NAME 'uddiPublisherAssertion'
         SUP top
         STRUCTURAL
         MUST (uddiFromKey $
                 uddiToKey $
                 uddiKeyedReference $
                 uddiUUID)
         MAY (uddiv3DigitalSignature $
               uddiv3NodeId)
)

   The following are object class definitions to model new data
   structures needed to implement the UDDIv3 information model. These
   object class definitions have the "uddiv3" prefix to indicate that
   these attributes represent UDDI information model elements unique to
   UDDIv3.

5.9. uddiv3Subscription

   This structural object class represents a Subscription entity.

        (1.3.6.1.1.10.6.9 NAME 'uddiv3Subscription'
          SUP top
          STRUCTURAL
          MUST (uddiv3SubscriptionFilter $
                 uddiUUID)
          MAY (uddiAuthorizedName $
                 uddiv3SubscriptionKey $
                 uddiv3BindingKey $
                 uddiv3NotificationInterval $
                 uddiv3MaxEntities $
                 uddiv3ExpiresAfter $
                 uddiv3BriefResponse $
                 uddiv3NodeId)
)

5.10. uddiv3EntityObituary

   This structural object class represents an Obituary entry for and
   stores obituary information for deleted UDDIv3 entities needed for
   handling subscriptions.

        (1.3.6.1.1.10.6.10 NAME 'uddiv3EntityObituary'
          SUP top
          STRUCTURAL
          MUST (uddiv3EntityKey $
                 uddiUUID)
          MAY (uddiAuthorizedName $
                 uddiv3EntityCreationTime $
                 uddiv3EntityDeletionTime $
                 uddiv3NodeId)
)

6. Name Forms

   This section defines the required hierarchical structure rules and
   naming attributes for the object classes defined in Section 6.

   The OIDs for the structure rules in this document have been
   registered by the IANA.

6.1.   uddiBusinessEntityNameForm

   This name form defines the naming attribute for a businessEntity.

      (1.3.6.1.1.10.15.1 NAME 'uddiBusinessEntityNameForm'
        OC uddiBusinessEntity
        MUST (uddiBusinessKey)
)

6.2.   uddiContactNameForm

   This name form defines the naming attribute for a contact.

      (1.3.6.1.1.10.15.2 NAME 'uddiContactNameForm'
        OC uddiContact
        MUST (uddiUUID)
)

6.3.   uddiAddressNameForm

   This name form defines the naming attribute for an address.

      (1.3.6.1.1.10.15.3 NAME 'uddiAddressNameForm'
        OC uddiAddress
        MUST (uddiUUID)
)

6.4.   uddiBusinessServiceNameForm

   This name form defines the naming attribute for a businessService.

      (1.3.6.1.1.10.15.4 NAME 'uddiBusinessServiceNameForm'
        OC uddiBusinessService
        MUST (uddiServiceKey)
)

6.5.   uddiBindingTemplateNameForm

   This name form defines the naming attribute for a bindingTemplate.

      (1.3.6.1.1.10.15.5 NAME 'uddiBindingTemplateNameForm'
        OC uddiBindingTemplate
        MUST (uddiBindingKey)
)

6.6.   uddiTModelInstanceInfoNameForm

   This name form defines the naming attribute for a tModelInstanceInfo.

      (1.3.6.1.1.10.15.6 NAME 'uddiTModelInstanceInfoNameForm'
        OC uddiTModelInstanceInfo
        MUST (uddiTModelKey)
)

6.7.   uddiTModelNameForm

   This name form defines the naming attribute for a tModel.

      (1.3.6.1.1.10.15.7 NAME 'uddiTModelNameForm'
        OC uddiTModel
        MUST (uddiTModelKey)
)

6.8.   uddiPublisherAssertionNameForm

   This name form defines the naming attribute for a publisherAssertion.

      (1.3.6.1.1.10.15.8 NAME 'uddiPublisherAssertionNameForm'
        OC uddiPublisherAssertion
        MUST (uddiUUID)
)

6.9.   uddiv3SubscriptionNameForm

   This name form defines the naming attribute for a Subscription.

      (1.3.6.1.1.10.15.9 NAME 'uddiv3SubscriptionNameForm'
        OC uddiv3Subscription
        MUST (uddiUUID)
)

6.10.   uddiv3EntityObituaryNameForm

   This name form defines the naming attribute for an Entity Obituary.

      (1.3.6.1.1.10.15.10 NAME 'uddiv3EntityObituaryNameForm'
        OC uddiv3EntityObituary
        MUST (uddiUUID)
)

7. DIT Structure Rules

   This section defines the required hierarchical structure rules for
   the object classes defined in Section 6.

   Note that rule identifiers defined here show the relationship between
   structure rules. Implementations may use different identifiers but
   must follow the same hierarchical model.

7.1. uddiBusinessEntityStructureRule

```
( 1
  NAME 'uddiBusinessEntityStructureRule'
  FORM uddiBusinessEntityNameForm
)
```

7.2. uddiContactStructureRule

   This structure rule defines the object class containment for a
   contact.

```
( 2
  NAME 'uddiContactStructureRule'
  FORM uddiContactNameForm
  SUP ( 1 )
)
```

7.3. uddiAddressStructureRule

   This structure rule defines the object class containment for an
   address.

```
( 3
  NAME 'uddiAddressStructureRule'
  FORM uddiAddressNameForm
  SUP ( 2 )
)
```

7.4. uddiBusinessServiceStructureRule

   This structure rule defines the object class containment for a
   businessService.

```
( 4
  NAME 'uddiBusinessServiceStructureRule'
  FORM uddiBusinessServiceNameForm
  SUP ( 1 )
)
```

7.5. uddiBindingTemplateStructureRule

   This structure rule defines the object class containment for a
   bindingTemplate.

      (5
        NAME 'uddiBindingTemplateStructureRule'
        FORM uddiBindingTemplateNameForm
        SUP (4)
)

7.6. uddiTModelInstanceInfoStructureRule

   This structure rule defines the object class containment for a
   tModelInstanceInfo.

      (6
        NAME 'uddiTModelInstanceInfoStructureRule'
        FORM uddiTModelInstanceInfoNameForm
        SUP (5)
)

7.7. uddiTModelStructureRule

      (7
        NAME 'uddiTModelStructureRule'
        FORM uddiTModelNameForm
)

7.8. uddiPublisherAssertion

      (8
        NAME 'uddiPublisherAssertionStructureRule'
        FORM uddiPublisherAssertionNameForm
)

7.9. uddiv3SubscriptionStructureRule

      (9
        NAME 'uddiv3SubscriptionStructureRule'
        FORM uddiv3SubscriptionNameForm
)

7.10. uddiv3EntityObituaryStructureRule

```
( 10
  NAME 'uddiv3EntityObituaryStructureRule'
  FORM uddiv3EntityObituaryNameForm
)
```

8. Security Considerations

   Storing UDDI data into the directory enables the data to be examined
   and used outside the environment in which it was originally created.
   The directory entry containing the UDDI data could be read and
   modified within the constraints imposed by the access control
   mechanisms of the directory. With UDDIv3 [UDDIv3], publishers can
   digitally sign UDDI Entities enabling registry clients to validate
   the integrity of entries read from the UDDIv3 registry by verifying
   the digital signature.

   Each UDDI Entity has a uddiAuthorizedName attribute that contains an
   LDAP DN identifying the publisher/owner. The referenced LDAP object
   can provide the public key of the signer to a registry client for
   integrity validation of the UDDI Entity.

   Other general LDAP [LDAPv3] security considerations apply. Some of
   the UDDI attributes such as AccessPoints for services may contain
   sensitive information. Use of strong authentication mechanisms and
   data integrity/confidentiality services [RFC2829][RFC2830] is
   advised.

9. IANA Considerations

   Refer to RFC 3383, "Internet Assigned Numbers Authority (IANA)
   Considerations for the Lightweight Directory Access Protocol (LDAP)"
   [RFC3383].

9.1. Object Identifier Registration

   The IANA has registered an LDAP Object Identifier for use in this
   technical specification, according to the following template:

   Subject: Request for LDAP OID Registration
   Person & email address to contact for further information:
      Bruce Bergeson (bruce.bergeson@novell.com)
   Specification: RFC 4403
   Author/Change Controller: IESG
   Comments:
      The assigned OID (10) will be used as a base for identifying
      a number of UDDI schema elements defined in this document.

9.2. Object Identifier Descriptors

   The IANA has registered the LDAP Descriptors used in this technical
   specification as detailed in the following template:

   Subject: Request for LDAP Descriptor Registration Update
   Descriptor (short name): see table
   Object Identifier: see table
   Person & email address to contact for further information:
      Bruce Bergeson (bruce.bergeson@novell.com)
   Usage: see table
   Specification: RFC 4403
   Author/Change Controller: IESG
   Table:

   The following descriptors have been added:

```
   NAME                          Type    OID
   --------------                ----    ------------
   uddiBusinessKey               A       1.3.6.1.1.10.4.1
   uddiAuthorizedName            A       1.3.6.1.1.10.4.2
   uddiOperator                  A       1.3.6.1.1.10.4.3
   uddiName                      A       1.3.6.1.1.10.4.4
   uddiDescription               A       1.3.6.1.1.10.4.5
   uddiDiscoveryURLs             A       1.3.6.1.1.10.4.6
   uddiUseType                   A       1.3.6.1.1.10.4.7
   uddiPersonName                A       1.3.6.1.1.10.4.8
   uddiPhone                     A       1.3.6.1.1.10.4.9
   uddiEMail                     A       1.3.6.1.1.10.4.10
   uddiSortCode                  A       1.3.6.1.1.10.4.11
   uddiTModelKey                 A       1.3.6.1.1.10.4.12
   uddiAddressLine               A       1.3.6.1.1.10.4.13
```

```
      NAME                           Type     OID
      -------------                  ----     ------------
      uddiIdentifierBag              A        1.3.6.1.1.10.4.14
      uddiCategoryBag                A        1.3.6.1.1.10.4.15
      uddiKeyedReference             A        1.3.6.1.1.10.4.16
      uddiServiceKey                 A        1.3.6.1.1.10.4.17
      uddiBindingKey                 A        1.3.6.1.1.10.4.18
      uddiAccessPoint                A        1.3.6.1.1.10.4.19
      uddiHostingRedirector          A        1.3.6.1.1.10.4.20
      uddiInstanceDescription        A        1.3.6.1.1.10.4.21
      uddiInstanceParms              A        1.3.6.1.1.10.4.22
      uddiOverviewDescription        A        1.3.6.1.1.10.4.23
      uddiOverviewURL                A        1.3.6.1.1.10.4.24
      uddiFromKey                    A        1.3.6.1.1.10.4.25
      uddiToKey                      A        1.3.6.1.1.10.4.26
      uddiUUID                       A        1.3.6.1.1.10.4.27
      uddiIsHidden                   A        1.3.6.1.1.10.4.28
      uddiIsProjection               A        1.3.6.1.1.10.4.29
      uddiLang                       A        1.3.6.1.1.10.4.30
      uddiv3BusinessKey              A        1.3.6.1.1.10.4.31
      uddiv3ServiceKey               A        1.3.6.1.1.10.4.32
      uddiv3BindingKey               A        1.3.6.1.1.10.4.33
      uddiv3TmodelKey                A        1.3.6.1.1.10.4.34
      uddiv3DigitalSignature         A        1.3.6.1.1.10.4.35
      uddiv3NodeId                   A        1.3.6.1.1.10.4.36
      uddiv3EntityModificationTime   A        1.3.6.1.1.10.4.37
      uddiv3SubscriptionKey          A        1.3.6.1.1.10.4.38
      uddiv3SubscriptionFilter       A        1.3.6.1.1.10.4.39
      uddiv3NotificationInterval     A        1.3.6.1.1.10.4.40
      uddiv3MaxEntities              A        1.3.6.1.1.10.4.41
      uddiv3ExpiresAfter             A        1.3.6.1.1.10.4.42
      uddiv3BriefResponse            A        1.3.6.1.1.10.4.43
      uddiv3EntityKey                A        1.3.6.1.1.10.4.44
      uddiv3EntityCreationTime       A        1.3.6.1.1.10.4.45
      uddiv3EntityDeletionTime       A        1.3.6.1.1.10.4.46
      uddiBusinessEntity             O        1.3.6.1.1.10.6.1
      uddiContact                    O        1.3.6.1.1.10.6.2
      uddiAddress                    O        1.3.6.1.1.10.6.3
      uddiBusinessService            O        1.3.6.1.1.10.6.4
      uddiBindingTemplate            O        1.3.6.1.1.10.6.5
      uddiTModelInstanceInfo         O        1.3.6.1.1.10.6.6
      uddiTModel                     O        1.3.6.1.1.10.6.7
      uddiPublisherAssertion         O        1.3.6.1.1.10.6.8
      uddiv3Subscription             O        1.3.6.1.1.10.6.9
      uddiv3EntityObituary           O        1.3.6.1.1.10.6.10
      uddiBusinessEntityNameForm     N        1.3.6.1.1.10.15.1
      uddiContactNameForm            N        1.3.6.1.1.10.15.2
      uddiAddressNameForm            N        1.3.6.1.1.10.15.3
```

```
      NAME                            Type    OID
      --------------                  ----    ------------
      uddiBusinessServiceNameForm     N       1.3.6.1.1.10.15.4
      uddiBindingTemplateNameForm     N       1.3.6.1.1.10.15.5
      uddiTModelInstanceInfoNameForm  N       1.3.6.1.1.10.15.6
      uddiTModelNameForm              N       1.3.6.1.1.10.15.7
      uddiPublisherAssertionNameForm  N       1.3.6.1.1.10.15.8
      uddiv3SubscriptionNameForm      N       1.3.6.1.1.10.15.9
      uddiv3EntityObituaryNameForm    N       1.3.6.1.1.10.15.10
```

   where Type A is Attribute, Type O is ObjectClass, Type N is NameForm

   These assignments have been recorded in the following registry:

   http://www.iana.org/assignments/ldap-parameters

10. Normative References

   [LDAPv3]   Hodges, J. and R. Morgan, "Lightweight Directory Access
              Protocol (v3): Technical Specification", RFC 3377,
              September 2002.

   [RFC2252] Wahl, M., Coulbeck, A., Howes, T., and S. Kille,
              "Lightweight Directory Access Protocol (v3): Attribute
              Syntax Definitions", RFC 2252, December 1997.

   [UDDIdsr] UDDI.ORG, "UDDI version 2.03 Data Structure Reference,"
              http://uddi.org/pubs/DataStructure-V2.03-Published-
              20020719.htm

   [UDDIapi] "UDDI Version 2.04 API Specification",
              http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-
              20020719.htm

   [UDDIv3]   UDDI Version 3.0, Published Specification, 19 July 2002
              http://uddi.org/pubs/uddi-v3.00-published-20020719.htm

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC2829] Wahl, M., Alvestrand, H., Hodges, J., and R. Morgan,
              "Authentication Methods for LDAP", RFC 2829, May 2000.

   [RFC2830] Hodges, J., Morgan, R., and M. Wahl, "Lightweight Directory
              Access Protocol (v3): Extension for Transport Layer
              Security", RFC 2830, May 2000.

    [RFC3383] Zeilenga, K., "Internet Assigned Numbers Authority (IANA)
              Considerations for the Lightweight Directory Access
              Protocol (LDAP)", BCP 64, RFC 3383, September 2002.

    [XML]   Extensible Markup Language (XML) 1.0 (Second Edition) W3C
              Recommendation 6 October 2000 http://www.w3.org/TR/REC-xml

    [URL]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
              Resource Identifier (URI): Generic Syntax", STD 66, RFC
              3986, January 2005.

    [HTTP]   Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,
              Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext
              Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

Authors' Addresses

    Bruce Bergeson
    Novell, Inc.
    1800 S Novell Place
    Provo, UT 84606

    Phone: +1 801 861 3854
    EMail: bruce.bergeson@novell.com

    Kent Boogert
    Novell, Inc.
    1800 S Novell Place
    Provo, UT 84606

    Phone: +1 801 861 3212
    EMail: kent.boogert@novell.com

    Vijay Nanjundaswamy
    Oracle India Pvt. Ltd.
    Lexington Towers, Prestige St. John's Woods
    #18, 2nd Cross Road,
    Chikka Audugodi,
    Bangalore 560029
    India

    Phone: +11 9180 4108 5000
    EMail: vijay.nanjundaswamy@oracle.com