



UDDI Spec TC

1 Versioning Value Sets in a UDDI 2 Registry, Version 1.12

3 Technical Note

4 Document identifier:

5 uddi-spec-tc-tn-versioning-value-sets-v112-20030829

6 Current version:

7 [http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-versioning-value-](http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-versioning-value-sets-v112-20030829.htm)
8 [sets-v112-20030829.htm](http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-versioning-value-sets-v112-20030829.htm)

9 Latest version:

10 [http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-versioning-value-](http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-versioning-value-sets.htm)
11 [sets.htm](http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-versioning-value-sets.htm)

12 Editor:

13 Claus von Riegen, SAP

14 Contributors:

15 Tom Bellwood, IBM

16 Abstract:

17 Through the use of value sets in UDDI registries, businesses are able to find each other
18 and the services that meet their needs. However, value set publishers often change their
19 value sets by adding or deleting values and/or changing their meaning in order to meet
20 the needs of a certain domain. This document provides guidelines for providers of value
21 sets on how to register different versions of value sets for use in UDDI versions 2 and 3.

22 Status:

23 This document is updated periodically on no particular schedule. Send comments to the
24 editor.

25 Committee members should send comments on this technical note to the [uddi-](mailto:uddi-spec@lists.oasis-open.org)
26 spec@lists.oasis-open.org list. Others should subscribe to and send comments to the
27 uddi-spec-comment@lists.oasis-open.org list. To subscribe, send an email message to
28 uddi-spec-comment-request@lists.oasis-open.org with the word "subscribe" as the body
29 of the message.

30 For information on whether any patents have been disclosed that may be essential to
31 implementing this technical note, and any offers of patent licensing terms, please refer to
32 the Intellectual Property Rights section of the UDDI Spec TC web page ([http://www.oasis-](http://www.oasis-open.org/committees/uddi-spec/)
33 [open.org/committees/uddi-spec/](http://www.oasis-open.org/committees/uddi-spec/)).

34 **Table of Contents**

35	Introduction	3
36	1.1 Problem statement	3
37	1.2 Terminology	3
38	1.3 Background	3
39	2 Technical note Solution	4
40	2.1 Definitions	4
41	2.1.1 Normative per the UDDI Specification	4
42	2.1.2 Defined in this Technical Note	5
43	2.2 Technical note behavior	6
44	2.3 Discussion	6
45	2.4 update_entities Web service type	6
46	2.4.1 update_entities for UDDI Version 2 entities	7
47	2.4.2 update_entities for UDDI Version 3 entities	9
48	2.5 Examples	10
49	2.5.1 Example of uddi-org:isReplacedBy usage	10
50	2.5.2 Example of update_entities Web service	11
51	2.5.3 Example of update_entities API call	12
52	3 References	14
53	3.1 Normative	14
54	Appendix A. Acknowledgments	15
55	Appendix B. Revision History	16
56	Appendix C. Notices	17
57		

58 Introduction

59 Data has little value if it is lost within a mass of other data and cannot be distinguished or
60 discovered. This applies to UDDI data as well. If a client of UDDI cannot effectively find
61 information within a registry, the purpose of UDDI is considerably compromised. In UDDI there
62 are two mechanisms for providing organization to data: category systems and identifier systems.
63 Generically, these are called “value sets”. The UDDI specification **[UDDIV3]** provides several
64 value sets (for example, the UDDI Types category system `uddi-org:types`), although many others
65 exist and are continually produced (for example, the UDDI Business Registry provides a
66 geographic category system that is based on ISO 3166). Often industry verticals define their own
67 value sets, which are specific to the business they conduct. Many more have yet to be produced.
68 UDDI allows publishers to register these value sets, and associate them with UDDI entities, thus
69 allowing publishers and searchers to distinguish and discover business relevant information with
70 ever greater precision. But these value sets change over time, and if they do so in an
71 uncontrolled way searchers won’t know what to ask for and publishers won’t know how to
72 categorize and identify their UDDI entities.

73 1.1 Problem statement

74 This UDDI Technical Note describes the preferred way to deal with changes in value sets for
75 UDDI version 2 and 3. It outlines a set of practices that allow category systems and identifier
76 systems to change in useful ways without invalidating UDDI entities that use them. The practices
77 are almost identical between UDDI versions 2 and 3.

78 1.2 Terminology

79 The key words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *may*,
80 and *optional* in this document are to be interpreted as described in **[RFC2119]**.

81 1.3 Background

82 This document assumes a working knowledge of the UDDI Specification. In particular, it assumes
83 familiarity with how category systems and identifier systems are represented in UDDI, how they
84 are used to categorize and identify entities in UDDI registries, and how they are used in UDDI find
85 messages.

86 For more information see sections 1.6.5, 3.3.2.10, 3.3.2.11, 3.3.2.12, 3.3.2.13, 3.3.2.14, and
87 11.16 as well as Appendix E and F of **[UDDIV3]**.

88 This technical note takes “Versioning Taxonomy and Identifier Systems”, written by David
89 Ehnebuske and Barbara McKee, <http://uddi.org/pubs/tn-taxonomy-versioning-v1.05-Draft-20010906.pdf> and modifies it to fit in **[UDDIV3]** and follows the OASIS UDDI Specification TC
90 Technical Note format.
91

92 2 Technical note Solution

93 2.1 Definitions

94 2.1.1 Normative per the UDDI Specification

95 **value set**

96 A value set is a category system or an identifier system that is either checked or
97 unchecked.

98

99 **checked value set**

100 A value set whose use is inspected for conformance to the requirements of the value set.

101

102 **unchecked value set**

103 A value set whose use is not inspected for conformance to the requirements of the value
104 set.

105

106 **category system**

107 A category system is a mechanism for categorizing UDDI entities so that they can be
108 described and discovered based on certain categories. UDDI does not prescribe any
109 specific category system.

110

111 **identifier system**

112 An identifier system is a mechanism for Identifying UDDI entities so that they can be
113 described and discovered based on certain identifiers. UDDI does not prescribe any
114 specific identifier system.

115

116 **tModel**

117 A tModel is a keyed entity that provides a reference system based on abstraction. Two
118 primary uses are as sources for determining compatibility of Web services and as a
119 keyed namespace reference.

120

121 **identifierBag**

122 This is an optional container element of a UDDI entity that identifies it according to one or
123 more published identifier systems, for example, Dun & Bradstreet D-U-N-S[®] numbers or
124 tax identifiers. An identifierBag is a list of one or more keyedReference structures, each
125 representing a single identification.

126

127 **categoryBag**

128 This is an optional container element of a UDDI entity that identifies it according to one or
129 more published categorization systems. For example, UNSPSC product and service
130 categorizations codes can be used to describe a product or service offering. A
131 categoryBag is a list of one or more keyedReference structures, each representing a
132 single category. A categoryBag may also contain keyedReferenceGroups

133

134 **keyedReference (identifier and category)**

135 A keyedReference is a mechanism for representing a categorization or an identifier
136 name/value pair from a specific category or identifier system. It consists of three values:
137 tModelKey, KeyName, and KeyValue.

138

139 **keyedReferenceGroup**

140 A keyedReferenceGroup is a container for a list of keyedReference structures that
141 logically belong together.

142

143 **uddi-org:isReplacedBy**

144 It is often desirable to gracefully retire a tModel in favor of a replacement. For example,
145 when a Web service definition or a value set is replaced by a new version, the publisher
146 of the specification may wish to leave the tModel for the existing definition in place so that
147 existing uses will not be disturbed, while at the same time making it clear that there is a
148 replacement available. The UDDI isReplacedBy identifier system, coupled with the
149 behavior of UDDI with respect to deleting tModels, fill this need by allowing the obsolete
150 tModel to point to its replacement. (see section 11.1.6 of [UDDIV3])

151 **2.1.2 Defined in this Technical Note**

152 **valid code**

153 Any value that, per the definition of the given category or identifier system may appear as
154 the value of the keyValue attribute of a keyedReference that refers to the given category or
155 identifier system¹. Such a code is said to be “in” the category or identifier system; codes
156 that are not valid are not “in” the category or identifier system. For example, 15111505 is a
157 valid code and is therefore in the UNSPSC Version 3.1 taxonomy because ECCMA, the
158 author of UNSPSC has so indicated. The code 15111521 is not valid in UNSPSC Version
159 3.1 because ECCMA has chosen not to include it.

160

161 **code space**

162 The code space is the set of all valid codes for a category or identifier system. For
163 example, all of the codes in the UNSPSC Version 3.1 category system taken as a set
164 constitute the code space of the UNSPSC Version 3.1 category system.

165

166 **upward compatibility of code spaces**

167 A code in category system T_2 is upward compatible with a code in category system T_1 if
168 and only if the two codes are identical and if the code’s meaning in T_2 is a superset of its
169 meaning in T_1 . A code in an identifier system I_2 is upward compatible with a code in
170 identifier system I_1 if and only if the two codes are the same and the code’s meaning is the
171 same. For example, UNSPSC Version 6.01 code 55121701 means “Vinyl letters”. This
172 code is upward compatible with UNSPSC Version 6.03 because in that version 55121701
173 means “Vinyl letters or numbers” which is a superset of its meaning in Version 6.01.

174 Value set V_2 is *upward compatible* with value set V_1 if and only if every code in the code
175 space of V_1 has an upward compatible counterpart in the code space of V_2 . For example,
176 the UNSPSC Version 7 category system is not upward compatible with the UNSPSC

¹ Note that whether or not a given value is a valid code is independent of the value set’s characteristics in terms of checking.

177 Version 3.1 category system because the format of all of the codes changed in version 7.
178 In version 3.1 the form of the codes is "#####" (eight decimal digits) in version 7, the
179 format is "##.##.##.##.##" (five decimal digit pairs separated by ".").

180

181 **successor version**

182 The successor version is a version of a value set whose tModel is linked to via a uddi-
183 org:isReplacedBy keyedReference from that of a predecessor version.

184

185 **predecessor version**

186 The predecessor version is a version of a value set whose tModel contains an uddi-
187 org:isReplacedBy keyedReference that links to a successor version.

188 **2.2 Technical note behavior**

189 If a new version of a value set is upward compatible with the previous version, the existing tModel
190 should be modified to indicate that it represents the new version, and then be republished using
191 the same tModelKey.

192 If a new version of a value set is not upward compatible with the previous version, a new tModel
193 should be published to represent the new version of the value set. The existing tModel should
194 then be made the predecessor version by adding a keyedReference to its identifierBag that
195 refers, using the uddi-org:isReplacedBy identifier system described in section 11.1.6 in **[UDDIV3]**,
196 to the tModel of the new version of the value set.

197 **2.3 Discussion**

198 Changes to value sets that are upward compatible are preferable to those that are not because
199 registry entries containing codes from the old version are still correct under the new version
200 without change.

201 When codes in an existing version become obsolete in a new version, it is helpful for them to
202 remain valid in the new version because it lets the new version be treated as upwardly
203 compatible. New uses of obsolete codes may be discouraged through other means, for example
204 by marking them obsolete in value set browsers.

205 When new, upwardly compatible, versions of value sets are published that contain valid but
206 obsolete codes, value set providers should consider offering a Web service that implements the
207 uddi-org:updateEntities Web service type (see section 2.4). This can help users of the old version
208 eliminate the valid but obsolete codes from their UDDI entities.

209 When it is necessary to make a successor version – i.e., one that is not upwardly compatible –
210 publishers should:

- 211 • Continue to support the predecessor, for example with value set browsers, at least until
212 usage drops substantially.
- 213 • Chain the tModel for the predecessor version to the tModel for the successor (using uddi-
214 org:isReplacedBy, as defined in section 11.1.6 in **[UDDIV3]**) so tools can find the most
215 recent version.
- 216 • Consider offering a Web service that implements the uddi-org:updateEntities Web service
217 type (see section 2.4) Such a service accepts UDDI entities and returns a modified
218 version which is compatible with the successor version of the value set.

219 **2.4 update_entities Web service type**

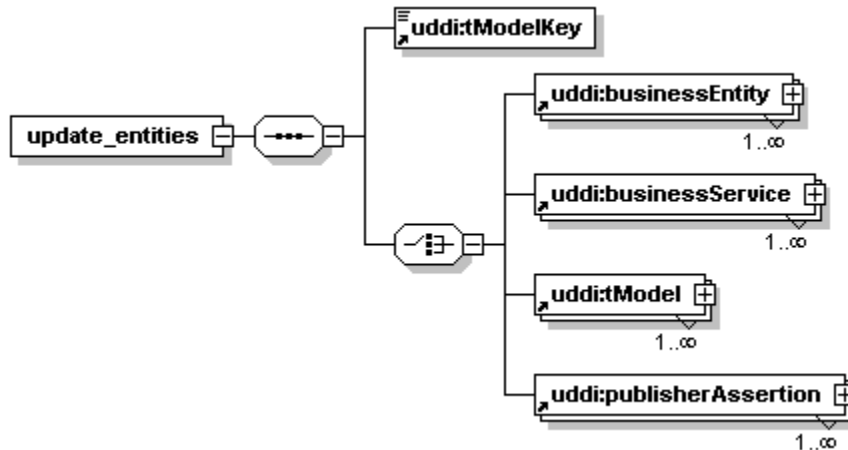
220 The update_entities Web service is used to bring a list of UDDI entities (businessEntity,
221 businessService, bindingTemplate, tModel, or publisherAssertion) up to date with respect to the

222 value set tModel whose tModelKey it is passed. There are two Web service type definitions for
223 the different UDDI versions the passed entities are based upon.

224 2.4.1 update_entities for UDDI Version 2 entities

225 **Syntax:**

226



227

228

229 **Arguments:**

230 • **tModelKey:** This key identifies the value set that is to be updated. keyedReferences in
231 the entities that are also passed referring to this value set or to its predecessor versions
232 are the targets of action for the invocation.

233 The following parameters are mutually exclusive

234 • **businessEntity:** The businessEntity elements whose contained (either directly or
235 indirectly) keyedReferences are to be operated upon.

236 • **businessService:** The businessService elements whose contained keyedReferences
237 are to be operated upon.

238 • **tModel:** The tModel elements whose contained keyedReferences are to be operated
239 upon.

240 • **publisherAssertion:** The publisherAssertion elements whose contained
241 keyedReferences are to be operated upon.

242 **Behavior:**

243 Web services of this type aid in updating entities that are categorized or identified by changed
244 versions of a value set. When value set providers need to create new versions, they may also
245 choose to provide services of this type to help those who use the value sets convert UDDI entities
246 that refer to older versions.

247 A Web service of this type does not change anything in a UDDI registry; it simply operates on the
248 XML for the UDDI entity it is passed, and returns a result. What, if anything, is done with the
249 result is up to the caller. (Typically, though, the caller will be the entity's publisher who will use the
250 result to republish the entity in a subsequent step.)

251 The resulting entity is unchanged except for those keyedReferences that refer to the given value
252 set.

- 253 • Changed keyedReferences (in terms of the document order) indicate that a conversion to
 254 the newest version of the value set, i.e. the one that has no successor, is possible by
 255 using the returned keyedReference instead.
- 256 • Unchanged keyedReferences (in terms of the document order) indicate that a conversion
 257 to the newest version of the value set is not possible without further investigation. This
 258 may be the case if, for example, a value in a predecessor version is replaced by two
 259 values in a successor version.

260 Because of this behavior and depending upon the trustworthiness of the service, users should
 261 inspect the result to ensure that it meets their needs.

262 **Returns:**

263 The updated entities are returned in a UDDI businessDetail, serviceDetail, tModelDetail, or
 264 publisherAssertions message, depending on the type of the entities passed.

265 **Caveats:**

266 If any error occurs in processing this message, a dispositionReport structure will be returned to
 267 the caller in a SOAP Fault. The following error number information will be relevant:

- 268 • **E_invalidKeyPassed:** signifies that the *tModelKey* value passed did not match the key
 269 for a value set tModel that this service is capable of processing.
- 270 • **E_invalidValue:** signifies that a value that was passed in a keyValue attribute does not
 271 belong to the value set's known code space. The error text should clearly indicate the
 272 value. This error does not necessarily indicate that the value set is checked and the value
 273 did not pass validation, it simply indicates that the update_entities Web service does not
 274 recognize the value to be in the value set.

275 **Bindings:**

276 Web services that implement update_entities_v2 should list as tModelInstanceInfo structures, the
 277 tModelKey for uddi-org:updateEntities_v2 and the tModelKey for each value set the Web service
 278 is capable of updating. This convention allows people to find implementations of
 279 update_entities_v2 capable of handling updates to value sets in which they are interested.

280 **tModel definition:**

281	Name:	uddi-org:updateEntities_v2
282	Description:	Service to update UDDI Version 2 entities that make use of
283	value sets	
284	UDDI Key (V3):	uddi:uddi.org:update_entities_v2
285	Derived V1/V2 format key:	uuid:74e22d5a-91fb-3b93-8052-64fb4d281556
286	Categorization:	specification, xmlSpec, soapSpec

287 **XML schema:**

```

288 <?xml version="1.0" encoding="UTF-8"?>
289 <xsd:schema targetNamespace="urn:uddi-org:valueset_versioning_v2"
290 xmlns:uddi="urn:uddi-org:api_v2" xmlns:uddi_vs_versioning="urn:uddi-
291 org:valueset_versioning_v2" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
292 elementFormDefault="qualified" attributeFormDefault="unqualified"
293 id="uddi_vs_versioning_v2">
294 <!-- Copyright (c) OASIS Open 2003, All Rights Reserved.-->
295 <xsd:import namespace="urn:uddi-org:api_v2"
296 schemaLocation="http://uddi.org/schema/uddi_v2.xsd"/>
297 <xsd:element name="update_entities">
298 <xsd:complexType>
299 <xsd:sequence>
300 <xsd:element ref="uddi:tModelKey"/>

```


301
302
303
304
305
306
307
308
309
310

```
<xsd:choice>
  <xsd:element ref="uddi:businessEntity" maxOccurs="unbounded"/>
  <xsd:element ref="uddi:businessService" maxOccurs="unbounded"/>
  <xsd:element ref="uddi:tModel" maxOccurs="unbounded"/>
  <xsd:element ref="uddi:publisherAssertion" maxOccurs="unbounded"/>
</xsd:choice>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

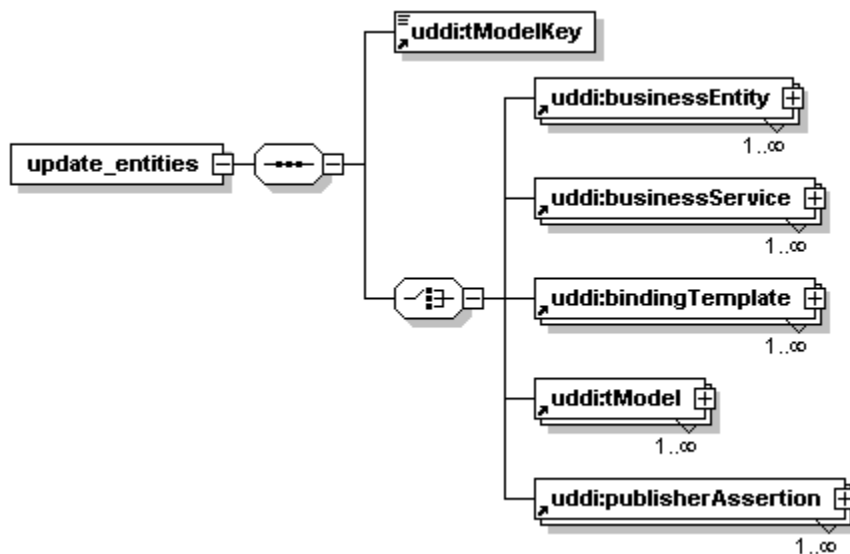
311 2.4.2 update_entities for UDDI Version 3 entities

312 The update_entities Web service type definition for UDDI Version 3 entities is a complete
313 analogue of the one defined for UDDI Version 2 entities (see section above), except that

- 314 • It covers the update of UDDI Version 3 entities instead since the information model and
315 semantics have changed between UDDI Version 2 and 3
- 316 • It covers the update of bindingTemplates since they are allowed to carry categoryBags in
317 UDDI Version 3

318 Therefore, only the changed syntax and XML schema are listed here. The behavior is adopted
319 from the V2 update_entities definition accordingly.

320 Syntax



321

322 tModel definition:

323 Name: uddi-org:updateEntities_v3
324 Description: Service to update UDDI Version 3 entities that make use of
325 value sets
326 UDDI Key (V3): uddi:uddi.org:update_entities_v3
327 Derived V1/V2 format key: uuid:06647e37-66ed-3973-9cce-857a0f43c12b
328 Categorization: specification, xmlSpec, soapSpec

329 XML schema:

330

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

331 <xsd:schema targetNamespace="urn:uddi-org:valueset_versioning_v3"
332 xmlns:uddi="urn:uddi-org:api_v3" xmlns:uddi_vs_versioning_v3="urn:uddi-
333 org:valueset_versioning_v3" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
334 elementFormDefault="qualified" attributeFormDefault="unqualified"
335 id="uddi_vs_versioning_v3">
336 <!-- Copyright (c) OASIS Open 2003, All Rights Reserved.-->
337 <xsd:import namespace="urn:uddi-org:api_v3"
338 schemaLocation="http://uddi.org/schema/uddi_v3.xsd"/>
339 <xsd:element name="update_entities">
340 <xsd:complexType>
341 <xsd:sequence>
342 <xsd:element ref="uddi:tModelKey"/>
343 <xsd:choice>
344 <xsd:element ref="uddi:businessEntity" maxOccurs="unbounded"/>
345 <xsd:element ref="uddi:businessService" maxOccurs="unbounded"/>
346 <xsd:element ref="uddi:bindingTemplate" maxOccurs="unbounded"/>
347 <xsd:element ref="uddi:tModel" maxOccurs="unbounded"/>
348 <xsd:element ref="uddi:publisherAssertion" maxOccurs="unbounded"/>
349 </xsd:choice>
350 </xsd:sequence>
351 </xsd:complexType>
352 </xsd:element>
353 </xsd:schema>

```

354 2.5 Examples

355 2.5.1 Example of uddi-org:isReplacedBy usage

356 In the example below, the UNSPSC Version 3.1 category system has been replaced by the
357 UNSPSC Version 7 category system. To indicate this, the uddi-org:isReplacedBy identifier
358 system is used to point the unspsc-org:unspsc:3-1 tModel to the unspsc-org:unspsc tModel. So
359 to do this, the unspsc-org:unspsc:3-1 tModel has a keyedReference added to its identifierBag, as
360 follows:

361 UDDI Version 2:

```

362 <tModel tModelKey="uuid:db77450d-9fa8-45d4-a7bc-04411d14e384">
363 <name>unspsc-org:unspsc:3-1</name>
364 ...
365 <identifierBag>
366 <!-- Use uddi-org:isReplacedBy to indicate that the UNSPSC 3.1
367 tModel is logically replaced by the UNSPSC 7 tModel. -->
368 <keyedReference
369 keyName="isReplacedBy:unspsc-org:unspsc"
370 keyValue="uuid:cd153257-086a-4237-b336-6bdcdbcc6634"
371 tModelKey="uuid:e59ae320-77a5-11d5-b898-0004ac49cc1e"/>
372 </identifierBag>
373 ...
374 </tModel>

```

375 UDDI Version 3:

```

376 <tModel tModelKey="uddi:db77450d-9fa8-45d4-a7bc-04411d14e384">
377 <name>unspsc-org:unspsc:3-1</name>
378 ...
379 <identifierBag>
380 <!-- Use uddi-org:isReplacedBy to indicate that the UNSPSC 3.1
381 tModel is logically replaced by the UNSPSC 7 tModel. -->
382 <keyedReference
383 keyName="isReplacedBy:unspsc-org:unspsc"
384 keyValue="uddi:cd153257-086a-4237-b336-6bdcdbcc6634"
385 tModelKey="uddi:uddi.org:identifier:isreplacedby"/>

```

386
387
388

```
</identifierBag>  
...  
</tModel>
```

389

390 The keyName attribute is commentary serving to help readability. The keyValue specifies which
391 tModel replaces this one – the unspsc-org:unspsc tModel in this case. The tModelKey specifies
392 that the keyedReference is using the uddi-org:identifier:isReplacedBy identifier system.

393 2.5.2 Example of update_entities Web service

394 In the example below, the bindingTemplate represents a Web service that implements the
395 update_entities Web service type, indicated by the first tModelInstanceInfo that references the
396 uddi-org:update_entities_v2 and uddi-org:update_entities_v3 tModels, respectively. The second
397 tModelInstanceInfo references the UNSPSC 3.1 category system and thus indicates that the Web
398 service is capable in updating values from this version to its predecessor versions.

399 update_entities_v2 in UDDI Version 2:

400
401
402
403
404
405
406
407
408
409
410

```
<bindingTemplate>  
  <accessPoint URLType="http">  
    http://www.example.com/unspsc_conversion  
  </accessPoint>  
  <tModelInstanceDetails>  
    <tModelInstanceInfo  
      tModelKey="TBD" />  
    <tModelInstanceInfo  
      tModelKey="uuid:db77450d-9fa8-45d4-a7bc-04411d14e384" />  
  </tModelInstanceDetails>  
</bindingTemplate>
```

411 update_entities_v3 in UDDI Version 2:

412
413
414
415
416
417
418
419
420
421
422

```
<bindingTemplate>  
  <accessPoint URLType="http">  
    http://www.example.com/unspsc_conversion  
  </accessPoint>  
  <tModelInstanceDetails>  
    <tModelInstanceInfo  
      tModelKey="TBD" />  
    <tModelInstanceInfo  
      tModelKey="uuid:db77450d-9fa8-45d4-a7bc-04411d14e384" />  
  </tModelInstanceDetails>  
</bindingTemplate>
```

423 update_entities_v2 in UDDI Version 3:

424
425
426
427
428
429
430
431
432
433

```
<bindingTemplate>  
  <accessPoint useType="endPoint">  
    http://www.example.com/unspsc_conversion  
  </accessPoint>  
  <tModelInstanceDetails>  
    <tModelInstanceInfo tModelKey="uddi:uddi.org:update_entities_v2" />  
    <tModelInstanceInfo  
      tModelKey="uddi:db77450d-9fa8-45d4-a7bc-04411d14e384" />  
  </tModelInstanceDetails>  
</bindingTemplate>
```

434 **update_entities_v3 in UDDI Version 3:**

```
435 <bindingTemplate>
436   <accessPoint useType="endPoint">
437     http://www.example.com/unspsc_conversion
438   </accessPoint>
439   <tModelInstanceDetails>
440     <tModelInstanceInfo tModelKey="uddi:uddi.org:update_entities_v3" />
441     <tModelInstanceInfo
442       tModelKey="uddi:db77450d-9fa8-45d4-a7bc-04411d14e384" />
443   </tModelInstanceDetails>
444 </bindingTemplate>
```

445 **2.5.3 Example of update_entities API call**

446 In the example below, an update_entities API call is used to update a businessEntity with respect
447 to the UNSPSC 3.1 category system, indicated by the tModelKey uddi:db77450d-9fa8-45d4-
448 a7bc-04411d14e384. The businessEntity contains a categoryBag that in turn contains two
449 keyedReferences, both representing UNSPSC 3.1 categories.

450 **UDDI Version 2:**

```
451 <update_entities xmlns="urn:uddi-org:valueset_versioning_v2"
452 xmlns:uddi="urn:uddi-org:api_v2">
453   <uddi:tModelKey>uuid:db77450d-9fa8-45d4-a7bc-04411d14e384</tModelKey>
454   <uddi:businessEntity businessKey="...">
455     ...
456     <uddi:categoryBag>
457       <uddi:keyedReference
458         ...
459         tModelKey="uuid:db77450d-9fa8-45d4-a7bc-04411d14e384" />
460       <uddi:keyedReference
461         ...
462         tModelKey="uuid:db77450d-9fa8-45d4-a7bc-04411d14e384" />
463     </uddi:categoryBag>
464     ...
465   </uddi:businessEntity>
466 </update_entities>
```

467 The businessDetail return structure shown below contains the businessEntity and two
468 keyedReferences, each corresponding to the ones given in the update_entities API call. Since the
469 first keyedReference still refers to the UNSPSC 3.1 category system, the update_entity Web
470 service was obviously not able to find an appropriate category in the UNSPSC 7 successor
471 version. The second keyedReference refers to the UNSPSC 7 category system and thus, was
472 updated.

```
473 <businessDetail xmlns="urn:uddi-org:api_v2">
474   <businessEntity businessKey="...">
475     ...
476     <categoryBag>
477       <keyedReference
478         ...
479         tModelKey="uuid:db77450d-9fa8-45d4-a7bc-04411d14e384" />
480       <keyedReference
481         ...
482         tModelKey="uuid:cd153257-086a-4237-b336-6bdcdbcc6634" />
483     </categoryBag>
484     ...
485   </businessEntity>
486 </businessDetail>
```

487 **UDDI Version 3:**

```
488 <update_entities xmlns="urn:uddi-org:valueset_versioning_v3"
489 xmlns:uddi="urn:uddi-org:api_v3">
490 <uddi:tModelKey>uddi:db77450d-9fa8-45d4-a7bc-04411d14e384</tModelKey>
491 <uddi:businessEntity businessKey="...">
492 ...
493 <uddi:categoryBag>
494 <uddi:keyedReference
495 ...
496 tModelKey="uddi:db77450d-9fa8-45d4-a7bc-04411d14e384" />
497 <uddi:keyedReference
498 ...
499 tModelKey="uddi:db77450d-9fa8-45d4-a7bc-04411d14e384" />
500 </uddi:categoryBag>
501 ...
502 </uddi:businessEntity>
503 </update_entities>
```

504 The businessDetail return structure shown below contains the businessEntity and two
505 keyedReferences, each corresponding to the ones given in the update_entities API call. Since the
506 first keyedReference still refers to the UNSPSC 3.1 category system, the update_entity Web
507 service was obviously not able to find an appropriate category in the UNSPSC 7 successor
508 version. The second keyedReference refers to the UNSPSC 7 category system and thus, was
509 updated.

```
510 <businessDetail xmlns="urn:uddi-org:api_v3">
511 <businessEntity businessKey="...">
512 ...
513 <categoryBag>
514 <keyedReference
515 ...
516 tModelKey="uddi:db77450d-9fa8-45d4-a7bc-04411d14e384" />
517 <keyedReference
518 ...
519 tModelKey="uddi:cd153257-086a-4237-b336-6bdcbdcc6634" />
520 </categoryBag>
521 ...
522 </businessEntity>
523 </businessDetail>
```

524 **3 References**

525 **3.1 Normative**

- 526 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
527 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 528 **[UDDIV3]** L. Clément et al, *UDDI Version 3.0 Specification*,
529 http://uddi.org/pubs/uddi_v3.htm, OASIS UDDI Spec TC Committee
530 Specification, July 2002.

531 **Appendix A. Acknowledgments**

532 The editor would like to thank David Ehnebuske and Barbara McKee (both IBM) for their initial
533 contribution of this technical note.

534

Appendix B. Revision History

535

Rev	Date	By Whom	What
1.09	1/02/2003	Pat Felsted	Updated to UDDI Version 3 Product Specification and Oasis technical note format.
1.10	22/4/2003	Claus von Riegen	<ul style="list-style-type: none">• Aligned terminology• Separated between normative UDDI terminology and TN definitions• Changed example to UNSPSC• Added two examples• Updated update_entities Web service behavior• Added XML schema for update_entities Web service
1.11	27/5/2003	Pat Felsted	Initial update for version 2 compatibility.
1.12	29/8/2003	Claus von Riegen	Final update with regard to UDDI Version 2

536

Appendix C. Notices

538 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
539 that might be claimed to pertain to the implementation or use of the technology described in this
540 document or the extent to which any license under such rights might or might not be available;
541 neither does it represent that it has made any effort to identify any such rights. Information on
542 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
543 website. Copies of claims of rights made available for publication and any assurances of licenses
544 to be made available, or the result of an attempt made to obtain a general license or permission
545 for the use of such proprietary rights by implementors or users of this specification, can be
546 obtained from the OASIS Executive Director.

547 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
548 applications, or other proprietary rights which may cover technology that may be required to
549 implement this specification. Please address the information to the OASIS Executive Director.

550 **Copyright © OASIS Open 2003. All Rights Reserved.**

551 This document and translations of it may be copied and furnished to others, and derivative works
552 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
553 published and distributed, in whole or in part, without restriction of any kind, provided that the
554 above copyright notice and this paragraph are included on all such copies and derivative works.
555 However, this document itself does not be modified in any way, such as by removing the
556 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS
557 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
558 Property Rights document must be followed, or as required to translate it into languages other
559 than English.

560 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
561 successors or assigns.

562 This document and the information contained herein is provided on an "AS IS" basis and OASIS
563 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
564 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
565 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
566 PARTICULAR PURPOSE.