UDDI Spec TC

Technical Note

Providing A Value Set For Use In UDDI Version 3

Document identifier:

uddi-spec-tc-tn-valuesetprovider-20030212

Location:

http://oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-valuesetprovider-20030212.htm

Editor:

Claus von Riegen, SAP

Contributors:

Tom Bellwood, IBM

Abstract:

Through the use of value sets in UDDI registries, businesses are able to find each other and the services that meet their needs. This document provides guidelines for providers of value sets on how to model, register, and validate their value sets for use in UDDI Version 3.

Status:

This document is a working draft.

Committee members should send comments on this technical note to the uddi-spec@lists.oasis-open.org list. Others should subscribe to and send comments to the uddi-spec-comment@lists.oasis-open.org list. To subscribe, send an email message to uddi-spec-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

Table of Contents

1 Introduction

In UDDI, a value set represents a set of values that can be used to provide meaning or context to a UDDI entity. Category, identifier, and relationship type systems are all value sets. Value sets play an important role within UDDI, because it is through their use that businesses are able to find each other and the services that meet their needs.

1.1 Problem statement

This paper guides the providers of value sets in the creation of value set services and in the registration of the value sets and these external value set services, following the recommended policies outlined in Chapter 9 of the UDDI Version 3 Specification **[UDDIV3]**

1.2 Terminology

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

2 Technical Note Solution

2.1 Definitions

A provider of a value set can allow unrestricted use or may choose to have references to it validated.

Unchecked value set

A value set that allows unrestricted references.

Checked value set

A value set that, each time an attempt is made to save data containing a reference to it, requires validation of this reference.

Cacheable value set

A checked value set that is checked against a private cache of values.

Externally validated value set

A checked value set that has an associated validation service that performs value checking.

2.2 Technical note behavior

2.2.1 Unchecked value sets

To make an unchecked value set available within a UDDI registry, a *tModel* is published. Information on registering a tModel can be found in Section 5.2.18 *save_tModel* in the UDDI Version 3 Specification.

The *name* of the value set should contain a URI that reveals its scope, its intent, or both. The value set's *description* briefly describes what the value set is used for and may be provided in multiple languages. The *overviewDoc* refers to a document describing the details of the value set, including where to find (or how to compute) values. In order to type the value set as being an unchecked one, it should be categorized with the value unchecked[1] using the value set named **uddi-org:types**[2]. Depending on the value set's type, it should also be categorized with one of the values "*categorization", "identifier", "relationship"* or "*categorizationGroup"* of the uddi-org:types value set.

Once the tModel for the unchecked value set is published in a UDDI registry it becomes immediately available for use in that registry. The tModel for an unchecked value set would look similar to this one:

```
<tModel tModelKey="uddi:example.com:categorization:partner_types">
   <name>example-com:partner_types</name>
   <description xml:lang="en">
      Extendable value set used to categorize my partners.
   </description>
```

[1] The *unchecked* and *checked* categorizations are the recommended means for determining whether a value set is checked or unchecked. During a save operation the UDDI implementation will attempt to find and execute acceptable validation algorithms for all checked value sets referenced in uddi:keyedReference or uddi:keyedReferenceGroup elements.

[2] This tModel's key is "uddi:uddi.org:categorization:types". See Section 11.1.1 *UDDI Types Category System* in the UDDI Version 3 Specification.

```
   <overviewDoc>
      <description xml:lang="en">Initial list of types.</description>
      <overviewURL>http://example.com/partner_types.html</overviewURL>
   </overviewDoc>
   <categoryBag>
      <keyedReference
         tModelKey="uddi:uddi.org:categorization:types"
         keyName="types:categorization"
         keyValue="categorization"/>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:unchecked"
         keyValue="unchecked"/>
   </categoryBag>
</tModel>
```

2.2.2 Checked value sets

A checked value set is one whose use is restricted to a set of valid values. Every time a UDDI data structure is saved that contains one or more references (in terms of keyedReference elements) to checked value sets, the references are validated or the save is rejected, indicating that the chosen value does not belong to the set of valid values. A common, simple validation algorithm is one that verifies that referenced values are from a pre-defined set. Value sets that have this kind of validation algorithm and which allow the set of valid values to be cached for satisfying the algorithm are considered *cacheable*. Contextual and more complex validation algorithms are *uncacheable*.

UDDI implementations may obtain sets of valid values for cacheable checked value sets through private means, through periodic invocation of a *get_allValidValues*[3] Web service or through incremental collection of acceptable values obtained from invoking a *validate_values* Web service. With caches of valid values, UDDI implementations can perform simple in-set validations themselves. The designation of the value set as *cacheable* is the recommended way for communicating this style of validation.

If a value set is not designated as cacheable, it is by default *uncacheable*. Validation algorithms for uncacheable checked value sets are not intuitively obvious. While it is possible for a UDDI implementation to gain private access to the validation algorithm for such a value set, the more typical means of gaining access to the validation algorithm is through the invocation of a validate_values Web service that is associated with the value set.

Value set providers must insure that they conform to the registry policies for value sets as described in Section 9.4.19 *Registry Value Set Policies* in the UDDI Version 3 Specification. A UDDI registry must decide if it supports checked value sets at all and if so, how it differentiates between checked and unchecked value sets.

This often means that value set provider and registry negotiate terms and conditions for validation services before a checked value set can actually be used. In the time period between the first save of the checked value set tModel and the acceptance of the registry, all save operations of UDDI data structures that contain one or more references to the checked value sets are rejected.

2.2.2.1 Providing a cacheable checked value set using *get_allValidValues*

A get_allValidValues Web service can be used to obtain the set of values that are valid for a value set. The tModel for this API, named **uddi-org:valueSetCaching_v3**[4], refers to the

[3] See Section 5.6 *Value Set API Set* in the UDDI Version 3 Specification.

[4] This tModel's key is "uddi:uddi.org:v3_valueSetCaching". See Section 11.2.7 *UDDI Value Set Caching API* in the UDDI Version 3 Specification.

uddi_vscache_v3_binding.wsdl file containing the message and binding information for invoking *get_allValidValues* using SOAP over HTTP.

There are four steps involved in providing a cacheable checked value set using get_allValidValues.

1. A get_allValidValues Web service that can return the set of values, either in its entirety or in chunks, should be designed and deployed. This is described in Section 5.6.3 *get_allValidValues* in the UDDI Version 3 Specification.
2. A tModel for the cacheable checked value set should be published in the desired UDDI registry. The tModel for a checked value set is the same as a tModel for an unchecked value set (see above), with three exceptions. First, the tModel is categorized with the value "checked" instead of the value "unchecked" using the **uddi-org:types** value set. Second, the tModel is categorized with the value "cacheable" also from the **uddi-org:types** category system. And third, the document that the overviewURL refers to should include any restrictions on the use of values in addition to where to find the values. An example of a typical tModel for a cacheable checked value set follows.

```
<tModel tModelKey="uddi:example.com:categorization:realtor_types">
   <name>example.com:realtor_types</name>
   <description xml:lang="en">
      Fixed value set used to categorize real estate firms.
   </description>
   <overviewDoc>
      <description xml:lang="en">Value set of real estate firm
         categorizations. Only listed values can be referenced.
         Offered to licensed members only.
      </description>
      <overviewURL>
         http://example.com/realtor_types.html
      </overviewURL>
   </overviewDoc>
   <categoryBag>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:categorization"
         keyValue="categorization"/>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:checked"
         keyValue="checked"/>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:cacheable"
         keyValue="cacheable"/>
   </categoryBag>
</tModel>
```

3. A bindingTemplate for the get_allValidValues Web service should be published in the same UDDI registry as the tModel. The technical fingerprint should reference the **uddi-org:valueSetCaching_v3** tModel as well as the tModel(s) for all value sets for which the set of values applies.[5] A bindingTemplate that represents the get_allValidValues Web service for the value set above follows.

[5] Derived value sets often share the same set of values as the value sets that they are derived from. Value set Web services that apply to derived value sets as well as the base value sets should reference in their technical fingerprints the tModels for the derived value sets that share the same set of values as the base value sets.

```
<bindingTemplate
   bindingKey="uddi:example.com:get_allValidValues:realtor_types"
   serviceKey="uddi:example.com:valueSetService">
   <description xml:lang="en">
      Returns the set of valid values for the realtor_types value set.
   </description>
   <accessPoint useType="endPoint">http://URL_of_service</accessPoint>
   <tModelInstanceDetails>
      <tModelInstanceInfo
         tModelKey="uddi:uddi.org:v3_valueSetCaching"/>
      <tModelInstanceInfo
         tModelKey="uddi:example.com:categorization:realtor_types"/>
   </tModelInstanceDetails>
</tModel>
```

4. The tModel should be updated to refer to the get_allValidValues Web service that is associated with the checked value set. The **uddi-org:validatedBy** category system is used to point the tModel to the bindingTemplate of the get_allValidValues Web service that the value set provider claims can provide the set of values. Recommended policy is that the bindingTemplate only applies to the value set if it references the tModel of the value set in its technical fingerprint. Re-publishing the tModel is the recommended technique for triggering cache invalidation. Thus when this final piece of the plumbing is put into place in the tModel, implementations that follow the recommended policy for handling cacheable checked value sets will asynchronously invoke the get_allValidValues service that the uddi:tModel refers to. The updated version of the example tModel for a cacheable checked value set follows.

```
<tModel tModelKey="uddi:example.com:categorization:realtor_types">
   <name>example.com:realtor_types</name>
   <description xml:lang="en">
      Fixed value set used to categorize real estate firms.
   </description>
   <overviewDoc>
      <description xml:lang="en">Value set of real estate firm
         categorizations. Only listed values can be referenced.
         Offered to licensed members only.
      </description>
      <overviewURL>
         http://example.com/realtor_types.html
      </overviewURL>
   </overviewDoc>
   <categoryBag>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:categorization"
         keyValue="categorization"/>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:checked"
         keyValue="checked"/>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:cacheable"
         keyValue="cacheable"/>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:validatedBy"
         keyName="validatedBy:get_allValidValues:realtor_types"
         keyValue=
            "uddi:example.com:get_allValidValues:realtor_types"/>
```

```
   </categoryBag>
</tModel>
```

2.2.2.2 Providing an externally validated checked value set

For checked value sets for which there is no private validation algorithm, a validate_values Web service may be used during a save operation to verify that checked value set references are valid. This is often the case when contextual checks are performed on the data being saved, or when the value set provider has declined to allow value set values to be cached.

A reason for a value set being externally validated may be the value set provider's requirement to check that references to the value set occur in businessEntity data structures only and that the businessEntity's name matches a corresponding string stored in a directory owned by the value set provider. Hence, the value set provider can guarantee that values are not inconsistent with other UDDI data within the same UDDI data structure.

Even more comprehensive contextual checks, such as ensuring that the business using them is known or perhaps approved by the value set provider prior to such use, or even the observation of digital signatures that are added to the data structure in which the value set reference occurs, are conceivable. In this latter case, the value set provider can guarantee that values are used authentically. For example, business identifiers can't be used by a business that does not own the identifier, hence making identifier-based queries more reliable.

The tModel for this API, named **uddi-org:valueSetValidation_v3**[6], refers to the **uddi_vs_v3_binding.wsdl** file containing the message and binding information for invoking validate_values using SOAP over HTTP.

There are four steps involved in providing an externally validated checked value set.

1. A validate_values Web service that verifies the acceptability of referenced values should be written and deployed. This is described in Section 5.6.2 *validate_values* in the UDDI Version 3 Specification. See Performing external validation below for more information related to the actual validation requirements.
2. A tModel for the checked value set should be published in the desired UDDI registry. The tModel for a checked value set is the same as a tModel for an unchecked value set, with a couple of exceptions[7]. First, the tModel is categorized with the value "checked" instead of the value "unchecked" from the **uddi-org:types** category system. And second, the document that the overviewURL refers to should include any restrictions on the use of values in addition to where to find the values. An example of a typical externally validated checked value set tModel follows.

```
<tModel tModelKey="uddi:example.com:categorization:realtor_types">
   <name>example.com:realtor_types</name>
   <description xml:lang="en">
      Fixed value set used to categorize real estate firms.
   </description>
   <overviewDoc>
      <description xml:lang="en">Value set of real estate firm
         categorizations.  Only listed values can be referenced.
         Offered to licensed members only.
      </description>
      <overviewURL>
         http://example.com/realtor_types.html
```

[6] This tModel's key is "uddi:uddi.org:v3_valueSetValidation". See Section 11.2.8 *UDDI Value Set Validation API* in the UDDI Version 3 Specification.

[7] Also, if the value set is cacheable and the cache is built incrementally based on successful calls to *validate_values*, the uddi:tModel is categorized with the *cacheable* value from the **uddi-org:types** category system.

```
      </overviewURL>
   </overviewDoc>
   <categoryBag>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:categorization"
         keyValue="categorization"/>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:checked"
         keyValue="checked"/>
   </categoryBag>
</tModel>
```

3. A bindingTemplate for the validate_values Web service should be published in the same UDDI registry as the tModel. The technical fingerprint should reference the **uddi-org:valueSetValidation_v3** tModel as well as the tModel(s) for all value sets for which the set of values applies. A bindingTemplate that represents the get_allValidValues Web service for the value set above follows.

```
<bindingTemplate
   bindingKey="uddi:example.com:validate_values:realtor_types"
   serviceKey="uddi:example.com:valueSetService">
   <description>
      Validates values for the realtor_types Value set.
   </description>
   <accessPoint useType="endPoint">http://URL_of_service</accessPoint>
   <tModelInstanceDetails>
      <tModelInstanceInfo
         tModelKey="uddi:uddi.org:v3_valueSetValidation"/>
      <tModelInstanceInfo
         tModelKey="uddi:example.com:categorization:realtor_types"/>
   </tModelInstanceDetails>
</tModel>
```

4. The tModel should be updated to refer to the validate_values Web service that is associated with the checked value set. This could also apply to cacheable value sets. The **uddi-org:validatedBy** category system is used to point the tModel to the bindingTemplate of the validate_values Web service that the value set provider claims can validate reference values. Recommended policy is that the bindingTemplate only applies to the value set if it references the tModel of the value set in its technical fingerprint. The updated version of the example tModel for an externally checked value set follows.

```
<tModel tModelKey="uddi:example.com:categorization:realtor_types">
   <name>example.com:realtor_types</name>
   <description xml:lang="en">
      Fixed value set used to categorize real estate firms.
   </description>
   <overviewDoc>
      <description xml:lang="en">Value set of real estate firm
         categorizations.  Only listed values can be referenced.
         Offered to licensed members only.
      </description>
      <overviewURL>
         http://example.com/realtor_types.html
      </overviewURL>
   </overviewDoc>
```

```
   <categoryBag>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:categorization"
         keyValue="categorization"/>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:types"
         keyName="types:checked"
         keyValue="checked"/>
      <keyedReference
         tModelKey="uddi:uddi-org:categorization:validatedBy"
         keyName="validatedBy:validate_values:realtor_types"
         keyValue="uddi:example.com:validate_values:realtor_types"/>
   </categoryBag>
</tModel>
```

2.2.2.2.1 Performing external validation

The validation service for a checked value set is invoked by the registry to validate attempts to publish entities that reference that value set. When an entity being published contains at least one keyedReference within it that references the tModelKey of the checked value set, the value set's associated validation service is invoked (unless the values specified in the entity are cached). The service is passed the complete entity being published. The validation service must inspect each relevant keyedReference in the entity for invalid values.

Because UDDI nodes call validation services synchronously during a publish operation, optimization of validation service invocations is a common practice. The frequency with which the validation service is called for an entity being saved that contains multiple references to the same value set or for value sets that share a common validation service cannot be assumed. Thus, if a validation service is capable of validating several value sets, and the validation service bindings for each of the value sets share the same access point, the validation service must validate in a single invocation all the references from all the value sets systems that it knows about in the entity passed.

A validation service must:

1. Check the validity of all keyedReference elements it is passed that are associated with checked value sets it is set up to validate.
2. Return the result of the validation in a *dispositionReport.*[8] If all applicable keyedReference elements are valid, ***success*** is returned in the *dispositionReport*. If validation fails for any keyedReference that the validation service is set up to validate, an error message containing at least one failing keyValue and its reason for failure should be captured in the errInfo element content and returned in a disposition report with the ***E_invalidValue*** or ***E_valueNotAllowed*** error. The validation service may stop after the first invalid keyedReference is detected, or it may check all of the applicable values.

[8] See Section 4.8 *Success and Error Reporting* in the UDDI Version 3 Specification.

3 References

3.1 Normative

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[UDDIV3] L. Clement et al, *UDDI Version 3.0 Specification*, http://uddi.org/pubs/uddi-v3.00-published-20020719.pdf, OASIS UDDI Spec TC Committee Specification, July 2002.

Appendix A. Acknowledgments

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

Appendix B. Revision History

[This appendix is optional, but helpful.]

Rev	Date	By Whom	What
0.1	October 24, 2002	Claus von Riegen	• Adoption of the UDDI Spec TC Template for Technical Notes • Detailed content check • Adjustment of formatting
0,2	December 4, 2002	Claus von Riegen	Section about registry policy for value sets added validate_value Web services can also apply to cacheable value sets Usage scenarios for externally checked value sets added

Appendix C. Notices

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

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