



## WSDL Technical Note

**Document identifier:**

~~[wsdl-TN-V2.00-Draft-20021022](#)~~

**Deleted:** wsdl-TN-V2.00-Draft-20021021

**Location:**

[http://tbd]

**Authors (alphabetically):**

John Colgrave, IBM [colgrave@uk.ibm.com](mailto:colgrave@uk.ibm.com)  
Karsten Januszewski, Microsoft [karstenj@microsoft.com](mailto:karstenj@microsoft.com)

**Abstract:**

This document is an OASIS UDDI Technical Note that defines a new approach to using WSDL in a UDDI Registry.

**Status:**

This document is a working draft.

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

**Deleted:** wsdl-TN-V2.00-Draft-20021021wsdl-TN-V2.00-Draft-20020926

## 21 Copyright

22 *Copyright (C) OASIS Open October 2002. All Rights Reserved.*

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

30 *The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or*  
31 *assigns.*

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

Deleted: wsdl-TN-V2.00-Draft-20021021

36	<a href="#">Table of Contents Copyright</a> .....	2
37	<a href="#">Table of Contents</a> .....	3
38	<a href="#">1 Introduction</a> .....	6
39	<a href="#">1.1 Goals and Requirements</a> .....	6
40	<a href="#">1.2 Relationship to Version 1 Best Practice</a> .....	6
41	<a href="#">1.3 tModel Keys</a> .....	7
42	<a href="#">2 Rationalizing Two Data Models: WSDL &amp; UDDI</a> .....	7
43	<a href="#">2.1 WSDL Data Model</a> .....	7
44	<a href="#">2.1.1 portType</a> .....	7
45	<a href="#">2.1.2 binding</a> .....	7
46	<a href="#">2.1.3 service and port</a> .....	7
47	<a href="#">2.1.4 import</a> .....	7
48	<a href="#">2.2 UDDI Data Model</a> .....	8
49	<a href="#">2.2.1 tModels</a> .....	8
50	<a href="#">2.2.2 businessService &amp; bindingTemplate</a> .....	8
51	<a href="#">2.3 Mapping WSDL and UDDI</a> .....	8
52	<a href="#">2.4 References to WSDL Components</a> .....	9
53	<a href="#">2.5 WSDL Extensibility Elements</a> .....	9
54	<a href="#">2.6 Mapping WSDL 1.1 in UDDI V2</a> .....	9
55	<a href="#">2.6.1 wsdl:portType → uddi:tModel</a> .....	9
56	<a href="#">2.6.2 wsdl:binding → uddi:tModel</a> .....	9
57	<a href="#">2.6.3 wsdl:service → uddi:businessService</a> .....	10
58	<a href="#">2.6.4 wsdl:port and wsdl:binding → uddi:bindingTemplate</a> .....	10
59	<a href="#">2.7 Mapping WSDL 1.1 in UDDI V3</a> .....	11
60	<a href="#">2.7.1 wsdl:portType → uddi:tModel</a> .....	11
61	<a href="#">2.7.2 wsdl:binding → uddi:tModel</a> .....	12
62	<a href="#">2.7.3 wsdl:service → uddi:businessService</a> .....	12
63	<a href="#">2.7.4 wsdl:port and wsdl:binding → uddi:bindingTemplate</a> .....	12
64	<a href="#">3 A Complete Example</a> .....	14
65	<a href="#">3.1 WSDL Sample</a> .....	14
66	<a href="#">3.2 UDDI V2 Model</a> .....	15
67	<a href="#">3.2.1 UDDI portType tModel</a> .....	15
68	<a href="#">3.2.2 UDDI binding tModel</a> .....	15
69	<a href="#">3.2.3 UDDI businessService and bindingTemplate</a> .....	15
70	<a href="#">3.3 Sample V2 Queries</a> .....	17
71	<a href="#">3.3.1 Find tModel for portType name</a> .....	17
72	<a href="#">3.3.2 Find bindings for portType</a> .....	17
73	<a href="#">3.3.3 Find Implementations of portType</a> .....	17
74	<a href="#">3.3.4 Find implementations of binding</a> .....	17
75	<a href="#">3.3.5 Find SOAP Implementations of portType</a> .....	17
76	<a href="#">3.3.6 Find SOAP/HTTP Implementations of portType</a> .....	18
77	<a href="#">3.3.7 Find the portType of a binding</a> .....	18
78	<a href="#">3.4 UDDI V3 Model</a> .....	18
79	<a href="#">3.4.1 UDDI portType tModel</a> .....	18
80	<a href="#">3.4.2 UDDI binding tModel</a> .....	18

Deleted: wsdl-TN-V2.00-Draft-20021021

81	3.4.3	UDDI businessService and bindingTemplate .....	19
82	3.5	Sample V3 Queries.....	20
83	3.5.1	Find tModel for portType name .....	20
84	3.5.2	Find bindings for portType.....	20
85	3.5.3	Find Implementations of portType.....	20
86	3.5.4	Find Implementations of binding .....	21
87	3.5.5	Find SOAP Implementations of portType .....	21
88	3.5.6	Find SOAP/HTTP Implementations of portType.....	21
89	3.5.7	Find the portType of a binding.....	21
90	4	References .....	21
91		..... Appendix A: Canonical tModels	
92		.....	21
93	A.1	WSDL Entity Type tModel.....	21
94	A.1.1	Design Goals .....	21
95	A.1.2	Definition .....	21
96	A.1.3	Values .....	22
97	A.1.4	Example of Use .....	22
98	A.2	WSDL Namespace tModel.....	22
99	A.2.1	Design Goals .....	22
100	A.2.2	Definition.....	22
101	A.2.3	Values .....	23
102	A.2.4	Example of Use .....	23
103	A.3	WSDL Local Name tModel.....	23
104	A.3.1	Design Goals .....	23
105	A.3.2	Definition .....	23
106	A.3.3	Values .....	24
107	A.3.4	Example of Use .....	24
108	A.4	WSDL portType Reference tModel .....	24
109	A.4.1	Design Goals .....	24
110	A.4.2	Definition .....	24
111	A.4.3	Values .....	24
112	A.4.4	Example of Use .....	24
113	A.5	WSDL URL Reference tModel .....	25
114	A.5.1	Design Goals .....	25
115	A.5.2	Definition .....	25
116	A.5.3	Values .....	25
117	A.5.4	Example of Use .....	25
118	A.6	SOAP Protocol tModel.....	25
119	A.6.1	Design Goals .....	25
120	A.6.2	Definition.....	26
121	A.6.3	Example of Use .....	26
122	A.7	HTTP Protocol tModel.....	26
123	A.7.1	Design Goals .....	26
124	A.7.2	Definition .....	26

Deleted: wsdl-TN-V2.00-Draft-20021021

125	<a href="#">A.7.3 Example of Use</a>	<a href="#">27</a>
126	<a href="#">A.8 HTTP SOAP Transport tModel</a>	<a href="#">27</a>
127	<a href="#">A.8.1 Design Goals</a>	<a href="#">27</a>
128	<a href="#">A.8.2 Definition</a>	<a href="#">27</a>
129	<a href="#">A.8.3 Example of Use</a>	<a href="#">28</a>
130	<a href="#">Appendix B: Using XPointer in overviewURL</a>	
131		<a href="#">28</a>
132	<a href="#">B.1 XPointer Syntax</a>	<a href="#">28</a>

**Deleted: ¶**

- ¶
- [Copyright](#) 2¶
- [Table of Contents](#) 3¶
- [1 Introduction](#) 6¶
- [1.1 Goals and Requirements](#) 6¶
- [1.2 Relationship to Version 1 Best Practice](#) 6¶
- [1.3 tModel Keys](#) 7¶
- [2 Rationalizing Two Data Models: WSDL & UDDI](#) 7¶
- [2.1 WSDL Data Model](#) 7¶
- [2.1.1 portType](#) 7¶
- [2.1.2 binding](#) 7¶
- [2.1.3 service and port](#) 7¶
- [2.1.4 import](#) 7¶
- [2.2 UDDI Data Model](#) 8¶
- [2.2.1 tModels](#) 8¶
- [2.2.2 businessService & bindingTemplate](#) 8¶
- [2.3 Mapping WSDL and UDDI](#) 8¶
- [2.4 References to WSDL Components](#) 9¶
- [2.5 Mapping WSDL 1.1 in UDDI V2](#) 9¶
- [2.5.1 wsdl:portType → uddi:tModel](#) 9¶
- [2.5.2 wsdl:binding → uddi:tModel](#) 9¶
- [2.5.3 wsdl:service → uddi:businessService](#) 10¶
- [2.5.4 wsdl:port → uddi:bindingTemplate](#) 10¶
- [2.5.5 soap:address → uddi:accessPoint](#) 10¶
- [2.6 Mapping WSDL 1.1 in UDDI V3](#) 10¶
- [2.6.1 wsdl:portType → uddi:tModel](#) 11¶
- [2.6.2 wsdl:binding → uddi:tModel](#) 11¶
- [2.6.3 wsdl:service → uddi:businessService](#) 11¶
- [2.6.4 wsdl:port → uddi:bindingTemplate](#) 12¶
- [2.6.5 soap:address → uddi:accessPoint](#) 12¶
- [3 A Complete Example](#) 12¶
- [3.1 WSDL Sample](#) 12¶
- [3.2 UDDI V2 Model](#) 13¶
- [3.2.1 UDDI portType tModel](#) 13¶
- [3.2.2 UDDI binding tModel](#) 14¶
- [3.2.3 UDDI businessService and bindingTemplate](#) 14¶
- [3.3 Sample V2 Queries](#) 15¶
- [3.3.1 Find tModel for the portType StockQuotePortType in the namespace http://example.com/stockquote/](#) 15¶
- [3.3.2 Find bindings for portType](#) 15¶
- [3.3.3 Find Implementations of portType](#) 16¶
- [3.3.4 Find implementations of binding](#) 16¶
- [3.3.5 Find SOAP Implementations of portType](#) 16¶
- [3.3.6 Find SOAP/HTTP Implementations of](#) 16¶

... [1]

**Deleted:** wsdl-TN-V2.00-Draft-20021021

## 134 1 Introduction

135 The Universal Description Discovery and Integration (UDDI) specification provides a platform-independent way of  
136 describing and discovering Web services and Web services providers. The UDDI data structures provide a  
137 framework for the description of basic service information, and architects an extensible mechanism to provide  
138 detailed service access information using any standard description language. Many such languages exist in  
139 specific industry domains and at different levels of the protocol stack. The Web Services Description Language  
140 (WSDL) is a general purpose XML language for describing the interface, protocol bindings and the deployment  
141 details of network services. WSDL complements the UDDI standard by providing a uniform way of describing the  
142 abstract interface and protocol bindings of arbitrary network services. The purpose of this document is to clarify the  
143 relationship between the two, describe how WSDL can be used to help create UDDI business service descriptions.  
144 The importance of mapping WSDL consistently and thoroughly is critical to the utility of UDDI.

### 145 1.1 Goals and Requirements

146 The primary goals of this new mapping are to represent sufficient information from the WSDL documents to allow  
147 the following types of queries without further recourse to the source WSDL documents, and to allow the appropriate  
148 WSDL documents to be retrieved once a match has been found. Given that the source WSDL documents can be  
149 distributed among the publishers using a UDDI registry, a UDDI registry provides a convenient central point where  
150 such queries can be executed.

151 This new best practice would satisfy the following types of queries for both design-time and run-time discovery:

- 152 1) Given the namespace and/or local name of a wsdl:portType, find the tModel that represents that  
153 portType.
- 154 2) Given a tModel representing a portType, find all tModels representing bindings for that portType.
- 155 3) Given a tModel representing a portType, find all bindingTemplates that represent implementations of  
156 that portType.
- 157 4) Given a tModel representing a binding, find all bindingTemplates that represent implementations of that  
158 binding.
- 159 5) Find all bindingTemplates that represent implementations of an extended binding, for example all  
160 SOAP/HTTP implementations, of a portType.

161 Some aspects of the mapping allow information to be retrieved directly without further queries being necessary.  
162 For example, given the tModel representing a binding, it is possible to retrieve the key of the tModel representing  
163 the portType that the binding refers to. Other aspects of the mapping may require multiple queries to be issued to  
164 UDDI.

165 A further goal is to allow deployment WSDL to be generated. With this new mapping, it is possible to generate a  
166 wsdl:service element directly from the mapping in UDDI.

### 167 1.2 Relationship to Version 1 Best Practice

168 This document builds on *Using WSDL in a UDDI Registry, Version 1.07* [2], providing an expanded modeling  
169 practice which encompasses the flexibility of WSDL.

170 As a Technical Note, this document does not replace the existing Best Practice. If the additional flexibility is not  
171 required the existing Best Practice can continue to be used, particularly when the UDDI model is published  
172 manually.

173 It is anticipated that implementations of the approach described in this Technical Note will be developed and that  
174 once experience with those implementations is obtained this Technical Note will become a Best Practice.

175 A final goal is to be compatible with the existing Best Practice in that a model published using the approach  
176 described in this document should be usable by a client that uses the Version 1 Best Practice approach.

Deleted: wsdl-TN-V2.00-Draft-20021021

## 177 1.3 tModel Keys

178 The tModel keys presented in this draft document are fictitious. When this document is approved the real tModels  
179 will be published to the UDDI Business Registry and the tModel keys in this document will be updated to the real  
180 values.

## 181 2 Rationalizing Two Data Models: WSDL & UDDI

182 A brief discussion of the two respective data models, WSDL and UDDI, follows. For a complete explanation of the  
183 these specifications, see [1] and [3].

### 184 2.1 WSDL Data Model

185 A review of WSDL in the context of the goals and requirements will help guide a new mapping practice in UDDI.

#### 186 2.1.1 portType

187 The central construct in WSDL is the portType. A portType is an abstract collection of operations that may be  
188 supported by one or more Web services. A WSDL portType defines these operations in terms of message  
189 definitions which usually rely on the XML Schema language to describe the representation of each message. A  
190 single WSDL file may contain multiple portType entities. Each portType is uniquely identified by the combination of  
191 its local name and the target namespace of the definitions element that contains the portType.

192 WSDL portTypes may be "implemented" by more than one Web service. Web services that purport to support a  
193 given portType must adhere not only to the message formats that are part of the WSDL definition, they must also  
194 adhere to the semantic agreement that is implicitly part of the portType. This allows applications to treat two web  
195 services as substitutable if and only if they implement a common portType.

#### 196 2.1.2 binding

197 WSDL portTypes are defined independently from the transport protocol used to transmit the messages. To allow  
198 transport protocol details to be expressed in WSDL, one must define a second construct, known as a binding. A  
199 WSDL binding adds transport specifics to a particular WSDL portType. Again, a single WSDL file may contain  
200 multiple bindings. A WSDL binding specifies its portType through a QName reference. The portType  
201 implemented by a binding may or may not be in the same target namespace as the binding itself. Like a portType,  
202 a binding is uniquely identified by the combination of its local name and the target namespace of the definitions  
203 element that contains the binding.

#### 204 2.1.3 service and port

205 Finally, WSDL defines a service as a collection of named ports, each of which is bound to a particular portType  
206 through a named binding. A service may expose multiple ports in order to make a single portType available over  
207 multiple protocols. Or, a service may expose multiple ports in order to expose more than one portType from a  
208 single logical entity. A WSDL port specifies the binding it implements through a QName reference.

#### 209 2.1.4 import

210 The import directive in WSDL allows the separation of these different entities into multiple files. As such, a WSDL  
211 file may be composed of a single portType, multiple portTypes, a single binding that imports its portType definition,  
212 multiple bindings, a single service or multiple services, etc. The WSDL data model provides great flexibility in  
213 terms of composition and reusability of WSDL entities.

214 Given this, the critical components of a WSDL file in terms of composition and identity are the target namespace of  
215 the definitions element and the local names which identify each portType, binding, service and port within the target  
216 namespace.

Deleted: wsdl-TN-V2.00-Draft-  
20021021

217 **2.2 UDDI Data Model**

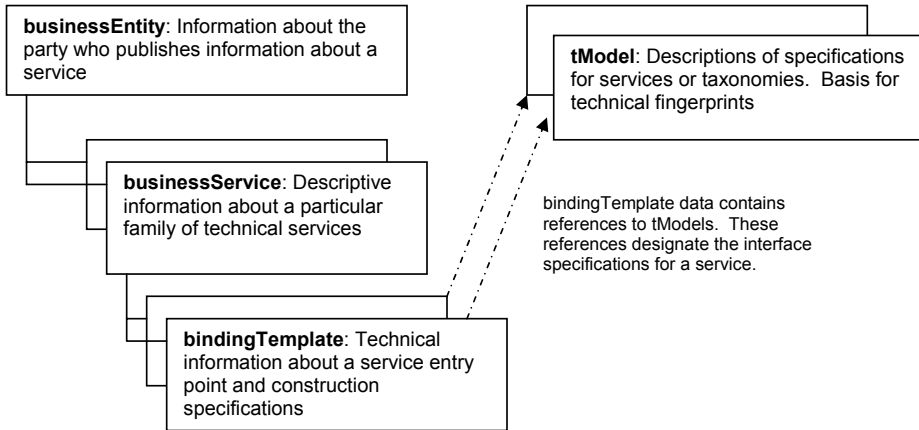
218 As an aid to understanding the sections ahead, we provide here a brief overview of two UDDI data structures that  
 219 are particularly relevant to the use of WSDL in the context of a UDDI registry: the tModel, also known as the service  
 220 type definition, and the businessService.

221 **2.2.1 tModels**

222 TModels represent unique concepts or constructs and provide the ability to describe compliance with a  
 223 specification, a concept, or a shared design. TModels have various uses in the UDDI registry. In the case of  
 224 mapping WSDL-described Web services, tModels are used to represent technical specifications like wire protocols,  
 225 interchange formats and sequencing rules. When a particular specification is registered with the UDDI repository as  
 226 a tModel, it is assigned a unique key, which is then used in the description of service instances to indicate  
 227 compliance with the specification.

228 Each tModel contains an overviewURL which provides an address where the specification itself can be retrieved,  
 229 for example, a WSDL file.

230 TModels can be decorated with searchable metadata by associating identifierBags and categoryBags with that  
 231 tModel. These bags contain keyedReferences: name/value pairs associated with a unique identifier. This  
 232 metadata can be used to search for these tModels through UDDI Inquiry API calls.



233 **2.2.2 businessService & bindingTemplate**

234 Services themselves are represented in UDDI by the businessService data structure, and the details of how and  
 235 where the service is accessed are provided by one or more bindingTemplate structures. The businessService  
 236 might be thought of as a logical container of services. The bindingTemplate structure contains the accessPoint of  
 237 the service itself, as well as references to the tModels it is said to implement.

238 **2.3 Mapping WDSL and UDDI**

239 To represent information about a WSDL file (such as target namespace, QName relationships and resource  
 240 locations) so that it is consistently described and discovered requires a mapping practice that acknowledges the  
 241 variety and flexibility within the WSDL model itself. This can be accomplished by a comprehensive use of UDDI's  
 242 key entity structures (tModels) and typed meta-data construct (keyedReferences).

243 By representing WSDL portType and binding entities as UDDI tModels, one can accurately represent the building  
 244 blocks of "abstract" WSDL interface information in UDDI. Similarly, by representing WSDL service and port as,  
 245 respectively, UDDI businessService and bindingTemplate, one achieves a similar parallel. Moreover, by  
 246 representing the target namespace and dependencies of a WSDL entity as keyedReferences on the UDDI entities,  
 247 one can represent strongly typed metadata about the WSDL entity within UDDI.

248 There are two important things to note about this mapping, especially as compared to the Version 1 *Using WSDL in*  
 249 *a UDDI Registry Best Practice*.

[wsdl-TN-V2.00-Draft-20021022](#)

**Deleted:** wsdl-TN-V2.00-Draft-20021021



250 First, this new approach means that a single WSDL file may be mapped to multiple tModels. For example, a WSDL  
 251 file located at <http://www.example.com/example.wsdl> which contains a single portType definition and 2 binding  
 252 definitions will map to three distinct tModels in UDDI. This decision differs significantly from the Version 1 Best  
 253 Practice, which always mapped the entirety of a WSDL file to a single tModel. The rationale for this new mapping  
 254 decision is allows for the modularity of WSDL to be expressed through UDDI. By decomposing WSDL into multiple  
 255 tModels, one can accurately model in UDDI exactly which portTypes and bindings a given Web service supports,  
 256 as opposed to being constrained to asserting that a Web service always supports the entirety of the WSDL file,  
 257 which may not be the case.

258 While there is an increased amount of data from a WSDL file modeled in UDDI, this new approach is in accord with  
 259 the original Best Practice in that it does not attempt to use UDDI as a repository for *all* of the data in a WSDL file.  
 260 Just as in the Version 1 Best Practice, one still must go outside of the UDDI registry to retrieve the portType and  
 261 binding information necessary for software applications to work with that Web service.

## 262 2.4 References to WSDL Components

263 As part of mapping WSDL in UDDI it is necessary to refer to various components in the WSDL document(s).  
 264 These references occur as overviewURL values. As noted above, in this mapping several tModels can refer to the  
 265 same WSDL file. The particular WSDL component SHOULD be determined by using the metadata contained  
 266 within the tModel's categoryBag. Alternatively, the overviewURL value MAY contain a fragment identifier which  
 267 identifies the particular WSDL component, but this is not required. If the optional fragment identifier is used then  
 268 the syntax described in Appendix B MUST be used.

Formatted: Heading 2,H2

## 269 2.5 WSDL Extensibility Elements

270 WSDL allows for extensibility elements to be included under many of the WSDL elements. The only extensibility  
 271 elements that are relevant to the UDDI mapping of a WSDL description are binding extensions, specifically the  
 272 extensibility element that can be added under the main wsdl:binding element and the extensibility element that can  
 273 be added under the wsdl:port element. The first of these is used to declare particular protocols and message  
 274 formats, the second is to provide address information.

275 These binding extensions affect only the bindingTemplate. The mappings defined below include details on the  
 276 SOAP 1.1 and HTTP GET/POST bindings defined in the WSDL 1.1 W3C Note and describe how other bindings  
 277 should be incorporated into the UDDI mapping.

Formatted: Bullets and  
Numbering

## 278 2.6 Mapping WSDL 1.1 in UDDI V2

### 279 2.6.1 wsdl:portType → uddi:tModel

280 A wsdl:portType MUST be modeled as a uddi:tModel.

281 The uddi:name element of the tModel MUST be the value of the name attribute of the portType.

282 The tModel MUST contain a categoryBag and the categoryBag MUST contain at least two keyedReferences:

- 283 1. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the  
 284 target namespace of the definitions element that contains the portType.<sup>1</sup>
- 285 2. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of  
 286 "portType".

287 The tModel MUST contain an overviewDoc with an overviewURL containing the location of the WSDL file.

Formatted: Bullets and  
Numbering

### 288 2.6.2 wsdl:binding → uddi:tModel

289 A wsdl:binding MUST be modeled as a uddi:tModel.

290 The uddi:name element of the tModel MUST be the value of the name attribute of the binding.

291 The tModel MUST contain a categoryBag and the categoryBag MUST contain at least three keyedReferences:

<sup>1</sup> WSDL 1.1 does not require the usage of a targetNamespace, but such a practice is not recommended. In the event that a WSDL file without a targetNamespace is registered in UDDI, it will not have a WSDL Namespace keyedReference and queries for these tModels based solely on the tModel name could return multiple results because no namespace can be specified.

Deleted: wsdl-TN-V2.00-Draft-  
20021021

- 292 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of "binding".
- 293 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the
- 294 target namespace of the definitions element that contains the binding.
- 295 3. A keyedReference with a tModelKey of the portType tModel and a keyValue of the tModelKey that
- 296 models the portType to which the binding relates.

297 The tModel categoryBag MAY contain an additional keyedReference for the uddi.org types taxonomy of wsdlSpec  
 298 for backward compatibility<sup>2</sup>.

299 The tModel MUST contain an overviewDoc with an overviewURL containing the location of the WSDL file.

Formatted: Bullets and  
Numbering

### 300 **2.6.3 wsdl:service → uddi:businessService**

301 A wsdl:service MUST be modeled as a uddi:businessService. An existing businessService MAY be used or a new  
 302 businessService MAY be created.

303 If a new businessService is created, the uddi:name of this businessService MUST be the value of the name  
 304 attribute of the wsdl:service<sup>3</sup>.

305 The bindingTemplates of the businessService MUST include bindingTemplates that model the ports of the service,  
 306 as described in the following sections.

307 The businessService MUST contain a categoryBag and the categoryBag MUST contain three keyedReferences:

- 308 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of "service".
- 309 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the
- 310 target namespace of the definitions element that contains the service.
- 311 3. A keyedReference with a tModelKey of the WSDL Local Name tModel and a keyValue that is the
- 312 value of the name attribute of the wsdl service.

Formatted: Bullets and  
Numbering

### 313 **2.6.4 wsdl:port and wsdl:binding → uddi:bindingTemplate**

314 A wsdl :port MUST be modeled as a uddi:bindingTemplate.

315 The bindingTemplate tModelInstanceDetails element MUST contain the following tModelInstanceInfo elements:

- 316 1. A tModelInstanceInfo with a tModelKey value of the key of the tModel that models the wsdl:binding
- 317 that this port implements. The instanceParms of this tModelInstanceInfo MUST contain the
- 318 wsdl:port local name.
- 319 2. A tModelInstanceInfo with a tModelKey value of the key of the tModel that models the portType<sup>4</sup>.
- 320 3. A tModelInstanceInfo with a tModelKey value of the key of the standard tModel that represents a
- 321 URL of a WSDL document. The instanceParms of this tModelInstanceInfo MUST contain the URL
- 322 of the WSDL file.

Formatted: Bullets and  
Numbering

Formatted: Heading 4,H4

#### 323 **2.6.4.1 soap:binding**

324 If the wsdl:binding contains a soap:binding then the bindingTemplate MUST include the following additional  
 325 tModelInstanceInfo element:

Deleted: s

- 326 1. A tModelInstanceInfo with a tModelKey value of the key of the standard tModel that represents the
- 327 SOAP extension to WSDL.

328 If the value of the transport attribute of the soap:binding element is http://schemas.xmlsoap.org/soap/http then the  
 329 bindingTemplate MUST include the following additional tModelInstanceInfo element:

Deleted: wsdl-TN-V2.00-Draft-  
20021021

<sup>2</sup> By categorizing a wsdl:binding tModel according to the Version 1 UDDI/WSDL Best Practice, backward compatibility is maintained. However, wsdl:portType tModels should not be categorized with this designation, as the wsdl:portType tModel will not contain sufficient information to compose a complete WSDL binding.

<sup>3</sup> Because an existing businessService could be used and it is unlikely in this case that the businessService name is the same as the WSDL service name, the keyedReference in the categoryBag MUST be used to retrieve the WSDL service name, not the businessServiceName.

<sup>4</sup> While it may seem redundant to create tModelInstanceInfos for both portType and-binding, doing so allows one to query for services based on known portTypes, as opposed to just known bindings.

330 | 1. A tModelInstanceInfo with a tModelKey value of the key of the standard tModel that represents the  
 331 | HTTP transport for SOAP.

Formatted: Bullets and Numbering

332 | If the value of the transport attribute is anything else then the bindingTemplate MUST include an additional  
 333 | tModelInstanceInfo element with a tModelKey value of the key of an appropriate transport tModel.

Formatted: Heading 4,H4

#### 334 | **2.6.4.2 http:binding**

335 | If the wsdl:binding contains an http:binding then the bindingTemplate MUST include an additional  
 336 | tModelInstanceInfo element with a tModelKey value of the key of the standard tModel that represents the HTTP  
 337 | protocol. Note that this is a different tModel from the HTTP transport tModel and in this case there is no separate  
 338 | transport tModel.

339 | The instanceParms contain the the value of the verb attribute of the http:binding, usually either "GET" or "POST"  
 340 | but other values are possible.

Formatted: Heading 4,H4

#### 341 | **2.6.4.3 Other Bindings**

342 | Other binding extensibility elements are handled in a similar fashion. It is assumed that vendors that provide other  
 343 | bindings will provide the appropriate protocol and transport tModels.

Formatted: Heading 4,H4

#### 344 | **2.6.4.4 soap:address → uddi:accessPoint**

Formatted: Bullets and Numbering

345 | A soap:address MUST be modeled as a uddi:accessPoint in the bindingTemplate that models the wsdl:port that  
 346 | contains the soap:address.

347 | The value of the URLType attribute of the accessPoint MUST correspond to the transport specified by the  
 348 | soap:binding, or "other" if no correspondence exists. In the case of the HTTP transport for example, the value of  
 349 | the URLType attribute MUST be "http".

350 | If "other" is used then a tModelInstanceInfo element referencing the appropriate vendor-defined transport tModel  
 351 | MUST have been added to the bindingTemplate as described above.

352 | The value of the accessPoint MUST be the value of the location attribute of the soap:address.

Formatted: Heading 4,H4

#### 353 | **2.6.4.5 http:address → uddi:accessPoint**

354 | An http:address MUST be mapped to a uddi:accessPoint. The value of the URLType attribute of the accessPoint  
 355 | MUST be "http".

356 | The value of the accessPoint MUST be the value of the location attribute of the http:address.

Formatted: Heading 4,H4

#### 357 | **2.6.4.6 Other Addresses**

358 | Any other address extensibility element MUST be modeled as a uddi:accessPoint. The value of the URLType  
 359 | attribute SHOULD be set to "other" if none of the other defined values of the attribute are appropriate.

360 | If the address information cannot be mapped to the value of the accessPoint, then the value of the accessPoint  
 361 | MUST be empty and the original WSDL description must be retrieved to obtain the address information.

Formatted: Bullets and Numbering

## 362 | **2.7 Mapping WSDL 1.1 in UDDI V3**

363 | The differences in UDDI data structures that are relevant are:

- 364 | 1. A bindingTemplate can have a categoryBag.
- 365 | 2. An accessPoint has a useType attribute not a URLType attribute and one of the values of useType  
 366 | is wsdlDeployment which indicates that the address information should be retrieved from the  
 367 | referenced WSDL document.
- 368 | 3. An overviewURL now has an optional useType attribute and a standard value of "wsdlInterface"  
 369 | has been defined to indicate "an abstract interface document". This mapping assumes that  
 370 | "wsdlInterface" is used with tModels that represent both portTypes and bindings.

Formatted: Bullets and Numbering

### 371 | **2.7.1 wsdl:portType → uddi:tModel**

Deleted: wsdl-TN-V2.00-Draft-20021021

372 | A wsdl:portType MUST be modeled as a uddi:tModel.

[wsdl-TN-V2.00-Draft-20021022](#)

373 The name of the tModel MUST be the value of the name attribute of the portType.

374 The tModel MUST contain a categoryBag and the categoryBag MUST contain at least the following  
375 keyedReferences:

- 376 1. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the  
377 target namespace of the definitions element that contains the portType.
- 378 2. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of  
379 "portType".

380 The tModel MUST contain an overviewDoc with an overviewURL containing the location of the WSDL file. The  
381 value of the useType attribute of the overviewURL MUST be "wsdlInterface".

382 **2.7.2 wsdl:binding → uddi:tModel**

Formatted: Bullets and Numbering

383 A wsdl:binding MUST be modeled as a uddi:tModel.

384 The name of the tModel MUST be the value of the name attribute of the binding.

385 The tModel MUST contain a categoryBag and the categoryBag MUST contain at least the following  
386 keyedReferences:

- 387 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of "binding".
- 388 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the  
389 target namespace of the definitions element that contains the binding.
- 390 3. A keyedReference with a tModelKey of the portType tModel and a keyValue of the tModel that  
391 models the portType to which the binding relates.

392 The tModel categoryBag MAY contain an additional keyedReference for the uddi.org types taxonomy of wsdlSpec  
393 for backward compatibility.

394 The tModel MUST contain an overviewDoc with an overviewURL containing the location of the WSDL file. The  
395 value of the useType attribute of the overviewURL MUST be "wsdlInterface".

Formatted: Bullets and Numbering

396 **2.7.3 wsdl:service → uddi:businessService**

397 A wsdl:service MUST be modeled as a uddi:businessService. An existing businessService MAY be used or a new  
398 businessService MAY be created.

399 If a new businessService is created, the uddi:name of this businessService MAY be the value of the name attribute  
400 of the wsdl:service.

401 The bindingTemplates of the businessService MUST include bindingTemplates that model the ports of the service,  
402 as described in the following sections.

403 The businessService MUST contain a categoryBag and the categoryBag MUST contain the following  
404 keyedReferences:

- 405 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of "service".
- 406 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the  
407 target namespace.
- 408 3. A keyedReference with a tModelKey of the WSDL Local Name tModel and a keyValue that is the  
409 value of the name attribute of the wsdl service.

Formatted: Bullets and Numbering

410 **2.7.4 wsdl:port and wsdl:binding → uddi:bindingTemplate**

411 A wsdl:port MUST be modeled as a uddi:bindingTemplate.

412 The bindingTemplate tModelInstanceDetails element MUST contain the following tModelInstanceInfo elements:

- 413 1. A tModelInstanceInfo with a tModelKey value of the key of the tModel that models the wsdl:binding  
414 that this port implements.
- 415 2. A tModelInstanceInfo with a tModelKey value of the key of the tModel that models the portType.

416 The bindingTemplate MUST contain a categoryBag and the categoryBag MUST contain the following  
417 keyedReferences:

Deleted: wsdl-TN-V2.00-Draft-20021021

- 418 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of "port".  
wsdl-TN-V2.00-Draft-20021022

Formatted: Bullets and Numbering

- 419 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the target  
 420 namespace of the definitions element that contains the port.  
 421 3. A keyedReference with a tModelKey of the WSDL Local Name tModel and a keyValue of the local  
 422 name of the port.<sup>5</sup>

Formatted: Heading 4,H4

423 **soap:binding**

424 If the wsdl:binding contains a soap:binding then the bindingTemplate MUST include the following additional  
 425 tModelInstanceInfo element:

Deleted: s

- 426 1. A tModelInstanceInfo with a tModelKey value of the key of the standard tModel that represents the  
 427 SOAP binding.

428 If the value of the transport attribute of the soap:binding element is http://schemas.xmlsoap.org/soap/http then the  
 429 bindingTemplate MUST include the following additional tModelInstanceInfo element:

- 430 1. A tModelInstanceInfo with a tModelKey value of the key of the standard tModel that represents the  
 431 HTTP transport for SOAP.

Formatted: Bullets and Numbering

432 If the value of the transport attribute is anything else then the bindingTemplate MUST include an additional  
 433 tModelInstanceInfo element with a tModelKey value of the key of an appropriate transport tModel.

Deleted: Other binding extensibility elements are handled in a similar fashion

434 **2.7.4.1 http:binding**

Formatted: Heading 4,H4

435 If the wsdl:binding contains an http:binding then the bindingTemplate MUST include an additional  
 436 tModelInstanceInfo element with a tModelKey value of the key of the standard tModel that represents the HTTP  
 437 protocol. Note that this is a different tModel from the HTTP transport tModel and in this case there is no separate  
 438 transport tModel.

439 The instanceParms element contains the value of the verb attribute of the http:binding, usually either "GET" or  
 440 "POST" but other values are possible.

Formatted: Heading 4,H4

441 **2.7.4.2 Other Bindings**

442 Other binding extensibility elements are handled in a similar fashion. It is assumed that vendors that provide other  
 443 bindings will provide the appropriate protocol and transport tModels.

444 **2.7.4.3 soap:address → uddi:accessPoint**

Deleted: The bindingTemplate MUST contain a categoryBag and the categoryBag MUST contain the following keyedReferences:¶  
 <#>A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of "port".¶  
 <#>A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the target namespace of the definitions element that contains the port.¶  
 <#>A keyedReference with a tModelKey of the WSDL Local Name tModel and a keyValue of the local name of the port.<sup>5</sup>¶

445 A soap:address MUST be modeled as a uddi:accessPoint in the bindingTemplate that models the wsdl:port that  
 446 contains the soap:address.

447 The useType of the accessPoint SHOULD be set to endPoint and the content of the accessPoint SHOULD be the  
 448 value of the location attribute of the soap:address. Alternatively, the useType of the accessPoint MAY be set to  
 449 wsdlDeployment and the content of the accessPoint SHOULD be a reference to the WSDL file containing the  
 450 address information.

451 **2.7.4.4 http:address → uddi:accessPoint**

452 An http:address MUST be mapped to a uddi:accessPoint in the bindingTemplate that models the wsdl:port that  
 453 contains the http:address.

Formatted: Heading 4,H4

454 The useType of the accessPoint SHOULD be set to endPoint and the content of the accessPoint SHOULD be the  
 455 value of the location attribute of the soap:address. Alternatively, the useType of the accessPoint MAY be set to  
 456 wsdlDeployment and the content of the accessPoint SHOULD be a reference to the WSDL file containing the  
 457 address information.

Formatted: Bullets and Numbering

Formatted: Heading 4,H4

458 **2.7.4.5 Other Addresses**

Formatted: Heading 4,H4

459 Any other address extensibility element MUST be mapped to a uddi:accessPoint in the bindingTemplate that  
 460 models the wsdl:port that contains the address.

<sup>5</sup> Note that in the v3 modeling, the local-name of the port is captured in a keyedReference, whereas in v2, it was captured in the tModelInstanceInfo instanceParms.

Deleted: wsdl-TN-V2.00-Draft-20021021

461 [The useType of the accessPoint SHOULD be set to endPoint and the content of the accessPoint SHOULD be the](#)  
 462 [value of the location attribute of the soap:address. Alternatively, the useType of the accessPoint MAY be set to](#)  
 463 [wsdlDeployment and the content of the accessPoint SHOULD be a reference to the WSDL file containing the](#)  
 464 [address information.](#)

## 465 3 A Complete Example

466 Consider the following sample based on the WSDL file presented in the WSDL 1.1 specification. This sample  
 467 shows how a single WSDL file is decomposed into two tModels (one for the portType and one for the binding) and  
 468 one businessService with one bindingTemplate. It then shows the kinds of UDDI API queries that can be used for  
 469 the purposes of discovery.

### 470 3.1 WSDL Sample

```

471     <?xml version="1.0" encoding="utf-8" ?>
472     <definitions
473         name="StockQuote"
474         targetNamespace="http://example.com/stockquote/"
475         xmlns:tns="http://example.com/stockquote/"
476         xmlns:xsd="http://example.com/stockquote/schema/"
477         xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
478         xmlns="http://schemas.xmlsoap.org/wsdl/">
479         <import
480             namespace="http://example.com/stockquote/schema/"
481             location="http://location/schema.xsd" />
482         <message name="GetLastTradePriceInput">
483             <part name="body" element="xsd:TradePriceRequest" />
484         </message>
485         <message name="GetLastTradePriceOutput">
486             <part name="body" element="xsd:TradePrice" />
487         </message>
488         <portType name="StockQuotePortType">
489             <operation name="GetLastTradePrice">
490                 <input message="tns:GetLastTradePriceInput" />
491                 <output message="tns:GetLastTradePriceOutput" />
492             </operation>
493         </portType>
494         <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
495             <soap:binding style="document"
496                 transport="http://schemas.xmlsoap.org/soap/http" />
497             <operation name="GetLastTradePrice">
498                 <soap:operation soapAction="http://example.com/GetLastTradePrice" />
499                 <input>
500                     <soap:body use="literal" />
501                 </input>
502                 <output>
503                     <soap:body use="literal" />
504                 </output>
505             </operation>
506         </binding>
507
508         <service name="StockQuoteService">
509             <documentation>My first service</documentation>
510             <port name="StockQuotePort" binding="tns:StockQuoteBinding">
511                 <soap:address location="http://location/sample"/>
512             </port>
513         </service>
514
515     </definitions>

```

516 Note that this wsdl file has one portType, one binding, one service and one port. As such, this sample represents  
 517 the simplest wsdl file. Also note that the location of this WSDL is at <http://location/sample.wsdl>.

Deleted: wsdl-TN-V2.00-Draft-20021021

## 518 3.2 UDDI V2 Model

### 519 3.2.1 UDDI portType tModel

```

520     <tModel tModelKey="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3" >
521         <name>
522             StockQuotePortType
523         </name>
524         <categoryBag>
525             <keyedReference
526                 tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
527                 keyValue="http://example.com/stockquote/"
528             />
529             <keyedReference
530                 tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
531                 keyValue="portType"
532             />
533         </categoryBag>
534         <overviewDoc>
535             <overviewURL>
536                 http://location/sample.wsdl
537             </overviewURL>
538         </overviewDoc>
539     </tModel>

```

### 540 3.2.2 UDDI binding tModel

```

541     <tModel tModelKey="uuid:49662926-f4a5-4ba5-b8d0-32ab388dadda">
542         <name>
543             StockQuoteSoapBinding
544         </name>
545         <categoryBag>
546             <keyedReference
547                 tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
548                 keyValue="http://example.com/stockquote/"
549             />
550             <keyedReference
551                 tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
552                 keyValue="binding"
553             />
554             <keyedReference
555                 tModelKey="uuid:acc0db62-2866-4081-8986-d62a7bf75ab3"
556                 keyValue="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3"
557             />
558             <keyedReference
559                 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
560                 keyValue="wsdlSpec"
561             />
562         </categoryBag>
563         <overviewDoc>
564             <overviewURL>
565                 http://location/sample.wsdl
566             </overviewURL>
567         </overviewDoc>
568     </tModel>

```

569 Note how this tModel creates a link to the portType tModel through placing the tModelKey of the portType tModel  
570 above as its keyValue.

### 571 3.2.3 UDDI businessService and bindingTemplate

```

572     <businessService
573         serviceKey="102b114a-52e0-4af4-a292-02700da543d4"

```

Deleted: wsdl-TN-V2.00-Draft-20021021



```

574         businessKey="1e65ea29-4e0f-4807-8098-d352d7b10368">
575     <name>StockQuoteService</name>
576     <bindingTemplates>
577         <bindingTemplate
578             bindingKey="f793c521-0daf-434c-8700-0e32da232e74"
579             serviceKey="102b114a-52e0-4af4-a292-02700da543d4">
580             <accessPoint URLType="http">
581                 http://location/sample
582             </accessPoint>
583             <tModelInstanceDetails>
584                 <tModelInstanceInfo
585                     tModelKey="uuid:454ef824-994f-4ce3-a892-3d97ad952707">
586                     <description xml:lang="en">
587                         This tModel represents the SOAP binding
588                         extensibility element.
589                     </description>
590                 <tModelInstanceInfo
591                     tModelKey="uuid:26de470c-0dab-4471-89f9-57450a524005">
592                     <description xml:lang="en">
593                         This tModel represents the HTTP transport using SOAP.
594                     </description>
595                 </tModelInstanceInfo>
596                 <tModelInstanceInfo
597                     tModelKey="uuid:49662926-f4a5-4ba5-b8d0-32ab388dadda">
598                     <description xml:lang="en">
599                         This tModel key represents the wsdl:binding that this
600                         wsdl:port implements.
601                     </description>
602                     <instanceDetails>
603                         <instanceParms>StockQuotePort</instanceParms>
604                     </instanceDetails>
605                 </tModelInstanceInfo>
606                 <tModelInstanceInfo
607                     tModelKey="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3">
608                     <description xml:lang="en">
609                         This tModel represents the wsdl:portType that
610                         this wsdl:port implements.
611                     </description>
612                 </tModelInstanceInfo>
613                 <tModelInstanceInfo
614                     tModelKey="uuid: acc1db62-2866-4081-8986-d62a7bf75ab4">
615                     <description xml:lang="en">
616                         The location of the WSDL file containing the port.
617                     </description>
618                     <instanceDetails>
619                         <instanceParms>http://location/sample.wsdl</instanceParms>
620                     </instanceDetails>
621                 </tModelInstanceInfo>
622             </tModelInstanceDetails>
623         </bindingTemplate>
624     </bindingTemplates>
625     <categoryBag>
626         <keyedReference
627             tModelKey="uuid:7950cc8d-259a-4a9a-8f13-e51632fe12d5"
628             keyValue="service"
629         />
630         <keyedReference
631             tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
632             keyValue="http://example.com/stockquote/"
633         />
634         <keyedReference
635             tModelKey="uuid:248ccb56-6365-446d-847a-91171ae79740"
636             keyValue="StockQuoteService"
637         />
638     </categoryBag>

```

Deleted: wsdl-TN-V2.00-Draft-20021021



```
639         </categoryBag>
640     </businessService>
```

### 641 3.3 Sample V2 Queries

642 This section shows how to perform various UDDI V2 queries given the model of the example.

#### 643 3.3.1 Find tModel for portType name

644 Find tModel for the portType StockQuotePortType in the namespace http://example.com/stockquote/

```
645     <find_tModel generic="2.0" xmlns="urn:uddi-org:api_v2">
646         <name>StockQuotePortType</name>
647         <categoryBag>
648             <keyedReference tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
649             keyValue="portType"/>
650             <keyedReference tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
651             keyValue="http://example.com/stockquote/" />
652         </categoryBag>
653     </find_tModel>
```

654 This should return the tModelKey uuid:e8cf1163-8234-4b35-865f-94a7322e40c3.

#### 655 3.3.2 Find bindings for portType

```
656     <find_tModel generic="2.0" xmlns="urn:uddi-org:api_v2">
657         <categoryBag>
658             <keyedReference tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
659             keyValue="binding"/>
660             <keyedReference tModelKey="uuid:acc0db62-2866-4081-8986-d62a7bf75ab3"
661             keyValue="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3"/>
662         </categoryBag>
663     </find_tModel>
```

664 This should return the tModelKey uuid:49662926-f4a5-4ba5-b8d0-32ab388dadda.

#### 665 3.3.3 Find Implementations of portType

```
666     <find_binding generic="2.0" xmlns="urn:uddi-org:api_v2">
667         <tModelBag>
668             <tModelKey>uuid:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
669         </tModelBag>
670     </find_binding>
```

671 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

#### 672 3.3.4 Find implementations of binding

```
673     <find_binding generic="2.0" xmlns="urn:uddi-org:api_v2">
674         <tModelBag>
675             <tModelKey>uuid:49662926-f4a5-4ba5-b8d0-32ab388dadda</tModelKey>
676         </tModelBag>
677     </find_binding>
```

678 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

#### 679 3.3.5 Find SOAP Implementations of portType

```
680     <find_binding generic="2.0" xmlns="urn:uddi-org:api_v2">
681         <tModelBag>
682             <tModelKey>uuid:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
683             <tModelKey>uuid:454ef824-994f-4ce3-a892-3d97ad952707</tModelKey>
684         </tModelBag>
685     </find_binding>
```

Formatted: Bullets and Numbering

Formatted: Body Text

Deleted: wsdl-TN-V2.00-Draft-20021021

686 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

### 687 3.3.6 Find SOAP/HTTP Implementations of portType

```
688     <find_binding generic="2.0" xmlns="urn:uddi-org:api_v2">
689         <tModelBag>
690             <tModelKey>uuid:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
691             <tModelKey>uuid:454ef824-994f-4ce3-a892-3d97ad952707</tModelKey>
692             <tModelKey>uuid:26de470c-0dab-4471-89f9-57450a524005</tModelKey>
693         </tModelBag>
694     </find_binding>
```

695 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

### 696 3.3.7 Find the portType of a binding

697 This is simply the keyValue of the keyedReference in the categoryBag with tModelKey="uuid:acc0db62-2866-4081-8986-d62a7bf75ab3" and therefore no query is required once the tModel of the binding is obtained.

## 699 3.4 UDDI V3 Model

### 700 3.4.1 UDDI portType tModel

```
701     <tModel tModelKey="uddi:e8cf1163-8234-4b35-865f-94a7322e40c3" >
702         <name>
703             StockQuotePortType
704         </name>
705         <categoryBag>
706             <keyedReference
707                 tModelKey="uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
708                 keyValue="http://example.com/stockquote/"
709             />
710             <keyedReference
711                 tModelKey="uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
712                 keyValue="portType"
713             />
714         </categoryBag>
715         <overviewDoc>
716             <overviewURL useType="wsdlInterface">
717                 http://location/sample.wsdl
718             </overviewURL>
719         </overviewDoc>
720     </tModel>
```

### 721 3.4.2 UDDI binding tModel

```
722     <tModel tModelKey="uddi:49662926-f4a5-4ba5-b8d0-32ab388dadda">
723         <name>
724             StockQuoteBinding
725         </name>
726         <categoryBag>
727             <keyedReference
728                 tModelKey="uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
729                 keyValue="http://example.com/stockquote/"
730             />
731             <keyedReference
732                 tModelKey="uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
733                 keyValue="binding"
734             />
735             <keyedReference
736                 tModelKey="uddi:acc0db62-2866-4081-8986-d62a7bf75ab3"
737                 keyValue="uddi:e8cf1163-8234-4b35-865f-94a7322e40c3"
738             />
```

Deleted: wsdl-TN-V2.00-Draft-20021021

```

739         <keyedReference
740             tModelKey="uddi:clacf26d-9672-4404-9d70-39b756e62ab4"
741             keyValue="wsdlSpec"
742         />
743     </categoryBag>
744     <overviewDoc>
745         <overviewURL useType="wsdlInterface">
746             http://location/sample.wsdl
747         </overviewURL>
748     </overviewDoc>
749 </tModel>

```

750 Note how this tModel creates a link to the portType tModel through placing the tModelKey of the portType tModel  
751 above as its keyValue.

### 752 3.4.3 UDDI businessService and bindingTemplate

```

753 <businessService
754     serviceKey="uddi:102b114a-52e0-4af4-a292-02700da543d4"
755     businessKey="uddi:1e65ea29-4e0f-4807-8098-d352d7b10368">
756     <name>StockQuoteService</name>
757     <bindingTemplates>
758         <bindingTemplate
759             bindingKey="uddi:f793c521-0daf-434c-8700-0e32da232e74"
760             serviceKey="uddi:102b114a-52e0-4af4-a292-02700da543d4">
761             <accessPoint useType="endPoint">
762                 http://location/sample
763             </accessPoint>
764             <tModelInstanceDetails>
765                 <tModelInstanceInfo
766                     tModelKey="uddi:454ef824-994f-4ce3-a892-3d97ad952707">
767                     <description xml:lang="en">
768                         This tModel represents the SOAP binding
769                         extensibility element.
770                     </description>
771                 </tModelInstanceInfo>
772                 <tModelInstanceInfo
773                     tModelKey="uddi:26de470c-0dab-4471-89f9-57450a524005">
774                     <description xml:lang="en">
775                         This tModel represents the HTTP transport using SOAP.
776                     </description>
777                 </tModelInstanceInfo>
778                 <tModelInstanceInfo
779                     tModelKey="uddi:49662926-f4a5-4ba5-b8d0-32ab388dadda">
780                     <description xml:lang="en">
781                         This tModel key represents the wsdl:binding that this
782                         wsdl:port implements.
783                     </description>
784                 </tModelInstanceInfo>
785                 <tModelInstanceInfo
786                     tModelKey="uddi:e8cf1163-8234-4b35-865f-94a7322e40c3">
787                     <description xml:lang="en">
788                         This tModel represents the wsdl:portType that
789                         this wsdl:port implements.
790                     </description>
791                 </tModelInstanceInfo>
792             </tModelInstanceDetails>
793             <categoryBag>
794                 <keyedReference
795                     tModelKey="uddi:7950cc8d-259a-4a9a-8f13-e51632fe12d5"
796                     keyValue="port"
797                 />
798                 <keyedReference
799                     tModelKey="uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
800                     keyValue="http://example.com/stockquote/"
801                 />

```

Deleted: wsdl-TN-V2.00-Draft-20021021

```

801         <keyedReference
802             tModelKey="uddi:248ccb56-6365-446d-847a-91171ae79740"
803             keyValue="StockQuotePort"
804         />
805     </categoryBag>
806 </bindingTemplate>
807 </bindingTemplates>
808 <categoryBag>
809     <keyedReference
810         tModelKey="uddi:7950ccb8d-259a-4a9a-8f13-e51632fe12d5"
811         keyValue="service"
812     />
813     <keyedReference
814         tModelKey="uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
815         keyValue="http://example.com/stockquote/"
816     />
817     <keyedReference
818         tModelKey="uddi:248ccb56-6365-446d-847a-91171ae79740"
819         keyValue="StockQuoteService"
820     />
821 </categoryBag>
822 </businessService>

```

## 823 3.5 Sample V3 Queries

824 This section shows how to perform various UDDI V3 queries given the model of the example.

### 825 3.5.1 Find tModel for portType name

Formatted: Bullets and Numbering

826 Find tModel for the portType StockQuotePortType in the namespace http://example.com/stockquote/

Formatted: Body Text

```

827 <find_tModel xmlns="urn:uddi-org:api_v3">
828     <name>StockQuotePortType</name>
829     <categoryBag>
830         <keyedReference tModelKey="uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
831             keyValue="portType"/>
832         <keyedReference tModelKey="uddi:5165f13f-139c-4649-adf8-89f66e437f55"
833             keyValue="http://example.com/stockquote/" />
834     </categoryBag>
835 </find_tModel>

```

836 This should return the tModelKey uddi:e8cf1163-8234-4b35-865f-94a7322e40c3.

### 837 3.5.2 Find bindings for portType

```

838 <find_tModel xmlns="urn:uddi-org:api_v3">
839     <CategoryBag>
840         <keyedReference tModelKey="uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
841             keyValue="binding"/>
842         <keyedReference tModelKey="uddi:acc0db62-2866-4081-8986-d62a7bf75ab3"
843             keyValue="uddi:e8cf1163-8234-4b35-865f-94a7322e40c3"/>
844     </categoryBag>
845 </find_tModel>

```

846 This should return the tModelKey uddi:49662926-f4a5-4ba5-b8d0-32ab388dadda.

### 847 3.5.3 Find Implementations of portType

```

848 <find_binding xmlns="urn:uddi-org:api_v3">
849     <tModelBag>
850         <tModelKey>uddi:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
851     </tModelBag>
852 </find_binding>

```

Deleted: wsdl-TN-V2.00-Draft-20021021

853 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

[wsdl-TN-V2.00-Draft-20021022](#)

### 854 3.5.4 Find Implementations of binding

```

855     <find_binding xmlns="urn:uddi-org:api_v3">
856       <tModelBag>
857         <tModelKey>uddi:49662926-f4a5-4ba5-b8d0-32ab388dadda</tModelKey>
858       </tModelBag>
859     </find_binding>

```

860 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

### 861 3.5.5 Find SOAP Implementations of portType

```

862     <find_binding xmlns="urn:uddi-org:api_v3">
863       <tModelBag>
864         <tModelKey>uddi:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
865         <tModelKey>(key of tModel representing SOAP extensibility
866 element)</tModelKey>
867       </tModelBag>
868     </find_binding>

```

869 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

### 870 3.5.6 Find SOAP/HTTP Implementations of portType

```

871     <find_binding xmlns="urn:uddi-org:api_v3">
872       <tModelBag>
873         <tModelKey>uddi:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
874         <tModelKey>(key of tModel representing SOAP extensibility
875 element)</tModelKey>
876         <tModelKey>uddi:1be8abb8-86e3-4ee5-99ac-b7fffbbed8110</tModelKey>
877       </tModelBag>
878     </find_binding>

```

879 This should return the bindingKey uddi:f793c521-0daf-434c-8700-0e32da232e74.

### 880 3.5.7 Find the portType of a binding

881 This is simply the keyValue of the keyedReference in the categoryBag with tModelKey="uddi:acc0db62-2866-4081-  
882 8986-d62a7bf75ab3" and therefore no query is required once the tModel of the binding is obtained.

## 883 4 References

- 884 1. UDDI Version 2.0 Data Structure Reference, July 7, 2002. Available at [http://uddi.org/pubs/DataStructure-V2.03-](http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.pdf)  
885 [Published-20020719.pdf](http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.pdf).
- 886 2. Using WSDL in a UDDI Registry, May 21, 2002. Available at <http://uddi.org/pubs/wsdlbestpractices.pdf>
- 887 3. Web Services Description Language (WSDL) 1.1, March 15, 2000. Available at <http://www.w3.org/TR/wsdl>
- 888 4. XPointer [xpointer\(\) Scheme, W3C Working Draft 10 July 2002](http://www.w3.org/TR/2002/WD-xptr-xpointer-20020710/). Available at [http://www.w3.org/TR/2002/WD-](http://www.w3.org/TR/2002/WD-xptr-xpointer-20020710/)  
889 [xptr-xpointer-20020710/](http://www.w3.org/TR/2002/WD-xptr-xpointer-20020710/).

## 890 Appendix A: Canonical tModels

### 891 A.1 WSDL Entity Type tModel

#### 892 A.1.1 Design Goals

893 This tModel provides a typing system based on the WSDL entity which the UDDI entity represents.

#### 894 A.1.2 Definition

895 **Name:** uddi.org:wsdl:types

896 **Description:** WSDL Type Category System

[wsdl-TN-V2.00-Draft-20021022](http://www.w3.org/TR/2002/WD-xptr-xpointer-20020710/)

**Deleted:** XML Pointer Language (

**Deleted:** ) Version 1.0, January 8, 2001

**Deleted:** [WD-](#)

**Deleted:** wsdl-TN-V2.00-Draft-20021021

897 **V3 format key:** uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5  
 898 **V1,V2 format key:** uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5  
 899 **Categorization:** categorization  
 900 **Checked:** no

901 A.1.2.1 tModel Structure

```

902     <tModel tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5" >
903         <name>uddi.org:wSDL:types</name>
904         <overviewDoc>
905             <overviewURL>
906                 http://uddi.org/pubs/uddi_wSDL_technical_note_v2.htm#wSDLTypes
907             </overviewURL>
908         </overviewDoc>
909         <categoryBag>
910             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
911             keyValue="unchecked" />
912             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
913             keyValue="categorization" />
914         </categoryBag>
915     </tModel>
    
```

916 A.1.3 Values

917 While this is an unchecked taxonomy, there are only four values that should be used with this taxonomy:

ID	Description	UDDI Entity
portType	Represents a UDDI entity categorized as a wsdl:portType	tModel
binding	Represents a UDDI entity categorized as a wsdl:binding	tModel
service	Represents a UDDI entity categorized as a wsdl:service	businessService
port	Represents a UDDI entity categorized as a wsdl:port	bindingTemplate (v3 only)

918

919 A.1.4 Example of Use

920 A tModel representing a portType tModel would have a categoryBag representing its type:

```

921     <categoryBag>
922         <keyedReference tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
923         keyName="WSDL Entity type" keyValue="portType" />
924         ...
925     </categoryBag>
    
```

926 **A.2 WSDL Namespace tModel**

927 A.2.1 Design Goals

928 Because each WSDL entity has an associated target namespace, there needs to be a way to capture that target  
 929 namespace. The WSDL Namespace tModel serves this purpose. *More than one tModel might be categorized with*  
 930 *the same namespace tModel* – in fact, this mapping would be quite common, as many WSDL files use a common  
 931 target namespace for <wsdl:portType>, <wsdl:binding> and <wsdl:service> elements.

932 A.2.2 Definition

933 **Name:** uddi.org:wSDL:namespace

934 **Description:** A tModel used to represent WSDL namespaces

935 **V3 format key:** uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3

**Deleted:** wsdl-TN-V2.00-Draft-20021021

936 **V1,V2 format key:** uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3

937 **Categorization:** categorization

938 **Checked:** No

939 A.2.2.1 tModel Structure

```

940     <tModel tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3" >
941         <name>uddi.org:wSDL:namespace</name>
942         <overviewDoc>
943             <overviewURL>
944                 http://uddi.org/pubs/uddi_wSDL_technical_note_v2.htm#wSDLNamespace
945             </overviewURL>
946         </overviewDoc>
947         <categoryBag>
948             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
949             keyValue="unchecked" />
950             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
951             keyValue="categorization" />
952         </categoryBag>
953     </tModel>

```

954 A.2.3 Values

955 The values used in this taxonomy come directly from the different target namespaces being represented by  
956 instances of this tModel.

957 A.2.4 Example of Use

958 A namespace keyedReference would be as follows:

```

959     <categoryBag>
960         <keyedReference tModelKey=" uuid:c0ec5422-44e9-4774-b2f1-
961         5a89eca389f3" keyName="namespace" keyValue="urn:foo" />
962     ...
963 </categoryBag>

```

## 964 A.3 WSDL Local Name tModel

965 A.3.1 Design Goals

966 Because each WSDL entity has a name attribute which is key to identifying that WSDL entity, there needs to be a  
967 way to capture that attribute in the mapping. In the case of portType and binding, this is mapped to the tModel  
968 name element. However, in the case of the wsdl:service entity, the name element of the businessService is not  
969 appropriate and, in the case of wsdl:port, the bindingTemplate entity does not have a name element. The WSDL  
970 Local Name tModel serves this purpose of capturing that information in these two constructs.

971 A.3.2 Definition

972 **Name:** uddi.org:wSDL:localname

973 **Description:** A tModel used to represent WSDL local names

974 **V3 format key:** uddi:248ccb56-6365-446d-847a-91171ae79740

975 **V1,V2 format key:** uuid:248ccb56-6365-446d-847a-91171ae79740

976 **Categorization:** categorization

977 **Checked:** No

978 A.3.2.1 tModel Structure

```

979     <tModel tModelKey="uuid:248ccb56-6365-446d-847a-91171ae79740" >
980         <name>uddi.org:wSDL:localname</name>
981         <overviewDoc>
982             <overviewURL>
983                 http://uddi.org/pubs/uddi_wSDL_technical_note_v2.htm#wSDLLocalName
984             </overviewURL>
985         </overviewDoc>
986         <categoryBag>
987             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
988             keyValue="unchecked" />

```

Deleted: wsdl-TN-V2.00-Draft-20021021

```

989         <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
990         keyValue="categorization" />
991     </categoryBag>
992 </tModel>

```

### 993 A.3.3 Values

994 The values used in this taxonomy come directly from the different target namespaces being signified by this tModel.

### 995 A.3.4 Example of Use

996 A namespace keyedReference would be as follows:

```

997     <categoryBag>
998         <keyedReference tModelKey="uuid:248ccb56-6365-446d-847a-91171ae79740"
999         keyValue="StockQuoteService" />
1000     ...
1001 </categoryBag>

```

## 1002 A.4 WSDL portType Reference tModel

### 1003 A.4.1 Design Goals

1004 There is also a need to be able to relate a wsdl:binding tModel to the wsdl:portType that it implements. In order to  
 1005 accomplish this goal, a tModel needs to be established that allow arcs to be created between tModels. This is  
 1006 accomplished by placing the tModel key of a given tModel as the keyValue of a keyedReference. In order to create  
 1007 these arcs, a tModel is needed that represents the wsdl:portType.

### 1008 A.4.2 Definition

1009 **Name:** uddi.org:wsdl:portTypeReference

1010 **Description:**

1011 **V3 format key:** uddi:acc0db62-2866-4081-8986-d62a7bf75ab3

1012 **V1,V2 format key:** uuid:acc0db62-2866-4081-8986-d62a7bf75ab3

1013 **Categorization:**

#### 1014 A.4.2.1 tModel Structure

```

1015 <tModel tModelKey="uuid:acc0db62-2866-4081-8986-d62a7bf75ab3" >
1016     <name>uddi.org:wsdl:portTypeReference</name>
1017     <description xml:lang="en">This tModel is a namespace tModel that can
1018     be used to identify other tModels as portType tModels.</description>
1019     <overviewDoc>
1020         <overviewURL>
1021             http://uddi.org/pubs/uddi_wsdl_technical_note_v2.htm#portTypeReference
1022         </overviewURL>
1023     </overviewDoc>
1024     <categoryBag>
1025         <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
1026         keyValue="categorization" />
1027         <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
1028         keyValue="unchecked" />
1029     </categoryBag>
1030 </tModel>
1031

```

### 1032 A.4.3 Values

1033 The values used in this taxonomy come directly from the tModelKeys that represent wsdl:portType tModels.

### 1034 A.4.4 Example of Use

1035 One would add the following keyed reference to signify wsdl:binding that implemented a known portType:

```

1036     <categoryBag>
1037         <keyedReference
1038         tModelKey="uuid:acc0db62-2866-4081-8986-d62a7bf75ab3"
1039         keyName="wsdl:portType Reference"
1040         keyValue="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3"
1041     />

```

**Deleted:** wsdl-TN-V2.00-Draft-20021021



1042 ...  
1043 </categoryBag>

1044 Note that the keyValue is a GUID, which, if queried for using get\_tModelDetail, would return a tModel that  
1045 represented a given portType.

1046 **A.5 WSDL URL Reference tModel**

1047 **A.5.1 Design Goals**

1048 It may be necessary to retrieve the WSDL description of a port to retrieve address information where this cannot be  
1049 mapped to the value of the UDDI V2 accessPoint. To enable this, a V2 bindingTemplate includes a  
1050 tModelInstanceInfo element that references this tModel and the instanceParms contains the URL of the WSDL file.

1051 Although this tModelInstanceInfo is not necessary for UDDI V3, as the wsdlDeployment accessPoint can be used,  
1052 a V3 key is given for this tModel to ease migration to UDDI V3.

1053 **A.5.2 Definition**

1054 **Name:** uddi.org:wsdl:portURL

1055 **Description:**

1056 **V3 format key:** uddi:acc1db62-2866-4081-8986-d62a7bf75ab4

1057 **V1,V2 format key:** uuid:acc1db62-2866-4081-8986-d62a7bf75ab4

1058 **Categorization:** wsdlSpec

1059 **A.5.2.1 tModel Structure**

```
1060 <tModel tModelKey="uuid:acc1db62-2866-4081-8986-d62a7bf75ab4" >
1061 <name>uddi.org:wsdl:portURL</name>
1062 <description xml:lang="en">This tModel is used to include the URL for a
1063 WSDL document that includes a port.</description>
1064 <overviewDoc>
1065 <overviewURL>
1066 http://uddi.org/pubs/uddi_wsdl_technical_note_v2.htm#portURL
1067 </overviewURL>
1068 </overviewDoc>
1069 <categoryBag>
1070 <keyedReference tModelKey="uuid:clacf26d-9672-4404-9d70-39b756e62ab4"
1071 keyValue="wsdlSpec" />
1072 </categoryBag>
1073 </tModel>
```

1074 **A.5.3 Values**

1075 The values used in this taxonomy come directly from the locations of WSDL files that contain port definitions.

1076 **A.5.4 Example of Use**

1077 One would add the following tModelInstanceInfo to a bindingTemplate to provide the URL of the WSDL file that  
1078 contained the port element:

```
1079 <tModelInstanceInfo
1080 tModelKey="uuid: acc1db62-2866-4081-8986-d62a7bf75ab4">
1081 <description xml:lang="en">
1082 The URL of the WSDL file that contains the port.
1083 </description>
1084 <instanceDetails>
1085
1086 <instanceParms>http://example.com/stockquote/stockquoteservice.wsdl</instanc
1087 eParms>
1088 </instanceDetails>
1089 </tModelInstanceInfo>
```

1090 **A.6 SOAP Protocol tModel**

1091 **A.6.1 Design Goals**

1092 The WSDL specification outlines usage of the SOAP extension to WSDL. This tModel is used to signify that a  
1093 given bindingTemplate uses the SOAP extension.

wsdl-TN-V2.00-Draft-20021022

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Font: 10 pt

Formatted: Font: 10 pt, Font color: Black

Formatted: codeSample

Deleted: ¶

Formatted: Bullets and Numbering

Deleted: wsdl-TN-V2.00-Draft-20021021

**A.6.2** Definition

**Name:** <http://schemas.xmlsoap.org/wsdl/soap/>

**Description:** Used to represent the WSDL SOAP extension

**V3 format key:** [uddi:454ef824-994f-4ce3-a892-3d97ad952707](http://www.ietf.org/rfc/rfc4801.txt)

**V1,V2 format key:** [uuid:454ef824-994f-4ce3-a892-3d97ad952707](http://www.ietf.org/rfc/rfc4801.txt)

**Categorization:** protocol

Formatted: Bullets and Numbering

**A.6.2.1** tModel Structure

```

1101     <tModel tModelKey="uuid:454ef824-994f-4ce3-a892-3d97ad952707">
1102         <name>http://schemas.xmlsoap.org/wsdl/soap/</name>
1103         <overviewDoc>
1104             <overviewURL>
1105                 http://uddi.org/pubs/uddi_wsdl_technical_note_v2.htm#soap
1106             </overviewURL>
1107         </overviewDoc>
1108         <categoryBag>
1109             <keyedReference>
1110                 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
1111                 keyName="Protocol"
1112                 keyValue="protocol"
1113             </keyedReference>
1114         </categoryBag>
1115     </tModel>

```

Formatted: Bullets and Numbering

**A.6.3** Example of Use

In the case when a `uddi:bindingTemplate` implements a `wsdl:binding`, it is important to model the protocol used by that binding. In such a way, a query can be run: show all bindings that support a given protocol.

```

1119     <bindingTemplates>
1120         <bindingTemplate
1121             bindingKey="f793c521-0daf-434c-8700-0e32da232e74"
1122             serviceKey="102b114a-52e0-4af4-a292-02700da543d4">
1123             <accessPoint URLType="http">
1124                 http://location/sample
1125             </accessPoint>
1126             <tModelInstanceDetails>
1127                 <tModelInstanceInfo
1128                     tModelKey="uuid:454ef824-994f-4ce3-a892-3d97ad952707">
1129                     <description xml:lang="en">
1130                         This tModel key represents the soap protocol
1131                     </description>
1132                 </tModelInstanceInfo>
1133                 ... ..
1134             </tModelInstanceDetails>
1135         </bindingTemplate>
1136     </bindingTemplates>

```

Formatted: Bullets and Numbering

**A.7** HTTP Protocol tModel**A.7.1** Design Goals

The WSDL specification outlines usage of the HTTP binding extension for WSDL. This tModel is used to signify that a given bindingTemplate uses the HTTP protocol.

**A.7.2** Definition

**Name:** [uddi.org/wsdl:http](http://uddi.org/wsdl:http)

**Description:** Used to represent the WSDL HTTP binding extension

**V3 format key:** [uddi:454ef824-994f-4ce3-a892-3d97ad952708](http://www.ietf.org/rfc/rfc4801.txt)

**V1,V2 format key:** [uuid:454ef824-994f-4ce3-a892-3d97ad952708](http://www.ietf.org/rfc/rfc4801.txt)

**Categorization:** ~~protocol~~

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Deleted: wsdl-TN-V2.00-Draft-20021021

1147 **A.7.2.1 tModel Structure**

Formatted: Bullets and Numbering

```

1148 <tModel tModelKey="uuid:454ef824-994f-4ce3-a892-3d97ad952708">
1149 <name>uddi.org:wSDL:http</name>
1150 <overviewDoc>
1151 <overviewURL>
1152 http://uddi.org/pubs/uddi_wSDL_technical_note_v2.htm#http
1153 </overviewURL>
1154 </overviewDoc>
1155 <categoryBag>
1156 <keyedReference>
1157 tModelKey="uuid:clacf26d-9672-4404-9d70-39b756e62ab4"
1158 keyName="Protocol"
1159 keyValue="protocol"
1160 />
1161 </categoryBag>
1162 </tModel>
    
```

1163 **A.7.3 Example of Use**

Formatted: Bullets and Numbering

1164 In the case when a uddi:bindingTemplate implements a wsdl:binding, it is important to model the protocol used by  
 1165 that binding. In such a way, a query can be run: show all bindings that support a given protocol.

```

1166 <bindingTemplates>
1167 <bindingTemplate>
1168 bindingKey="f793c521-0daf-434c-8700-0e32da232e74"
1169 serviceKey="102b114a-52e0-4af4-a292-02700da543d4">
1170 <accessPoint URLType="http">
1171 http://location/sample
1172 </accessPoint>
1173 <tModelInstanceDetails>
1174 <tModelInstanceInfo>
1175 tModelKey="uuid:454ef824-994f-4ce3-a892-3d97ad952708">
1176 <description xml:lang="en">
1177 This tModel key represents the http protocol
1178 </description>
1179 </tModelInstanceInfo>
1180 ... ..
1181 </tModelInstanceDetails>
1182 </bindingTemplate>
1183 </bindingTemplates>
    
```

Formatted: codeSample

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Deleted: SOAP over

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

1184 **A.8 HTTP SOAP Transport tModel**

1185 **A.8.1 Design Goals**

1186 Other transport tModels may be defined in the future.

1187 **A.8.2 Definition**

1188 **Name:** http://schemas.xmlsoap.org/soap/http

1189 **Description:** Used to represent soap over http

1190 **V3 format key:** uddi:26de470c-0dab-4471-89f9-57450a524005

1191 **V1,V2 format key:** uuid:26de470c-0dab-4471-89f9-57450a524005

1192 **Categorization:** transport

1193 **A.8.2.1 tModel Structure**

Formatted: Bullets and Numbering

```

1194 <tModel tModelKey="uuid:26de470c-0dab-4471-89f9-57450a524005" >
1195 <name>http://schemas.xmlsoap.org/soap/http</name>
1196 <overviewDoc>
1197 <overviewURL>
1198 http://uddi.org/pubs/uddi_wSDL_technical_note_v2.htm#http
1199 </overviewURL>
1200 </overviewDoc>
1201 <categoryBag>
1202 <keyedReference tModelKey="uuid:clacf26d-9672-4404-9d70-39b756e62ab4"
    
```

Deleted: wsdl-TN-V2.00-Draft-20021021

```

1203         keyName="Wire/transport protocol"
1204         keyValue="transport" />
1205     </categoryBag>
1206 </tModel>
    
```

**A.8.3 Example of Use**

Formatted: Bullets and Numbering

In the case when a uddi:bindingTemplate implements a wsdl:binding, it is important to model the transport used by that binding. In such a way, a query can be run: show all bindings that support a given transport.

```

1210     <bindingTemplates>
1211     <bindingTemplate
1212         bindingKey="f793c521-0daf-434c-8700-0e32da232e74"
1213         serviceKey="102b114a-52e0-4af4-a292-02700da543d4">
1214     <accessPoint URLType="http">
1215         http://location/sample
1216     </accessPoint>
1217     <tModelInstanceDetails>
1218     <tModelInstanceInfo
1219         tModelKey="uuid:26de470c-0dab-4471-89f9-57450a524005">
1220         <description xml:lang="en">
1221             This tModel key represents the wsdl:binding transport
1222         </description>
1223     </tModelInstanceInfo>
1224     ... ..
1225     </tModelInstanceDetails>
1226     </bindingTemplate>
1227 </bindingTemplates>
    
```

**Appendix B: Using XPointer in overviewURL**

**B.1 XPointer Syntax**

As mentioned earlier, it is optional to have a fragment identifier in an overviewURL that identifies a WSDL document. It is RECOMMENDED that applications that use the overviewURL use the UDDI metadata, the values in keyedReferences etc., to precisely define the target WSDL component. If an application does not use the UDDI metadata then the fragment identifier provides a generic mechanism to achieve the same identification of a particular WSDL component.

As the WSDL 1.1 schema does not allow for id attributes on WSDL elements, we cannot simply use a fragment identifier of the form #foo.

If the optional fragment identifier is used, the syntax defined by XPointer [4] MUST be used for the fragment identifier. It should be noted that at the time of writing this Technical Note, XPointer is a set of Working Draft documents and is therefore subject to change.

Deleted: p

Considering Example 2 in the WSDL 1.1 W3C Note, the form of the reference to the portType named StockQuotePortType would be

```

1242 http://example.com/stockquote/stockquote.wsdl#xmlns\(wsdl=http://schemas.xmlsoap.org/wsdl/\)
1243 xpointer(/wsdl:definitions/wsdl:portType\[@name="StockQuotePortType"\])
    
```

Deleted: /

Deleted: ") whereas the full Xpointer syntax for the same reference, including the namespace, would be [http://example.com/stockquote/stockquote.wsdl#xmlns\(x=http://example.com/stockquote/definitions\)](http://example.com/stockquote/stockquote.wsdl#xmlns(x=http://example.com/stockquote/definitions)) xpointer(//x:definitions/portType[@name="StockQuotePortType"])

Deleted: wsdl-TN-V2.00-Draft-20021021

Copyright .....	2
<i>Table of Contents</i> .....	3
1 Introduction .....	6
1.1 Goals and Requirements .....	6
1.2 Relationship to Version 1 Best Practice.....	6
1.3 tModel Keys.....	7
2 Rationalizing Two Data Models: WSDL & UDDI.....	7
2.1 WSDL Data Model .....	7
2.1.1 portType.....	7
2.1.2 binding .....	7
2.1.3 service and port.....	7
2.1.4 import.....	7
2.2 UDDI Data Model.....	8
2.2.1 tModels .....	8
2.2.2 businessService & bindingTemplate.....	8
2.3 Mapping WDSL and UDDI.....	8
2.4 References to WSDL Components .....	9
2.5 Mapping WSDL 1.1 in UDDI V2.....	9
2.5.1 wsdl:portType → uddi:tModel .....	9
2.5.2 wsdl:binding → uddi:tModel.....	9
2.5.3 wsdl:service → uddi:businessService.....	10
2.5.4 wsdl:port → uddi:bindingTemplate .....	10
2.5.5 soap:address → uddi:accessPoint .....	10
2.6 Mapping WSDL 1.1 in UDDI V3.....	10
2.6.1 wsdl:portType → uddi:tModel .....	11
2.6.2 wsdl:binding → uddi:tModel.....	11
2.6.3 wsdl:service → uddi:businessService.....	11
2.6.4 wsdl:port → uddi:bindingTemplate .....	12
2.6.5 soap:address → uddi:accessPoint .....	12
3 A Complete Example .....	12
3.1 WSDL Sample .....	12
3.2 UDDI V2 Model .....	13
3.2.1 UDDI portType tModel .....	13
3.2.2 UDDI binding tModel.....	14
3.2.3 UDDI businessService and bindingTemplate.....	14
3.3 Sample V2 Queries .....	15
3.3.1 Find tModel for the portType StockQuotePortType in the namespace http://example.com/stockquote/ .....	15
3.3.2 Find bindings for portType .....	15
3.3.3 Find Implementations of portType .....	16
3.3.4 Find implementations of binding.....	16
3.3.5 Find SOAP Implementations of portType .....	16
3.3.6 Find SOAP/HTTP Implementations of portType .....	16
3.3.7 Find the portType of a binding .....	16

3.4	UDDI V3 Model .....	16
3.4.1	UDDI portType tModel .....	16
3.4.2	UDDI binding tModel .....	17
3.4.3	UDDI businessService and bindingTemplate .....	17
3.5	Sample V3 Queries .....	18
3.5.1	Find tModel for the portType StockQuotePortType in the namespace http://example.com/stockquote/ .....	18
3.5.2	Find bindings for portType .....	19
3.5.3	Find Implementations of portType .....	19
3.5.4	Find Implementations of binding .....	19
3.5.5	Find SOAP Implementations of portType .....	19
3.5.6	Find SOAP/HTTP Implementations of portType .....	19
3.5.7	Find the portType of a binding .....	20
4	References .....	20
		Appendix A: Canonical tModels
A.1	WSDL Entity Type tModel .....	20
A.1.1	Design Goals .....	20
A.1.2	Definition .....	20
A.1.3	Values .....	20
A.1.4	Example of Use .....	21
A.2	WSDL Namespace tModel .....	21
A.2.1	Design Goals .....	21
A.2.2	Definition .....	21
A.2.3	Values .....	21
A.2.4	Example of Use .....	21
A.3	WSDL Local Name tModel .....	22
A.3.1	Design Goals .....	22
A.3.2	Definition .....	22
A.3.3	Values .....	22
A.3.4	Example of Use .....	22
A.4	WSDL portType Reference tModel .....	22
A.4.1	Design Goals .....	22
A.4.2	Definition .....	22
A.4.3	Values .....	23
A.4.4	Example of Use .....	23
A.5	SOAP Protocol tModel .....	23
A.5.1	Design Goals .....	23
A.5.2	Definition .....	23
A.5.3	Example of Use .....	24
A.6	SOAP over HTTP Transport tModel .....	24
A.6.1	Design Goals .....	24
A.6.2	Definition .....	24
A.6.3	Example of Use .....	24
		Appendix B: Using XPointer in overviewURL
B.1	XPointer Syntax .....	25