software AG

CentraSite

Access via UDDI

Version 9.5 SP1

November 2013

CentraSite

Table of Contents

Access via UDDI

UDDI (Universal Description, Discovery and Integration) is a registry model that can be used as the basis of a service oriented architecture. It is an OASIS standard and its full specification and other related links are available at **https://www.oasis-open.org/**.

The current release of UDDI is V3.0.2. In this document, the UDDI version number is abbreviated to "V3".

The following sections of this document describe CentraSite's support for this UDDI release, and how to configure your UDDI environment within CentraSite.

The document contains the following sections:

Overview of the UDDI Standard	Provides a summary of the OASIS UDDI standard.
Summary of UDDI Support in CentraSite	Describes CentraSite's coverage of the UDDI V3 specification.
CentraSite's UDDI Architecture	Provides a summary of the architecture of CentraSite's UDDI implementation.
UDDI Representation of the CentraSite Object Model	Provides a summary of how the CentraSite Object Model is represented by UDDI.
Configuring the UDDI Environment	Describes how to configure the UDDI environment in CentraSite by modifying configuration properties.
Predefined Value Sets	Lists the value sets from the UDDI specification that are supported by CentraSite.
Predefined tModels	Lists CentraSite's predefined tModels.
UDDI V3 API	Describes the UDDI V3 APIs, which you can use to interact with the CentraSite Registry/Repository directly from UDDI-compliant browsers and integration development environment (IDE) tools.
Using Third-Party IDE Tools with CentraSite	Lists the integrated development environment (IDE) tools that are supported by CentraSite, and describes how to use some of them with CentraSite.
UDDI Extensions	Describes the UDDI extensions implemented by CentraSite.
UDDI Error Messages	Lists UDDI-related error codes, explanations and actions.

1 Overview of the UDDI Standard

UDDI (Universal Description Discovery & Integration) is a platform-independent standard, maintained by the OASIS consortium (**https://www.oasis-open.org**), that describes a Service Oriented Architecture (SOA) registry and its interfaces. UDDI allows clients to discover registered businesses (organizations or providers) and the web services they provide. UDDI also provides programming interfaces to create and update the stored registry information.

The UDDI data model defines the following entity types:

Entity Type	Description
businessEntity	Represents a business.
businessService	Represents one or more web services provided by a business.
bindingTemplate	Describes how to use a web service.
tModel	Categorizes a web service type.
publisherAssertion	Represents a relationship between two business entities.

UDDI defines a set of APIs for accessing and modifying the data stored in the registry.

These APIs include:

- **UDDI Inquiry**
  This API allows you to search for registry entries and retrieve information about them. The search mechanism allows the use of browse patterns (i.e. wildcards) to be used, so that a set of matching business entities can be returned. The use of a subsequent so-called drill-down pattern allows information to be retrieved from a business entity.

- **UDDI Publication**
  This API allows you to add entries to the registry or modify existing entries.

▪ **UDDI Security**
This API determines which security settings apply for a registry entity.

▪ **Custody and Ownership Transfer**
This API set is for transferring UDDI objects between multiple nodes of a UDDI registry and transferring the ownership of UDDI objects between users.

▪ **Replication**
This API set is for synchronizing the data of multi-node UDDI registries.

In addition, the UDDI specification defines the following client API set:

▪ Value Set

For full information, the specification of the current UDDI version and of previous versions is available at **http://uddi.xml.org/specification**.

2 Summary of UDDI Support in CentraSite

CentraSite provides extensive support for UDDI V3. The full list of features of UDDI V3 is available at the OASIS web site mentioned above.

The features currently not supported in CentraSite are as follows:

- **Value Set API Set**
  This API is not supported.

- **Replication**
  Replication is not supported.

- **Publishing Across Multiple Registries**
  This feature is not supported.

- **UDDI Policies**
  UDDI policies are not supported.

    Note: Note that UDDI policies are not the same as web service policies.

- CentraSite supports ownership transfer but not inter-node custody transfer.

- **Multi-Version Support**
  The following features are not supported:

  - Migrating version 2 keys to a version 3 registry

  - Multiple `xml:lang` attributes of the same language

  - Supporting external value set providers across versions

  - White space handling

  - Multiple `overviewDoc` data

  - Multiple `personName` data

3 CentraSite's UDDI Architecture

Overview

CentraSite behaves like a UDDI registry, as described in the UDDI specification. The main components of CentraSite's UDDI environment are:

- The CentraSite Registry/Repository, in which the UDDI objects are stored.

- One or more UDDI servlets running on different application servers. Each UDDI servlet implements the web services of the UDDI API sets.

The following figure illustrates the multiple UDDI servlet scenario, in which multiple UDDI clients and a JAXR client interact with a single CentraSite registry/repository:
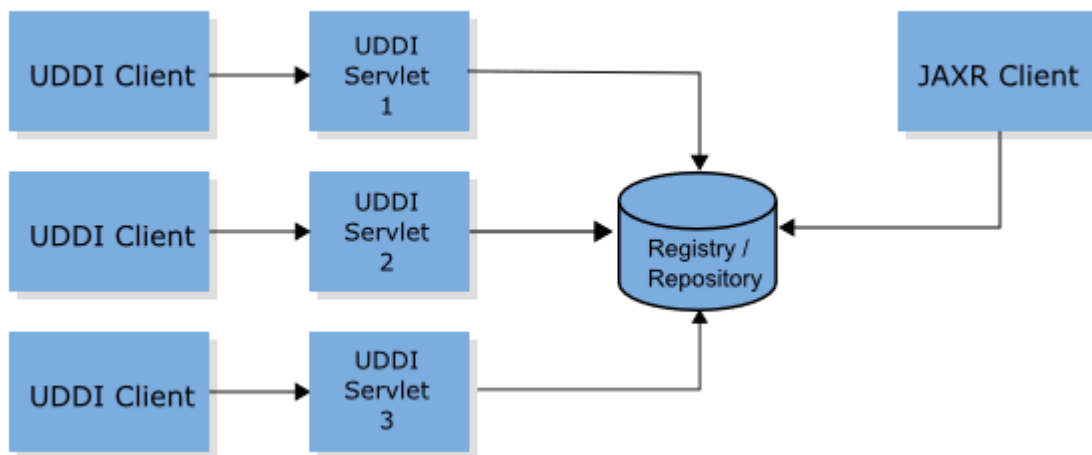
Although there can be multiple UDDI servlets, a CentraSite installation is a single-node UDDI registry. Each UDDI servlet just provides an alternative endpoint for the UDDI web services.

When UDDI data is stored in the CentraSite Registry/Repository, it is mapped to a data model that is common for JAXR and UDDI clients. The data model is an XML representation of JAXR data. Since this representation is also used to store data from the JAXR API, UDDI and JAXR clients act on the same data. Note that JAXR instance-based security is the basis of the CentraSite UDDI security, so changes in the JAXR instance-based security may affect UDDI security.

Client Access to CentraSite via UDDI

UDDI clients can access the CentraSite registry using the following URLs.

The URL for the inquiry API is: *http://<hostname>:53307/UddiRegistry/inquiry*, where *<hostname>* is the name of the host machine. For example, if the UDDI client is running on the same machine as the UDDI servlet, the URL is *http://localhost:53307/UddiRegistry/inquiry*.

The URL for the publish API is: *http://<hostname>:53307/UddiRegistry/publish*.

Every UDDI servlet in a multiple UDDI servlet environment has these endpoints.

Localization

Localization for UDDI means that the error messages are localized. These messages are given in the content of an `errorInfo` element in the disposition report. Following is an example disposition report:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" ↵
xmlns:xsd="http://www.w3.org/2001/XMLSchema" ↵
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>Client</faultcode>
      <faultstring>Client Error</faultstring>
      <faultactor />
      <detail>
      <dispositionReport generic="3.0" xmlns="urn:uddi-org:api_v3">
        <result errno="10120">
          <errInfo errCode="E_authTokenRequired">
            The authentication token value dummy passed in the authInfo
            argument of the UDDI request is not valid.
          </errInfo>
        </result>
      </dispositionReport>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Modeling of the Node Business Entity

The modeling of the Node Business Entity follows the *Recommended Modeling of Node Business Entity* specified in section 6.2.2 of the UDDI 3.0.2 specification.

The following sections describe the registry constructs used to model the Node Business Entity.

- Key Generator tModel
- Node Business Entity
- The Inquiry Service
- The Publish Service
- The Security Service
- Custody and Ownership Transfer Service
- WSDL

Key Generator tModel

The UDDI objects needed for the self-modeling of the node business entity have keys defined internally by CentraSite that belong to the key partition that is owned by the key generator:

```
<tModel
  tModelkey="...:keyGenerator"
  xmlns="urn:uddi-org:api_v3">
    <name xml:lang="en-US">centrasite-node-com:keyGenerator</name>
    <description xml:lang="en-US">
        Key generator for self registering node business entity
    </description>
    <categoryBag>
        <keyedReference
          tModelkey="...:keyGenerator"
          keyName="uddi-org:types:keyGenerator"
          keyValue="keyGenerator"/>
      </categoryBag>
    </categoryBag>
</tModel>
```

The key generator is preloaded in the UDDI registry.

Node Business Entity

The node business entity that is preloaded into the Registry/Repository looks like this:

```
<businessEntity
  businessKey="...">
    <name xml:lang="en-US">NodeBusinessEntity</uddi:name>
    <description xml:lang="en-US">
        Node Business Entity of the CentraSite UDDI registry
    </description>
    <categoryBag>
        <uddi:keyedReference
          keyName=""
          keyValue="node"
          tModelkey="...:keyGenerator"/>
    </categoryBag>
</businessEntity>
```

The node business entity references certain `businessService` objects which reflect the UDDI API sets. The first UDDI registry interacting with the Registry/Repository adds the `businessService` objects. It also adds the `bindingTemplate` objects pointing to the API set's endpoints offered by the UDDI registry. Every additional UDDI registry in a multiple UDDI servlet environment adds `bindingTemplate` objects pointing to the additional endpoints where the UDDI services can be called. The tModels referenced by the `businessService` objects are preloaded into the UDDI registry. Due to the fact that multiple `bindingTemplate` objects are defined for each UDDI registry , the `bindingTemplate` objects get node generated keys.

- Inquiry API set

- Publication API set

- Security Policy API set

- Custody and Ownership Transfer API Set

The Inquiry Service

```
<businessService
  serviceKey="uddi:centrasite.node.com:service_inquiry"
  businessKey="..."
  xmlns="urn:uddi-org:api_v3">
    <name xml:lang="en-US">UDDI Inquiry Services</name>
    <description xml:lang="en-US">Web Service supporting UDDI Inquiry APIs
    </description>
    <bindingTemplates>
        <bindingTemplate
          bindingKey="..."
          serviceKey="uddi:centrasite.service.com:inquiry">
            <description xml:lang="en-US">
                This binding supports the UDDI Programmer's API Specification
```

```
                    For inquiry
            </description>
            <accessPoint useType="endPoint">
                http://localhost:53307/UddiRegistry/inquiry
            </accessPoint>
            <tModelInstanceDetails>
                <tModelInstanceInfo
                  tModelkey="...:keyGenerator">
                    <description xml:lang="en-US">
                        This access point supports the UDDI Version 2.0
                        Programmer's API Specification for inquiry
                    </description>
                </tModelInstanceInfo>
            </tModelInstanceDetails>
        </bindingTemplate>
        <bindingTemplate
          bindingKey="..."
          serviceKey="uddi:centrasite.node.com:service_inquiry">
            <description xml:lang="en-US">
                This binding supports the UDDI Programmer's API Specification
                for inquiry
            </description>
            <accessPoint useType="endPoint">
                http://localhost:53307/UddiRegistry/inquiry
            </accessPoint>
            <tModelInstanceDetails>
                <tModelInstanceInfo tModelkey="...:keyGenerator">
                    <description xml:lang="en-US">
                        This access point supports the UDDI Version 3.0
                        Programmer's API Specification for inquiry
                    </description>
                </tModelInstanceInfo>
            </tModelInstanceDetails>
        </bindingTemplate>
    </bindingTemplates>
</businessService>
```

The Publish Service

```
<businessService
  serviceKey="uddi:centrasite.node.com:service_publish"
  businessKey="..."
  xmlns="urn:uddi-org:api_v3">
    <name xml:lang="en-US">UDDI Publish API Services</name>
    <description xml:lang="en-US">
        Web Service supporting UDDI specifications
    </description>
    <bindingTemplates>
        <bindingTemplate
          bindingKey="..."
          serviceKey="uddi:centrasite.node.com:service_publish">
```

```
                    <description xml:lang="en">
                        This binding supports the UDDI Programmer's API Specification
                        for publication
                    </description>
                    <accessPoint useType="endPoint">
                        http://localhost:53307/UddiRegistry/publish
                    </accessPoint>
                    <tModelInstanceDetails>
                        <tModelInstanceInfo
                          tModelkey="...:keyGenerator">
                            <description xml:lang="en">
                                This binding supports the UDDI Version 2.0 Programmer's
                                API Specification for publication
                            </description>
                        </tModelInstanceInfo>
                    </tModelInstanceDetails>
            </bindingTemplate>
            <bindingTemplate
              bindingKey="uddi:centrasite.node.com:binding_publish_v3"
              serviceKey="uddi:centrasite.node.com:service_publish">
                <description xml:lang="en">
                    This binding supports the UDDI Programmer's API Specification
                    for publication
                </description>
                <accessPoint useType="endPoint">
                    http://localhost:53307/UddiRegistry/publish
                </accessPoint>
                <tModelInstanceDetails>
                    <tModelInstanceInfo tModelkey="...:keyGenerator">
                        <description xml:lang="en">
                            This binding supports the UDDI Version 3.0 Programmer's
                            API Specification for publication
                        </description>
                    </tModelInstanceInfo>
                </tModelInstanceDetails>
            </bindingTemplate>
        </bindingTemplates>
</businessService>
```

The Security Service

```
<businessService
  serviceKey="uddi:centrasite.node.com:service_security"
  businessKey="..."
  xmlns="urn:uddi-org:api_v3">
    <name xml:lang="en">UDDI Security Service</name>
    <description xml:lang="en-US">
        Web Service supporting UDDI Security API
    </description>
    <bindingTemplates>
        <bindingTemplate
```

```
            bindingKey="..."
            serviceKey=" uddi:centrasite.node.com:service_security">
              <description xml:lang="en">
                  This binding to authenticate with the UDDI services using the
                  UDDI Security API.
              </description>
              <accessPoint useType="endPoint">
                 http://localhost:53307/UddiRegistry/publish
              </accessPoint>
              <tModelInstanceDetails>
                  <tModelInstanceInfo tModelkey="...:keyGenerator">
                      <description xml:lang="en">
                          This binding's supports the UDDI v3 Security API.
                      </description>
                  </tModelInstanceInfo>
              </tModelInstanceDetails>
          </bindingTemplate>
      </bindingTemplates>
</businessService>
```

Custody and Ownership Transfer Service

```
<businessService
  serviceKey="uddi:centrasite.node.com:service_ownership_transfer"
  businessKey="..."
  xmlns="urn:uddi-org:api_v3">
    <name xml:lang="en">UDDI Custody and Ownership Transfer API</name>
    <description xml:lang="en-US">
       Web Service providing partly support for the UDDI Custody and Ownership
       Transfer API
    </description>
    <bindingTemplates>
        <bindingTemplate
          bindingKey="uddi..."
          serviceKey=" uddi:centrasite.node.com:service_ownership_transfer">
            <description xml:lang="en-US">
                This binding provides partly support for the UDDI Custody and
                Ownership Transfer API.
            </description>
            <accessPoint useType="endPoint">
               http://localhost:53307/UddiRegistry/publish
            </accessPoint>
            <tModelInstanceDetails>
                <tModelInstanceInfo
                  tModelkey="...:keyGenerator">
                    <description xml:lang="en-US">
                        This binding provides partly support for the UDDI
                        Custody and Ownership Transfer API
                    </description>
                </tModelInstanceInfo>
            </tModelInstanceDetails>
```

```
        </bindingTemplate>
    </bindingTemplates>
</businessService>
```

WSDL

The referenced tModels refer to the WSDL file shown below. Each port, specified by the `<port>` element, specifies an access point. The WSDL representation is a description of the web services that are provided by the UDDI servlets.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="UDDI_API_V3"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:api_v3_binding="urn:uddi-org:api_v3_binding"
  xmlns:tns="urn:uddi-org:api_v3"
  targetNamespace="urn:uddi-org:api_v3">

    <documentation>
    UDDI V3 Security, Publication, Inquiry and Transfer APIs.
    </documentation>

    <import
      namespace="urn:uddi-org:api_v3_binding"
      location="http://uddi.org/wsdl/uddi_api_v3_binding.wsdl"/>
    <import
      namespace="urn:uddi-org:api_v3_binding"
      location="http://uddi.org/wsdl/uddi_custody_v3_binding.wsdl"/>
    <service name="UDDI_Security_SoapService">
        <port
          name="UDDI_Security_PortType"
          binding="api_v3_binding:UDDI_Security_SoapBinding">
            <soap:address
              location="http://localhost:53307/UddiRegistry/publish"/>
        </port>
    </service>
    <service name="UDDI_Publication_SoapService">
        <port
          name="UDDI_Publication_PortType"
          binding="api_v3_binding:UDDI_Publication_SoapBinding">
            <soap:address
              location="http://localhost:53307/UddiRegistry/publish"/>
        </port>
    </service>
    <service name="UDDI_Inquiry_SoapService">
        <port
          name="UDDI_Inquiry_PortType"
          binding="api_v3_binding:UDDI_Inquiry_SoapBinding">
            <soap:address
              location="http://localhost:53307/UddiRegistry/inquiry"/>
        </port>
```

```
    </service>
    <service name="UDDI_Ownership_Transfer_SoapService">
        <port
          name="UDDI_Inquiry_PortType"
          binding="api_v3_binding:UDDI_Inquiry_SoapBinding">
            <soap:address
              location="http://localhost:53307/UddiRegistry/inquiry"/>
        </port>
    </service>
</definitions>
```

The WSDL defines the proper endpoints of each web service. Each UDDI registry adds a port pointing to its endpoints. The first UDDI registry also inserts the service elements. The WSDL file can be accessed via the URIs:

- *http://localhost:53307/UddiRegistry/inquiry?WSDL*

- *http://localhost:53307/UddiRegistry/publish?WSDL*

The WSDL is stored in the CentraSite repository.

4   UDDI Representation of the CentraSite Object Model

This section contains the following topics.

Attributes

The purpose of attributes in CentraSite is to associate values, classifications and associations with names. For the UDDI representation in CentraSite, the attributes are grouped together in a `uddi:keyedReferenceGroup` holding all attributes of the UDDI object. CentraSite supports attributes of the following types:

- Key/Value Pair Attributes
- Rich Text Attributes
- Document Attributes
- Relationship Attributes

Key/Value Pair Attributes

Key/value pair attributes are represented in a `uddi:categoryBag`, as follows:

```
<categoryBag>
  <keyedReference
     tModelKey="uddi:uddi.org:categorization:general_keywords"
     keyName="Encryption Required" keyValue="false"/>
  <keyedReference
     tModelKey="uddi:uddi.org:categorization:general_keywords"
     keyName="Demo Available" keyValue="false"/>
</categoryBag>
```

Rich Text Attributes

For tModels and instanceDetails, Rich Text attributes are represented in a `uddi:overviewDoc` element, as follows.

```
…
<overviewDoc>
<description>
  Detailed Description
</description>
<overviewUrl useType="uddi:centrasite.com:attributes:richText">
  http://10.22.21.94:2020/...
</overviewUrl>
</overviewDoc>
…
```

For all other UDDI objects, especially `uddi:business` and `uddi:businessService`, a modified, UDDI-conformant `uddi:keyedReferenceGroup` approach provides more flexibility. However, the

document referencing semantics cannot be interpreted by a standard UDDI client. Therefore, for Rich Text attributes a dedicated `uddi:keyedReferenceGroup` is introduced, as follows:

```
<categoryBag>
  <keyedReferenceGroup tModelKey="uddi:centrasite.com:attributes:richText">
    <keyedReference
          tModelKey="uddi:uddi.org:categorization:general_keywords"
          keyName="Detailed Description"
          keyValue="http://... "/>
  </keyedReferenceGroup>
</categoryBag>
```

The `uddi:tModelKey` is pointing to `uddi:uddi.org:categorization:general_keyword`s, to make the `uddi:keyName` meaningful. The `uddi:keyValue` points to the URL of the rich text. All `uddi:keyedReference`s representing a rich text attribute are stored in a single `uddi:keyedReferenceGroup`.

Document Attributes

CentraSite supports two different documents types:

- `reference`: This type is used for referencing documents that can be shared between registry objects. These are also called supporting documents.

- `contains`: This type is used for documents that belong exclusively to the definition of a registry object. Examples are WSDL and schema documents that define a Web service.

For tModels and instanceDetails, documents are represented in a `uddi:overviewDoc` element, as follows.

```
…
<overviewDoc>
<description>
  sample category/Documentation
</description>
<overviewUrl useType="uddi:centrasite.com:attributes:document:reference">
  http://10.22.21.94:2020/...
</overviewUrl>
</overviewDoc>
<overviewDoc>
<description>
  sample Asset File
</description>
<overviewUrl useType="uddi:centrasite.com:attributes:document:contains">
  http://10.22.21.94:2020/...
</overviewUrl>
</overviewDoc>
…
```

For all other UDDI objects, a modified `uddi:keyedReferenceGroup`-based approach is used, which conforms to the UDDI specification:

```
<categoryBag>
<keyedReferenceGroup
      tModelKey="uddi:centrasite.com:attributes:document:reference">
    <keyedReference
        tModelKey="uddi:uddi.org:categorization:general_keywords"
        keyName="sample category/Documentation"
        keyValue="http://10.22.21.94:2020/..."/>
</keyedReferenceGroup>
<keyedReferenceGorup
      tModelKey="uddi:centrasite.com:attributes:document:contains">
    <keyedReference
        tModelKey="uddi:uddi.org:categorization:general_keywords"
        keyName="Asset File" keyValue="http://10.22.21.94:2020/..."/>
  </keyedReferenceGroup>
</categoryBag>
```

For both document types (`reference` and `contains`), a separate `uddi:keyedReferenceGroup` is needed. The `uddi:tModelKey` of the `uddi:keyedReferenceGroup` specifies the document type. For each document type, a separate `uddi:tModel` is defined. Each `udddi:keydReference` represents a single document. The `uddi:tModelKey` points to `uddi:uddi.org:categorization:general_keywords`, to add meaning to the `uddi:keyName`. The `uddi:keyName` specifies the location of the file, and the `uddi:keyValue` holds the URL of the document. All `uddi:keyedReferences` representing documents of one type are stored in a single `uddi:keyedReferenceGroup`.

Relationship Attributes

Relationships are represented in a `uddi:keyedReferenceGroup`, as follows:

```
<categoryBag>
<keyedReferenceGroup
            tModelKey="uddi:centrasite.com:attributes:relationShip">
    <keyedReference
                tModelKey="uddi:uddi.org:categorization:general_keywords"
                keyValue="uddi:3447…"
                keyName="Invokes"/>
    <keyedReference
                tModelKey="uddi:uddi.org:categorization:general_keywords"
                keyValue="uddi:3447…"
                keyName="RelatedTo"/>
  </keyedReferenceGroup>
</categoryBag>
```

The `uddi:tModelKey` indicates that a relationship is represented. Another `uddi:tModelKey` uses `uddi:uddi.org:categorization:general_keywords` to add meaning to the `uddi:keyNames`. The `uddi:keyValue` points to the reference registry object, and the `uddi:keyName` specifies the type of

the relationship. The available relationship types are determined by the attribute meta data. All `uddi:keyedReference`s representing relationships are stored in a single `uddi:keyedReferenceGroup`.

Metrics Definition

Following the approach proposed by the Governance Interoperability Framework, saving metrics information for service objects is accomplished by publishing a uddi:tModel. This means that a uddi:tModel holding the metrics information is published and attached to the given uddi:businessService.

Following is a simple uddi:tModel the metrics information:

```
 <tModel tModelKey="uddi:3d32ac10-5dd1-11da-88b8-51d47e6188b2" deleted="false" ↵
xmlns="urn:uddiorg:api_v3">
<name>Metrics</name>
<description>Metrics of EchoAccessPoint</description>
<categoryBag>
<keyedReference
  tModelKey="…(key of object type taxonomy)"
  keyName="Metrics"
  keyValue="Metrics"/>

  <keyedReference
    tModelKey="uddi:centrasite.com:management:metrics:total.request.count"
    keyName="Count of hits"
    keyValue="14"/>

</categoryBag>
</tModel>
```

The uddi:keyedReference of the uddi:categoryBag hold the metrics information. The following metrics are supported.

- Total Request Count
- Success Request Count
- Fault Count
- Average Response Time
- Minimum Response Time
- Maximum Response Time
- Availability
- Service Liveliness

For each of the supported metrics an according taxonomy is defined that represents the according value set.

For attaching a metrics uddi:tModel to a given uddi:service, a uddi:keyedReference is used, which references the metrics-reference taxonomy. Following is an example of a uddi:businessService that references a metrics uddi:tModel:

```
<businessService serviceKey="uddi:1d233560-5dc8-11da-88b7-51d47e6188b2" ↵
businessKey="uddi:a2b32100-
5ac0-11da-8540-e2406020853d" xmlns="urn:uddi-org:api_v3">
  <name>EchoHeadersService</name>
  <description>wsdl:type representing service</description>
  <bindingTemplates>
    <bindingTemplate bindingKey="uddi:1d25a660-5dc8-11da-88b7-51d47e6188b2"
      serviceKey="uddi:1d233560-5dc8-11da-88b7-51d47e6188b2">
      <description>wsdl:type representing port</description>
      <accessPoint
useType="http://schemas.xmlsoap.org/soap/http">http://tracy:4400/sst/runtime.asvc/com.actional.soap
station.Echo
      </accessPoint>
      <tModelInstanceDetails>
        <tModelInstanceInfo
          tModelKey="uddi:1cd8bee0-5dc8-11da-88b7-51d47e6188b2">
          <instanceDetails>
            <instanceParms>EchoHeaders</instanceParms>
          </instanceDetails>
        </tModelInstanceInfo>
        <tModelInstanceInfo
          tModelKey="uddi:1cafda20-5dc8-11da-88b7-51d47e6188b2"/>
      </tModelInstanceDetails>
      <categoryBag>
        <keyedReference tModelKey="uddi:uddi.org:wsdl:types"
          keyName="uddi.org:wsdl:types" keyValue="port"/>
      </categoryBag>
    </bindingTemplate>
  </bindingTemplates>
  <categoryBag>
  <keyedReference
    tModelKey="uddi:centrasite.com:management:metrics:reference"
    keyName="Metrics"
      keyValue="uddi:3d32ac10-5dd1-11da-88b8-51d47e6188b2"/>

  <keyedReference tModelKey="uddi:uddi.org:xml:namespace"
    keyName="uddi.org:xml:namespace"
      keyValue="http://sanity/test"/>

  <keyedReference tModelKey="uddi:uddi.org:wsdl:types"
    keyName="uddi.org:wsdl:types" keyValue="service"/>

  <keyedReference tModelKey="uddi:uddi.org:xml:localName"
    keyName="uddi.org:xml:localName"
```

```
        keyValue="EchoHeadersService"/>
  </categoryBag>
</businessService>
```

The following sections describe the Metrics Reference taxonomy, the Metrics Types taxonomy and the taxonomies for each of the supported metrics (i.e., Total Request Count, Success Request Count, etc.):

- The Metrics Reference Taxonomy
- The Metrics Types Taxonomy
- The Total Request Count Taxonomy
- The Success Request Count Taxonomy
- The Fault Request Count Taxonomy
- The Average Response Time Taxonomy
- The Minimum Response Time Taxonomy
- The Maximum Response Time Taxonomy
- The Availability Taxonomy
- The Service Liveliness Taxonomy

The Metrics Reference Taxonomy

The Metrics Reference taxonomy is for attaching metrics uddi:tModels to uddi:businessService. The uddi:tModel for defining the taxonomy looks as follows:

```
<tModel tModelKey="uddi: centrasite.com:management:metrics:reference">
  <name> centrasite -com:management:metrics:reference</name>
<description>
  Reference to a tModel containing all metrics about the Service
</description>
  <categoryBag>
    <keyedReference
      keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference
      keyName="uddi-org:types:checked"
      keyValue="checked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference
      keyName="entityKeyValues"
      keyValue="tModelKey"
      tModelKey="uddi:uddi.org:categorization:entitykeyvalues"/>
  </categoryBag>
</tModel>
```

The Metrics Types Taxonomy

The Metrics Types taxonomy is needed for classifying the metrics value set taxonomies:

```
<tModel tModelKey="uddi:centrasite.com:management:metrics:types">
  <name>CentraSite Metrics taxonomy </name>
<description>
Taxonomy holding all types of metrics known by CentraSite
</description>
  <categoryBag>
    <keyedReference
      keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference
      keyName="uddi-org:types:unchecked"
      keyValue="unchecked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference
      keyName="entityKeyValues"
      keyValue="tModelKey"
      tModelKey="uddi:uddi.org:categorization:entitykeyvalues"/>
  </categoryBag>
</tModel>
```

The Total Request Count Taxonomy

The following metrics tModel indicates the total number of requests:

```
<tModel tModelKey="uddi:centrasite.com:management:metrics:total.request.count">
  <name>Total Request Count</name>
  <description> Represents Metrics Total Request Count</description>
  <categoryBag>
    <keyedReference keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="uddi-org:types:unchecked"
      keyValue="unchecked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="Metric Type"
      keyValue="Metrics Type"
      tModelKey="uddi:centrasite.com:management:metrics:types"/>
  </categoryBag>
</tModel>
```

The Success Request Count Taxonomy

The following metrics tModel indicates the number of successful requests:

```
<tModel tModelKey="uddi:centrasite.com:management:metrics:success.request.count">
  <name>Success Request Count</name>
  <description>Represents Metrics Success  Request Count</description>
  <categoryBag>
    <keyedReference keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>
    <keyedReference keyName="uddi-org:types:unchecked"
      keyValue="unchecked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="Metric Type"
      keyValue="Metrics Type"
      tModelKey="uddi:centrasite.com:management:metrics:types"/>

  </categoryBag>
</tModel>
```

The Fault Request Count Taxonomy

The following metrics tModel indicates the number of faults:

```
<tModel tModelKey="uddi:centrasite.com:management:metrics:fault.request.count">
  <name>Fault Request Count</name>
  <description>Represents Metrics Fault Request Count</description>
  <categoryBag>
    <keyedReference keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="uddi-org:types:unchecked"
      keyValue="unchecked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="Metric Type"
      keyValue="Metrics Type"
      tModelKey="uddi:centrasite.com:management:metrics:types"/>
  </categoryBag>
</tModel>
```

The Average Response Time Taxonomy

The following metrics tModel indicates the average response time of requests:

```
<tModel tModelKey="uddi:centrasite.com:management:metrics:average.response.time">
  <name>Average Response Time</name>
  <description>Represents Metrics Average Response Time</description>
  <categoryBag>
    <keyedReference keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="uddi-org:types:unchecked"
      keyValue="unchecked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="Metric Type"
      keyValue="Metrics Type"
      tModelKey="uddi:centrasite.com:management:metrics:types"/>
  </categoryBag>
</tModel>
```

The Minimum Response Time Taxonomy

The following metrics tModel indicates the minimum response time of requests:

```
<tModel tModelKey="uddi:centrasite.com:management:metrics:minimum.response.time">
  <name>Minimum Response Time</name>
<description>Represents Metrics Minimum Response Time</description>
  <categoryBag>
    <keyedReference keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="uddi-org:types:unchecked"
      keyValue="unchecked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="Metric Type"
      keyValue="Metrics Type"
      tModelKey="uddi:centrasite.com:management:metrics:types"/>
  </categoryBag>
</tModel>
```

The Maximum Response Time Taxonomy

The following metrics tModel indicates the maximum response time of requests:

```
<tModel tModelKey="uddi:centrasite.com:management:metrics:maximum.response.time">
  <name>Maximum Response Time</name>
  <description>Represents Metrics Maximum Response Time</description>
  <categoryBag>
    <keyedReference keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="uddi-org:types:unchecked"
      keyValue="unchecked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="Metric Type"
      keyValue="Metrics Type"
      tModelKey="uddi:centrasite.com:management:metrics:types"/>
  </categoryBag>
</tModel>
```

The Availability Taxonomy

The following metrics tModel indicates the Virtual/Proxy service availability:

```
<tModel tModelKey="uddi:centrasite.com:management:metrics:availability">
  <name>Availability</name>
  <description>Represents Metrics Availability</description>
  <categoryBag>
    <keyedReference keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="uddi-org:types:unchecked"
      keyValue="unchecked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="Metric Type"
      keyValue="Metrics Type"
      tModelKey="uddi:centrasite.com:management:metrics:types"/>
  </categoryBag>
</tModel>
```

The Service Liveliness Taxonomy

The following metrics tModel indicates the uptime/downtime of Virtual/Proxy services:

```
<tModel tModelKey="uddi:centrasite.com:management:metrics:service.liveliness">
  <name>Service Liveliness</name>
  <description>Represents Metrics Service Liveliness</description>
  <categoryBag>
    <keyedReference keyName="uddi-org:types:categorization"
      keyValue="categorization"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="uddi-org:types:unchecked"
      keyValue="unchecked"
      tModelKey="uddi:uddi.org:categorization:types"/>

    <keyedReference keyName="Metric Type"
      keyValue="Metrics Type"
      tModelKey="uddi:centrasite.com:management:metrics:types"/>
  </categoryBag>
</tModel>
```

Representing Targets and Target Types

CentraSite stores Targets and Target Types as taxonomies. Thus, a deployment of a service to a target can be represented via a `uddi:keyedReference`.

Representing Status

Services that have a certain status are classified according to the LCM taxonomy.

Representing Version

Version information is represented by mapping the `jaxr:Slot` holding the version information. A `uddi:categoryBag` holding version information appears as follows:

```
<categoryBag>
<keyedReference tModelKey="uddi:uddi.org:categorization:general_keywords" ↵
keyName="Version" keyValue="1.0.0"/>
</categoryBag>
```

Mapping WS-PolicyAttachments

The mapping of WS-PolicyAttachments follows the W3C specification Web Service Policy Attachment version 1.5. This means that a tModel is created to represent the reusable policy expression.

5   Configuring the UDDI Environment

This section describes various configuration options for your UDDI environment.

Configuration Properties

In CentraSite the UDDI registry reflects its behavior in terms of JAXR objects stored in the CentraSite Registry/Repository. This representation can be used to parameterize the behavior.

The behavior of the UDDI processing in CentraSite can be configured using global and local properties.

- UDDI in a Multi-CAST Environment
- Deployment Descriptors
- Setting Global and Local UDDI Properties
- Getting Global and Local UDDI Properties

UDDI in a Multi-CAST Environment

CentraSite supports a multi-CAST (CentraSite Application Server Tier) UDDI registry environment. In this environment, multiple CASTs are running against a single CentraSite Registry/Repository. Each CAST comes with its own UDDI Registry web application. In addition, several pure JAXR-based clients interact with the Registry/Repository.

In such an environment the Registry/Repository represents a single UDDI registry node, albeit multiple UDDI Registries are involved. Each UDDI Registry provides its own endpoints for the UDDI services. Due to the multi-CAST scenario, two different sets of properties are needed:

- Local properties, which specify the behavior of a single UDDI web application. Local properties can be used to change the behavior of the web applications independently.

- Global properties, which specify the global behavior of the UDDI registry. Global properties cannot be changed separately for each UDDI registry.

CAST Registration/Deregistration

You can register/deregister CASTs (and retrieve a list of CASTs) by executing the following commands in the command line interface *CentraSiteCommand.cmd* (Windows) or *CentraSiteCommand.sh* (UNIX) of Command Central. The tool is located in `<CentraSiteInstallDir>`/utilities.

If you start this command line tool with no parameters, you receive a help text summarizing the required input parameters.

The parameters of the command are case-sensitive, so for example the parameter "-url" must be specified as shown and not as "-URL".

Retrieving List of Available CASTs

To retrieve the list of available CASTs, use a command of the following form:

```
CentraSiteCommand list CAST [-url <CENTRASITE-URL>] -user <USER-ID> -password
<PASSWORD>
```

The following table describes the complete set of input parameters that you can use with the `list CAST` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".

For example:

```
CentraSiteCommand list CAST [-url "http://localhost:53307/CentraSite/CentraSite"] ↵
-user "Administrator" -password
"manage"
```

Registering a CAST

To register a CAST, use a command of the following form:

```
CentraSiteCommand add CAST [-url <CENTRASITE-URL>] -user <USER-ID> -password
<PASSWORD>
```

The following table describes the complete set of input parameters that you can use with the `add CAST` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.

Parameter	Description
`-password`	The password of the user identified by the parameter "-user".

For example:

```
CentraSiteCommand add CAST [-url "http://localhost:53307/CentraSite/CentraSite"] ↵
-user "Administrator" -password
"manage"
```

Deregistering a CAST

To deregister a CAST, use a command of the following form:

```
CentraSiteCommand remove CAST [-url <CENTRASITE-URL>] -user <USER-ID> -password
<PASSWORD>
```

The following table describes the complete set of input parameters that you can use with the `remove CAST` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".

For example:

```
CentraSiteCommand remove CAST [-url "http://localhost:53307/CentraSite/CentraSite"] ↵
-user "Administrator" -password
"manage"
```

Deployment Descriptors

In addition to the local and global properties that are stored in the registry, the web application needs a minimum of information to contact the CentraSite Registry/Repository. These parameters are:

- com.centrasite.uddi.store.db
- com.centrasite.uddi.store.dbUserId
- com.centrasite.uddi.store.dbUserPasswordHandle

The credentials are for the unauthenticated read access. Usually the guest account is used here.

If you wish to change the password of the guest account, or if you wish to use another user account instead of the guest account, proceed as follows:

▶ **To change the user ID/password of the web application login account**

1   In the file *<CentraSiteInstallDir>\cast\cswebapps\UddiRegistry\WEB-INF\web.xml*, specify the user ID and password in `com.centrasite.uddi.store.dbUserId` and `com.centrasite.uddi.store.dbUserPasswordHandle`.

    If your configuration includes more than one CAST, you must make this change on each CAST.

2   Update the corresponding user ID/password information that is stored in the CentraSite Registry Repository. Do this by following the instructions given in *Users, Groups, Roles, and Permissions > Changing Passwords of Predefined Users and Login Users* in *CentraSite > Administration*.

3   Restart CentraSite.

Setting Global and Local UDDI Properties

You can set the global and local UDDI properties by executing the following command in the command line interface *CentraSiteCommand.cmd* (Windows) or *CentraSiteCommand.sh* (UNIX) of Command Central. The tool is located in *<CentraSiteInstallDir>*/utilities.

If you start this command line tool with no parameters, you receive a help text summarizing the required input parameters.

The parameters of the command are case-sensitive, so for example the parameter "-url" must be specified as shown and not as "-URL".

▶ **To set global properties**

1   Create an XML configuration file that contains the following predefined UDDI properties. This file should be in Java XML properties format. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <!DOCTYPE properties (View Source for full doctype...)>
- <properties version="1.0">
  <comment>Test UDDI Global Configuration XML</comment>
  <entry key="com.centrasite.uddi.UDDIOperatorName">CentraSite</entry>
  <entry key="com.centrasite.uddi.UDDIDefaultLanguage">en_US</entry>
  <entry key="com.centrasite.uddi.UDDIValueValidation">false</entry>
  <entry key="com.centrasite.uddi.UDDIKeyGeneratorChecks">true</entry>
  <entry key="com.centrasite.uddi.UDDIDiscoveryURLGeneration">false</entry>
  <entry key="com.centrasite.uddi.UDDIEncoding">utf-8</entry>
  <entry key="com.centrasite.uddi.UDDISubscriptionDuration">P1M</entry>
  <entry key="com.centrasite.uddi.UDDIAuthTokenExpiration">P1D</entry>
  <entry key="com.centrasite.uddi.UDDITransferTokenExpiration">P1D</entry>
```

```
   <entry key="com.centrasite.uddi.UDDIV2Inquiry">true</entry>
   <entry key="com.centrasite.uddi.UDDIV2Publish">true</entry>
   <entry ↵
key="com.centrasite.uddi.UDDISendEmptyWebServiceNotifications">false</entry>
   <entry key="com.centrasite.uddi.UDDISendEmptyEmailNotifications">true</entry>
   <entry ↵
key="com.centrasite.uddi.UDDIMinimalNotificationInterval">P0Y0M0DT0H0M30S</entry>
   <entry key="com.centrasite.uddi.UDDINumberOfRetries">3</entry>
   <entry key="com.centrasite.uddi.UDDIMaxSubscriptionThreads">5</entry>
   </properties>
```

Descriptions of these properties are as follows:

Global Property	Description
UDDIOperatorName	Sets the `operator` attribute in UDDI V2 replies. The default value is "CentraSite".
UDDIDefaultLanguage	Sets the default language. All valid language identifiers are allowed. The default value is "en-US".
UDDIValueValidation	Activates or deactivates internal value validation. Valid values: "true" or "false" (default). For more information, see **Checked Value Set Validation**.
UDDIKeyGeneratorChecks	Activates or deactivates the enforcement of keyGenerator tModels for publisher assigned keys. Valid values: "true" (default) or "false".
UDDIDiscoveryURLGeneration	Activates or deactivates the generation of discoveryURL. Valid values: "true" or "false" (default).
UDDIEncoding	Specifies the XML encoding of the UDDI responses. Valid values: "utf-8" (default) or "utf-16".
UDDISubscriptionDuration	The duration of a registry subscription. You can specify a default subscription expiration period for each UDDI web application separately. The subscription duration is specified via an xs:duration instance. It specifies a duration in terms of years, months, days, hours, minutes and seconds. The default value is 1 month (xs:duration("P1M")). How a string holding an xs:duration instance is generated is specified by the following section of the XML Schema (Part 2) specification. For example, to specify a duration of 1 year, 2 months, 3 days, 10 hours and 30 minutes, specify: P1Y2M3DT10H30M. You can also indicate a duration of minus 120 days as: -P120D. Reduced precision and truncated representations of this format are allowed provided they conform to the following: If the number

Global Property	Description
	of years, months, days, hours, minutes, or seconds in any expression equals zero, the number and its corresponding designator *may* be omitted. However, at least one number and its designator *must* be present. The seconds part *may* have a decimal fraction. The designator 'T' shall be absent if all of the time items are absent. The designator 'P' must always be present. For example, P1347Y, P1347M and P1Y2MT2H are all allowed; P0Y1347M and P0Y1347M0D are allowed. P-1347M is not allowed although –P1347M is allowed. P1Y2MT is not allowed. **Note:** A UDDI subscription survives an application server restart.
UDDIAuthTokenExpiration	The duration of an authorization token. This duration is specified via an xs:duration instance. It specifies a duration in terms of years (either 0 or 1), months, days, hours, minutes and seconds. The default value is 1 day (xs:duration("P1D")).
UDDITransferTokenExpiration	The duration of a transfer token. This duration is specified via an xs:duration instance. It specifies a duration in terms of years (either 0 or 1), months, days, hours, minutes and seconds. The default value is 1 day (xs:duration("P1D")).
UDDIV2Inquiry	Enables UDDI V2 Inquiry support. Valid values: "true" (default) or "false".
UDDIV2Publish	Enables UDDI V2 Publish support. Valid values: "true" (default) or "false".
UDDISendEmptyWebServiceNotifications	Specifies whether to allow Web service notifications to be sent even when no changes have occurred. Valid values: "true" (default) or "false".
UDDISendEmptyEmailNotifications	Specifies whether to allow email notifications to be sent even when no changes have occurred. Valid values: "true" or "false" (default).
UDDIMinimalNotificationInterval	The minimum time interval at which to send notifications to subscribers. This duration is specified via an xs:duration instance. It specifies a duration in terms of years (either 0 or 1), months, days, hours, minutes and seconds.

Global Property	Description
	The default value is 30 seconds (xs:duration("P30S")). The minimal resolution is 10 seconds (as well as the minimal possible value).

2   Execute the following command in this format:

```
CentraSiteCommand set UDDI [-url <CENTRASITE-URL>] -user <USER-ID>
-password <PASSWORD> -file <CONFIG-FILE>
```

The following table describes the complete set of input parameters that you can use with the set UDDI utility:

Parameter	Description
-url	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
-user	The user ID of a user who has the "CentraSite Administrator" role.
-password	The password of the user identified by the parameter "-user".
-file	The URI (file: or http:) of the configuration file.

For example:

```
CentraSiteCommand set UDDI [-url "http://localhost:53307/CentraSite/CentraSite"] ↵
-user "Administrator"
-password "manage" -file "config.xml"
```

▶ **To set local properties**

1   Create an XML configuration file that contains the following predefined UDDI properties. This file should be in Java XML properties format. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <!DOCTYPE properties (View Source for full doctype...)>
- <properties version="1.0">
  <comment>Test UDDI Local Configuration XML</comment>
  <entry key="com.centrasite.uddi.UDDIMaxResultSize">*</entry>
  <entry key="com.centrasite.uddi.UDDIMaxSearchKeys">*</entry>
  <entry key="com.centrasite.uddi.UDDIMaxSearchNames">*</entry>
  <entry ↵
key="com.centrasite.uddi.UDDIHTTPGetServicesUrl">http://localhost:53307/UddiRegistry</entry>
  <entry key="com.centrasite.uddi.UDDIResponseValidation">false</entry>
  <entry key="com.centrasite.uddi.UDDIRequestValidation">false</entry>
  </properties>
```

Descriptions of these properties are as follows:

Local Property	Description
UDDIMaxResultSize	Specifies the maximum inquiry result size. Default: An unlimited size (denoted by *).
UDDIMaxSearchKeys	Specifies the maximum number of search keys returned by an inquiry. Default: An unlimited number (denoted by *).
UDDIMaxSearchNames	Specifies the maximum number of search names returned by an inquiry. Default: An unlimited number (denoted by *).
UDDIHTTPGetServicesUrl	Holds the URI for the HTTP Get calls to retrieve UDDI objects from the registry. Default value: http://localhost:53307/UddiRegistry.
UDDIResponseValidation	Enables schema validation on UDDI V2 and V3 responses. Valid values: "true" or "false" (default). **Note:** The appropriate schema file should be stored in the CentraSite repository, in the /projects/uddi folder.
UDDIRequestValidation	Enables schema validation on incoming UDDI V2 and V3 requests. Valid values: "true" or "false" (default). **Note:** The appropriate schema file should be stored in the CentraSite repository, in the /projects/uddi folder.

2   Execute the following command in this format:

```
CentraSiteCommand set UDDI [-url <CENTRASITE-URL>] -user <USER-ID>
-password <PASSWORD> -file <CONFIG-FILE>
```

The following table describes the complete set of input parameters that you can use with the set UDDI utility:

Parameter	Description
-url	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
-user	The user ID of a user who has the "CentraSite Administrator" role.
-password	The password of the user identified by the parameter "-user".
-file	The URI (file: or http:) of the configuration file.

For example:

```
CentraSiteCommand set UDDI [-url "http://localhost:53307/CentraSite/CentraSite"] ↵
-user "Administrator"
-password "manage" -file "config.xml"
```

> **Note:** The UDDI representation of local properties is based on the general approach of
> mapping jaxr:Slots to UDDI.

Getting Global and Local UDDI Properties

You can retrieve the global and local UDDI properties by executing the following command in the command line interface *CentraSiteCommand.cmd* (Windows) or *CentraSiteCommand.sh* (UNIX) of Command Central. The tool is located in `<CentraSiteInstallDir>`/utilities.

If you start this command line tool with no parameters, you receive a help text summarizing the required input parameters.

The parameters of the command are case-sensitive, so for example the parameter "-url" must be specified as shown and not as "-URL".

```
CentraSiteCommand get UDDI [-url <CENTRASITE-URL>] -user <USER-ID>
-password <PASSWORD> ↵
```

The following table describes the complete set of input parameters that you can use with the `get UDDI` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".

For example:

```
CentraSiteCommand get UDDI [-url "http://localhost:53307/CentraSite/CentraSite"] ↵
-user "Administrator"
-password "manage"
```

Executing the above CentraSite Command will print the UDDI global and local configurations accordingly.

Schema Validation of UDDI Requests

All incoming UDDI V3 requests are validated against the XML schemas for UDDI requests, as defined in the OASIS UDDI specification. Invalid requests are rejected. The schema files are stored in the CentraSite repository in the folder */projects/uddi*.

The validation can be activated or deactivated by the local configuration properties `Request Validation` and `Response Validation` (see *Setting Global and Local Properties* for details).

Checked Value Set Validation

CentraSite offers the internal validation of checked value sets. If a `keyedReference` object is published that points to a checked value set, CentraSite checks if the `keyValue` belongs to the value set. UDDI 3.0.2 specifies checked value sets specified by value set taxonomies and value sets that have a validation algorithm. All value sets are represented in UDDI by a tModel. The validation of `keyedReference` objects can be switched off. For this purpose, the global configuration property `Internal Value Set Validation` can be used. To set this property, see *Setting Global and Local Properties*.

If the property is set to "false", no checking is performed but `keyedReference` objects pointing to an existing taxonomy are still mapped to a `jaxr:internal` classification. Invalid references are mapped to external classifications. This means that even a `keyedReference` object with an invalid value that points to a classificationScheme with a concept taxonomy is not rejected but mapped to an external classification. If the property is set to "yes", `keyedReference` objects pointing to checked value sets are checked. Invalid `keyedReference` objects are rejected with the error `E_invalidValue`.

6 Predefined Value Sets

This section lists the value sets from the UDDI specification that are supported by CentraSite.

- Type category system
- Relationships category system
- Entity Key Values category system
- NAICS 1997 Release
- NAICS 2002 Release
- UNSPSC Version 7.3
- ISO 3166 Geographic Code System
- UDDI v2 OwningBusiness category system
- UDDI v2 IsReplacedBy identifier system
- UDDI Entity Key Values category System
- UDDI "Derived From" category system
- UDDI Nodes category system
- General Keyword category system
- Postal Address Structure

7 Predefined tModels

The following is a list of the predefined tModels in CentraSite:

```
AssociationType
CentraSite
CentraSiteFilterType
ClassificationGroup
ContentType
Databases
Interstage Business Process Manager
Object
ObjectType
Origin
PhoneType
PostalAddressAttributes
Products
RepositoryObjectType
URLType
UseType
crossvision Application Composer
crossvision Information Integrator
crossvision Legacy Integrator - ApplinX
crossvision Legacy Integrator - EntireX
crossvision Service Orchestrator
dnb-com:D-U-N-S
http://schemas.xmlsoap.org/ws/2003/03/localpolicyreference
http://schemas.xmlsoap.org/ws/2003/03/policytypes
http://schemas.xmlsoap.org/ws/2003/03/remotepolicyreference
node-centrasite-com:keyGenerator
ntis-gov:naics:1997
ntis-gov:naics:2002
ntis-gov:sic:1987
thomasregister-com:supplierID
ubr-uddi-org:iso-ch:3166-2003
ubr-uddi-org:iso-ch:6523-1998:icd
ubr-uddi-org:postalAddress
```

```
uddi-org: protocol:keyGenerator
uddi-org:UTS-10
uddi-org:andAllKeys
uddi-org:approximateMatch:SQL99
uddi-org:binarySort
uddi-org:bindingSubset
uddi-org:caseInsensitiveMatch
uddi-org:caseInsensitiveSort
uddi-org:caseSensitiveMatch
uddi-org:caseSensitiveSort
uddi-org:categorization:keyGenerator
uddi-org:combineCategoryBags
uddi-org:derivedFrom
uddi-org:diacriticsInsensitiveMatch
uddi-org:diacriticsSensitiveMatch
uddi-org:entityKeyValues
uddi-org:exactMatch
uddi-org:fax
uddi-org:ftp
uddi-org:general_keywords
uddi-org:homepage
uddi-org:hostingRedirector
uddi-org:http
uddi-org:inquiry
uddi-org:inquiry_v2
uddi-org:inquiry_v3
uddi-org:isReplacedBy
uddi-org:keyGenerator
uddi-org:mutualAuthenticatedSSL3
uddi-org:node_custody_transfer_v3
uddi-org:nodes
uddi-org:orAllKeys
uddi-org:orLikeKeys
uddi-org:ownership_transfer_v3
uddi-org:owningBusiness_v3
uddi-org:protocol:http
uddi-org:protocol:soap
uddi-org:publication
uddi-org:publication_v2
uddi-org:publication_v3
uddi-org:relationships
uddi-org:replication_v3
uddi-org:security_v3
uddi-org:serverAuthenticatedSSL3
uddi-org:serviceSubset
uddi-org:signaturePresent
uddi-org:smtp
uddi-org:sortByDateAsc
uddi-org:sortByDateDesc
uddi-org:sortByNameAsc
uddi-org:sortByNameDesc
uddi-org:sortorder:keyGenerator
```

```
uddi-org:suppressProjectedServices
uddi-org:taxonomy
uddi-org:taxonomy_v2
uddi-org:telephone
uddi-org:transport:keyGenerator
uddi-org:types
uddi-org:v3_policy
uddi-org:validatedBy
uddi-org:valueSetCaching_v3
uddi-org:valueSetValidation_v3
uddi-org:wsdl:address
uddi-org:wsdl:categorization:protocol
uddi-org:wsdl:categorization:transport
uddi-org:wsdl:portTypeReference
uddi-org:wsdl:types
uddi-org:xml:localName
uddi-org:xml:namespace
uddi.org:bpel:types
uddi:ebxml.org:collaborationprotocolagreement:v1.0:template
uddi:ebxml.org:collaborationprotocolagreement:v2.0:template
uddi:ebxml.org:collaborationprotocolprofile:v1.0
uddi:ebxml.org:collaborationprotocolprofile:v2.0
uddi:ebxml.org:messageservice:v1.0
uddi:ebxml.org:messageservice:v2.0
uddi:oasis-open.org:wsrp:v1_bindings
uddi:oasis-open.org:wsrp:service_type
uddi:oasis-open.org:wsrp:v1_service_description_porttype
uddi:oasis-open.org:wsrp:v1_markup_porttype
uddi:oasis-open.org:wsrp:v1_registration_porttype
uddi:oasis-open.org:wsrp:v1_portlet_management_porttype
uddi:oasis-open.org:wsrp:v1_service_description_binding_soap
uddi:oasis-open.org:wsrp:v1_markup_binding_soap
uddi:oasis-open.org:wsrp:v1_registration_binding_soap
uddi:oasis-open.org:wsrp:v1_portlet_management_binding_soap
uddi:uddi.org:bpel:wsdlporttypereference
uddi:uddi.org:propertyset
uddi:uddi.org:update_entities_v2
uddi:uddi.org:update_entities_v3
uddi:w3.org:ws-policy:v1.5:attachment:policytypes
uddi:w3.org:ws-policy:v1.5:attachment:remotepolicyreference
uddi:w3.org:ws-policy:v1.5:attachment:localpolicyreference
uddi:untmg.org:businessprocessspecificationschema:v1.10
unspsc-org:unspsc
unspsc-org:unspsc:3-1
unspsc-org:unspsc:v6.0501
```

8   UDDI V3 APIs

Overview

The CentraSite Registry/Repository supports the Java API for XML Registries (JAXR). It also supports UDDI V3- and V2. These APIs enable you to interact with the CentraSite Registry/Repository directly from UDDI-compliant browsers and integration development environment (IDE) tools. For more information about IDE tools, see *Using Third-Party IDE Tools with CentraSite*.

CentraSite provides Javadocs that you can use to create UDDI V3 clients. The Javadocs provide the interfaces you need for implementing the Publish, Inquiry, Security and Taxonomy APIs in your clients. To access the Javadocs, use either of the following links:

jd/uddiv3ClientAPI/index.html

or:

`<CentraSite_installation_root>\Documentation\en\jd\uddiv3ClientAPI`

CentraSite supports the following APIs.

Use this API...	To enable the client to...
Publish	Execute any UDDI publishing API call. For example, you can publish services to CentraSite and publish proxy endpoints for services that already exist in CentraSite. You can publish the following UDDI objects: Organization, Service, ServiceBinding and tModel.
Inquiry	Interrogate CentraSite to retrieve service information. When an active run-time policy's virtual service executes, the Inquiry API will pull the virtual service's information from CentraSite. You can publish the following UDDI objects: Organization, Service, ServiceBinding and tModel.
Security	Execute UDDI security API calls, using authorization tokens.
Taxonomy	Fetch taxonomies and their immediate children. The taxonomies are represented in the tree structure. The Taxonomy API is a custom API.

Classes and Interfaces

The major classes and interfaces available in the Javadocs are described below.

- RegistryService
- RegistryConfiguration
- RegistryFramework
- RegistryAgent
- UDDI_Security_SoapService
- UDDI_Inquiry_SoapService
- UDDI_Publication_SoapService

- UDDI_Taxonomy_SoapService
- CentraSiteBusinessService

RegistryService

`RegistryService` is the core interface to communicate with the UDDI Registry using the UDDI V3 API. This interface contains utility methods to get service stubs for the Publish, Inquiry, Security and Taxonomy APIs. The UDDI operations are performed using their respective service stubs. It also contains a method to connect to the CentraSite Registry/Repository, using authentication tokens.

RegistryConfiguration

`RegistryConfiguration` is a bean class that is used to connect to the CentraSite Registry/Repository, based on the Registry/Repository's configuration details, such as its host, port and URLs. The URLs include the Security URL, Inquiry URL, Publish URL and Taxonomy URL. This class also contains the user credentials of the Registry/Repository.

RegistryFramework

`RegistryFramework` is a helper interface that can be used to get attribute values, relationships and documents. This class contains the helper method `getServiceModifiedDate`, which uses `get_operationalInfo` to get the modified date of the service.

> **Note:** Unlike the UDDI specification, serviceBinding is a contained element of a businessService entity in CentraSite. Thus, when a service's bindingTemplate entity is updated, then the service will be updated as well. This means that when you use the helper method `getServiceModifiedDate`, it will return the same modification time for both the `modified` and `modifiedIncludingChildren` attributes.

RegistryAgent

`RegistryAgent` is an interface that enables a policy enforcement point to query the virtual services in the CentraSite Registry/Repository, and to publish run-time performance metrics to the CentraSite Registry/Repository.

This interface contains the following helper methods:

- `findVirtualServices`, which finds the virtual services that are deployed to the runtime target
- `getVirtualServiceWSDLURL`, which returns the WSDL URL for the specified virtual service
- `saveMetrics`, which saves the run-time performance metrics for the virtual services

UDDI_Security_SoapService

UDDI_Security_SoapService is an interface that can be used for all UDDI Security operations. This interface contains the methods get_authToken and discard_authToken. An instance of this interface can be obtained from RegistryService.

UDDI_Inquiry_SoapService

UDDI_Inquiry_SoapService is an interface that contains methods for all Inquiry operations. An instance of this interface can be obtained from RegistryService.

UDDI_Publication_SoapService

UDDI_Publication_SoapService is an interface that contains methods for all UDDI Publish operations. An instance of this interface can be obtained from RegistryService.

UDDI_Taxonomy_SoapService

UDDI_Taxonomy_SoapService is an interface that uses the method get_conceptDetail to fetch taxonomies and their immediate children. The taxonomies are represented in the tree structure. An instance of this interface can be obtained from RegistryService.

CentraSiteBusinessService

CentraSiteBusinessService is a wrapper class for the BusinessService class. This class contains all methods contained in BusinessService, as well as these additional methods:

- getAttachedPolicyDocURL, which returns the attached policy associated with the service.
- getAttachedPolicyTModelKey, which returns the attached policy tModel key associated with the service.
- getAttributes, which returns the attributes associated with the service in a Map.
- getDocuments, which returns the documents associated with the service in a Map.
- getRelatedObjectKey, which returns the UDDI key of the object that has the specified relationship.

Examples

- Getting the Value of an Attribute
- Getting the Proxy Services for a Specified Target
- Inquiring about a Business Service
- Publishing a Business Service
- Fetching Taxonomies

Getting the Value of an Attribute

The following example shows how to get the value of an attribute named `Life Cycle Status`.

```
//Creating configuration object with host, port
//and user credentials of the registry
RegistryConfiguration regConfig =
   new RegistryConfiguration("localhost", "53307",
   "DefaultUser", "PwdFor_CS21");

//Creating registry service instance using the RegistryConfiguration
RegistryService regService =
   RegistryService.Factory.newInstance(regConfig);

//connection is made (get_authToken will be issued to registry)
regService.connect();

//Inquiring the registry for the service using find_service call

UDDI_Inquiry_SoapService inquirySoapService =
   regService.getInquirySoapService();
FindService findService = new FindService();
Name name = new Name();
name.setValue("UDDI Security Service");
findService.setName(new Name[] {name});
findService.setAuthInfo(regService.getAuthToken());
System.out.println("Name....."+ findService);
ServiceList serviceList = inquirySoapService.find_service(findService);
ServiceInfos serviceInfos = serviceList.getServiceInfos();

//Getting the service Key for the first service
ServiceInfo serviceInfo = serviceInfos.getServiceInfo(0);
String serviceKey = serviceInfo.getServiceKey();

//Getting the service detail
GetServiceDetail getServiceDetail = new GetServiceDetail();
getServiceDetail.setServiceKey(new String[] {serviceKey});
getServiceDetail.setAuthInfo(regService.getAuthToken());
ServiceDetail serviceDetail =
   inquirySoapService.get_serviceDetail(getServiceDetail);
```

```
BusinessService businessService =
   serviceDetail.getBusinessService(0);
//Creating instance of CentraSiteBusinessService
CentraSiteBusinessService csBusinessService = new
CentraSiteBusinessService(businessService);
//Getting the value for the attribute "Life Cycle Status"
String attributeValue =
   csBusinessService.getAttributeValue("Life Cycle Status");
```

Getting the Proxy Services for a Specified Target

The following example shows how to get the proxy services for a specified target.

```
//Creating configuration object with host, port
//and user credentials of the registry
RegistryConfiguration regConfig =
   new RegistryConfiguration("localhost",
   "53307", "DefaultUser", "PwdFor_CS21");

//Creating registry service instance using the RegistryConfiguration
RegistryService regService =
   RegistryService.Factory.newInstance(regConfig);

//connection is made (get_authToken will be issued to registry)
regService.connect();

//Getting the RegistryAgent instance using RegistryService
RegistryAgent registryAgent = regService.getRegistryAgent();

//Getting the ServiceInfos which will contain
//a list of the services deployed in the "Actional" target
ServiceInfos proxyServices =
   registryAgent.findProxyServices("Actional");
```

Inquiring about a Business Service

The following example shows how to fetch the details of a business service, using the UDDI Inquiry API.

```
//RegistryConfiguration containing the host, port, userId and
//password to connect to registry
RegistryConfiguration regConfig =
   new RegistryConfiguration("hostName", "port", "userId", "password");

//Creating the RegistryService using RegistryConfiguration
RegistryService regService =
   RegistryService.Factory.newInstance(regConfig);

//connecting to registry. This method will fetch the AuthToken
//using get_authTokenAPI
```

```
regService.connect();

//Inquiring the registry for the service using find_service call
UDDI_Inquiry_SoapService inquirySoapService =
    regService.getInquirySoapService();

//Constructing the find_service inquiry call
FindService findService = new FindService();
Name name = new Name();
name.setValue("UDDI Inquiry Service");
findService.setName(new Name[] {name});

//Issuing find_service inquiry call to
//CentraSite registry using UDDI_Inquiry_SoapService
ServiceList serviceList = inquirySoapService.find_service(findService);
ServiceInfos serviceInfos = serviceList.getServiceInfos();

//Getting the service Key for the first service
ServiceInfo serviceInfo = serviceInfos.getServiceInfo(0);
String serviceKey = serviceInfo.getServiceKey();

//Getting the service detail
GetServiceDetail getServiceDetail = new GetServiceDetail();
getServiceDetail.setServiceKey(new String[] {serviceKey});
getServiceDetail.setAuthInfo(regService.getAuthToken());
ServiceDetail serviceDetail =
    inquirySoapService.get_serviceDetail(getServiceDetail);
BusinessService businessService =
    serviceDetail.getBusinessService(0);
System.out.println("Fetched Service Name : " +
    businessService.getName()[0].getValue());
```

Publishing a Business Service

The following example shows how to publish a business service, using the UDDI Publish API.

```
//RegistryConfiguration containing the host, port,
//userId and password to connect to registry
RegistryConfiguration regConfig =
    new RegistryConfiguration("hostName", "port", "userId", "password");

//Creating the RegistryService using RegistryConfiguration
RegistryService regService =
    RegistryService.Factory.newInstance(regConfig);

//connecting to registry. This method will fetch the
//AuthToken using get_authTokenAPI
regService.connect();

//Getting the UDDI_Publication_SoapService to publish the
//sample business service
```

```
UDDI_Publication_SoapService publishSoapService =
   regService.getPublishSoapService();

//Constructing the save service call for sample business service
SaveService saveService = new SaveService();
BusinessService businessService = new BusinessService();
Name name = new Name();
name.setValue("Sample Business Service");
businessService.setName(new Name[] {name});

//Setting the auth token using the registry service
saveService.setAuthInfo(regService.getAuthToken());
saveService.setBusinessService(new BusinessService[] {businessService});

//Saving the business service using UDDI_Publication_SoapService
publishSoapService.save_service(saveService);
```

Fetching Taxonomies

The following example shows how to fetch taxonomies, using the Taxonomy API.

```
//RegistryConfiguration containing the host, port,
//userId and password to connect to registry
RegistryConfiguration regConfig =
   new RegistryConfiguration("hostName", "port", "userId", "password");

//Creating the RegistryService using RegistryConfiguration
RegistryService regService =
   RegistryService.Factory.newInstance(regConfig);

//connecting to registry. This method will fetch the
//AuthToken using get_authTokenAPI
regService.connect();

//Getting the taxonomy soap service which is used fetch the taxonomies
UDDI_Taxonomy_SoapService taxonomySoapService =
   regService.getTaxonomySoapService();

//Constructing the get_conceptDetail request
GetConceptDetail getConceptDetail = new GetConceptDetail();

//Fetching the NAICS taxonomy
getConceptDetail.setConceptKey(new String[]
   {"uddi:uddi.org:ubr:categorization:naics:1997"});

//Using UDDI_Taxonomy_SoapService we are fetching the
//taxonomies from CentraSite registry
ConceptDetail conceptDetail =
   taxonomySoapService.get_conceptDetail(getConceptDetail);
```

9 Using Third-Party IDE Tools with CentraSite

Overview

An Integrated Development Environment (IDE) tool for Web services is a user interface provided by any vendor that enables you to publish (submit data) and inquire (search data) in any UDDI registry. CentraSite supports any IDE tool that complies with WSDL and UDDIV3 or UDDIV2 inquiry and publish semantics. Using IDE tools with CentraSite, you can publish, inquire and delete Web services.

- Supported IDE Tools
- Specifying the Inquiry, Publish and Security URLs

Supported IDE Tools

The following IDE tools can be used with CentraSite Version 8.0.

- **WTP Eclipse 1.5.2 plug-in**

- **IBM Rational Application Developer 6.0**

- Parasoft JTest 7.5

- PushToTest

- UDDI4J

- RUDDI

Specifying the Inquiry, Publish and Security URLs

When using any IDE tool, you need to obtain a user account in CentraSite, and also provide an inquiry URL for inquiring a Web service and a publish URL for publishing a Web service. You also need to provide the security URL for those tools that require it (for example, ALSB uses the security URL to get the AuthToken).

The UDDI Publish, Inquiry and Security services are hosted at the following URLs on the CentraSite host machine:

Inquiry URL:

```
http://<hostName>:<port>/UddiRegistry/inquiry
```

Publish URL:

```
http://<hostName>:<port>/UddiRegistry/publish
```

Security URL:

```
http://<hostName>:<port>/UddiRegistry/security
```

where `<hostName>` is the host name or IP address of the machine on which CentraSite is installed and `<port>` is the port on which CentraSite is listening for http requests.

> **Note:** The `save_binding` call in UDDI sends the access point but does not send the WSDL URL. Therefore, the WSDL URL shows only the access point.

WTP Eclipse 1.5.2 Plug-In

This section describes how to search for a business and publish a service from that business, using the WTP Eclipse client. For more information about using the WTP Eclipse client, see **http://www.eclipse.org/webtools/**.

▶ **To download the plug-in and access the Web Services Explorer**

1   Download the Eclipse Web Tools version 1.5.2 from **http://download.eclipse.org/webtools/downloads/**. This version contains Eclipse 3.2.1.

2   Run your virus scan product to ensure the Eclipse Web Services Explorer opens properly.

3   On the Eclipse SDK screen, click **Window > Open Perspective**.

4   On the **Open Perspective** dialog box, select **J2EE** and then click **OK**.

5   Click **Run > Launch the Web Services Explorer**.

   The **Web Services Explorer** portlet displays.

6   In the Navigator pane, click **UDDI Main**.

7   On the Open Registry screen, type the following URL in the Inquiry URL box:

   ```
   http://<hostName>:<port>/UddiRegistry/inquiry
   ```

   where `<hostName>` is the host name or IP address of the machine on which CentraSite is installed and `<port>` is the port on which CentraSite is listening for http requests.

8   Click **Go**.

   If the registry is active, Eclipse displays its details. On this page, you can find and publish services, businesses and service interfaces (tModels). For more information, refer to the WTP Eclipse product documentation.

9   Click **Go**.

IBM Rational Application Developer 6.0

For more information about the IBM Rational Application Developer 6.0, see **http://www.ibm.com/developerworks/rational/products/rad/**.

▶ **To connect to CentraSite**

1   From the **Window** menu of the tool, click **Open Perspective > J2EE**.

2   Click **Run > Launch the Web Services Explorer**.

3   In the navigation window, click **UDDI Main**.

4   In the **Registry Name** box, type `CentraSite`.

5   In the **Inquiry URL** box, type `http://<host>:<port>/UDDIRregistry/inquiry`, where `<host>` and `<port>` reflect the target CentraSite registry.

▶ **To publish entities**

1   In the Publish URL box, type `http://<host>:<port>/UDDIRegistry/publish`, where `<host>` and `<port>` reflect the target CentraSite registry.

2   Supply your user name and password.

3   Identify the registry to which you want to publish the entity.

4   Provide entity details.

When you attempt to find services and publish a new service, double authentication is required. But when you attempt to publish a service directly, authentication is involved once.

10 UDDI Extensions

Various extensions to the UDDI standard have been published by the standards bodies OASIS and W3C. The extensions described below are implemented in CentraSite's UDDI environment.

Using WSDL in a UDDI Registry

Since a UDDI registry houses information about web services and their providers, it is essential that the information contained in a web service's WSDL document is accurately mapped to the UDDI data model. This means that subsequent search operations to discover a registered web service are possible, based on the information that is mapped from the WSDL.

OASIS has published a recommendation for how the WSDL-to-UDDI mapping should be implemented, and CentraSite implements this recommendation fully. The technical description of the OASIS recommendation for implementing the WSDL-to-UDDI mapping is contained in the OASIS web site in the document **https://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v2.htm**.

Using WS-PolicyAttachment

An XML-based expression grammar for policies is described in the Web Services Policy Framework (WS-Policy) specification, published by the W3C at **http://www.w3.org/Submission/WS-Policy-Attachment/**. The specification also describes how a policy can be associated with a registry object.

The CentraSite UDDI registry supports WS-PolicyAttachment version 1.2 and 1.5. Policy attachments can be either WSDL-based or UDDI-based. Currently, CentraSite supports UDDI-based policy attachments. With UDDI-based policy attachments, the policies are modeled in the UDDI registry using UDDI elements.

- Version 1.2 Support
- Version 1.5 Support

Version 1.2 Support

The CentraSite WS-PolicyAttachment support for version 1.2 covers the following aspects of the WS-PolicyAttachment specification:

Supported Policy Subjects

CentraSite supports the UDDI-based policy attachments for the following policy subjects:

- Service Provider Policy Subject
- Service Policy Subject
- Endpoint Policy Subject

Referencing Remote Policy Expressions

An example for a remote policy reference is shown by the following uddi:businessService that is taken from the WS-PolicyAttachment specification:

```
<businessService serviceKey="…" >
  <name>…</name>
  <description>…</description>
  <bindingTemplates>…</bindingTemplates>
  <categoryBag>
    <keyedReference
        keyName="Policy Expression for example's Web services"
        keyValue="http://www.example.com/myservice/policy"
        tModelKey="uuid:a27078e4-fd38-320a-806f-6749e84f8005" />
  </categoryBag>
</businessService>
```

The uddi:businessService is attached to a WS-Policy that accessible through the URL http://www.example.com/myservice/policy. The uddi:keyedReference represents the attachment. It is referencing the remote policy reference category system via its uddi:tModelKey and its value holds the URI of the policy document.

Registering Reusable Policy Expressions

A reusable policy expression is represented by a dedicated uddi:tModel in the UDDI registry. The following uddi:tModel shows an example:

```
<tModel tModelKey="uuid:04cfa…">
  <name>…</name>
  <description xml:lang="EN">
    Policy Expression for example's Web services
  </description>
  <overviewDoc>
    <description xml:lang="EN">WS-Policy Expression</description>
    <overviewURL>http://www.example.com/myservice/policy</overviewURL>
  </overviewDoc>
  <categoryBag>
    <keyedReference
        keyName="Reusable policy Expression"
        keyValue="policy"
```

```
        tModelKey="uuid:fa1d77dc-edf0-3a84-a99a-5972e434e993" />
    <keyedReference
        keyName="Policy Expression for example's Web services"
        keyValue="http://www.example.com/myservice/policy"
        tModelKey="uuid:a27078e4-fd38-320a-806f-6749e84f8005" />
  </categoryBag>
</tModel>
```

The uddi:tModel comes with two uddi:keyedReferences. The first uddi:keyedReference specifies the uddi:tModel to represent a reusable policy expression. The second one points to the document holding the policy expression. An example that shows the attachment of a reusable policy expression is given by the following uddi:businessService:

```
<businessService serviceKey="…" >
  <name>…</name>
  <description>…</description>
  <bindingTemplates>…</bindingTemplates>
  <categoryBag>
    <keyedReference
        keyName="Policy Expression for example's Web services"
        keyValue="uuid:04cfa…"
        tModelKey="uuid:a27f7d45-ec90-31f7-a655-efe91433527c" />
  </categoryBag>
</businessService>
```

The uddi:businessService holds a keyedReference pointing to the uddi:tModel holding the reusable policy expression. The uddi:tModelKey of the uddi:keyedReference points to the local policy reference uddi:tModel.

Registering Policies in UDDI Version 3

CentraSite supports UDDI-based policy attachments for UDDI version 2 and 3.

tModels to Support UDDI-Based WS-PolicyAttachments

CentraSite provides the tModels necessary to support UDDI-based WS-PolicyAttachments. These tModels from version 1.2 are described in this section.

Remote Policy Reference Category System

This tModel is used to attach a policy to a UDDI entity by referencing the policy's URI.

```
<tModel tModelKey="uddi:schemas.xmlsoap.org:remotepolicyreference:2003_03" >
  <name>http://schemas.xmlsoap.org/ws/2003/03/remotepolicyreference</name>
  <description xml:lang="EN">
    Category system used for UDDI entities to point to an external ↵
WS-PolicyAttachment Policy Expression
    that describes their characteristics. See WS-PolicyAttachment specification for ↵
further details.
  </description>
  <categoryBag>
    <keyedReference
        keyName="uddi-org:types:categorization"
        keyValue="categorization"
        tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4" />
  </categoryBag>
</tModel>
```

WS-Policy Types Category System

This tModel is used to categorize tModels as representing Policy Expressions. There is only one valid value, namely "policy", that indicates this very fact. It is RECOMMENDED that tModels categorized as representing Policy Expressions reference no more and no less than this very Policy Expression using the Remote Policy Reference category system.

```
<tModel tModelKey=" uddi:schemas.xmlsoap.org:policytypes:2003_03" >
  <name>http://schemas.xmlsoap.org/ws/2003/03/policytypes</name>
  <description xml:lang="EN">
    WS-Policy Types category system used for UDDI tModels to characterize them
    as WS-Policy – based Policy Expressions.
  </description>
  <categoryBag>
    <keyedReference
        keyName="uddi-org:types:categorization"
        keyValue="categorization"
        tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4" />
  </categoryBag>
</tModel>
```

Local Policy Reference Category System

This tModel is used to attach a Policy Expression to a UDDI entity by referencing the UDDI entity that represents this Policy Expression. The Local Policy Reference category system is based on tModelKeys. It is expected that referenced tModels are registered with the same UDDI registry and are categorized as representing Policy Expressions using the WS-Policy Types category system.

```
UDDI Key (V3): uddi:schemas.xmlsoap.org:remotepolicyreference:2003_03
UDDI V1,V2 format key: uuid:a27f7d45-ec90-31f7-a655-efe91433527c
Categorization: categorization
Checked: Yes
```

```
<tModel tModelKey="uddi:schemas.xmlsoap.org:localpolicyreference:2003_03" >
  <name>http://schemas.xmlsoap.org/ws/2003/03/localpolicyreference</name>
  <description xml:lang="en">
   Category system used for UDDI entities to point to a WS-Policy Policy Expression ↵
tModel
   that describes their characteristics. See WS-PolicyAttachment specification for ↵
further details.
  </description>
  <categoryBag>
    <keyedReference
       keyName="uddi-org:types:categorization"
       keyValue="categorization"
       tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62aB4" />
    <keyedReference
       keyName="uddi-org:entityKeyValues"
       keyValue="tModelKey"
       tModelKey="uuid:916b87bf-0756-3919-8eae-97dfa325e5a4" />
  </categoryBag>
</tModel>
```

Version 1.5 Support

The CentraSite WS-PolicyAttachment support for version 1.5 covers the same aspects as for version 1.2. Additionally, CentraSite provides the following tModels necessary to support UDDI-based WS-PolicyAttachments:

- uddi:w3.org:ws-policy:v1.5:attachment:localpolicyreference

- uddi:w3.org:ws-policy:v1.5:attachment:policytypes

- uddi:w3.org:ws-policy:v1.5:attachment:remotepolicyreference

Extending UDDI Publisher API Set to Enable Physical Deletion of tModels

For the physical deletion of un-referenced tModels, CentraSite extends the UDDI Publication API set. Appendix H of the UDDI V3 Specification describes how UDDI can be extended. For removing a tModel from a registry, CentraSite introduces the request `purge_tModel`. This extension is only supported for UDDI version 3.

Arguments

The `purge_tModel` request has the same arguments as the `delete_tModel` V3 request:

▪ `authInfo`: This optional argument is an element that contains an authentication token.

▪ `tModelKey`: One or more required uddiKey values that represent specific instances of known tModel data.

Behavior

The request removes a hidden tModel from the registry that is not referenced by any other UDDI object.

Returns

On successful completion an empty message is returned.

Caveats

In the case of an error, a disposition report will be returned within a SOAP fault. In addition to the errors which are common to all API calls, the following errors are relevant here:

▪ `E_invalidKeyPassed`: Signifies that one of the uddiKey values passed did not match any known tModelKey values, or multiple instances of the same tModelKey values were passed. The error is also returned if the referenced tModel is not hidden, or it is still referenced. This means there are four conditions for this error. The different conditions should be reflected in the `errInfo` element:

  ▪ The specified tModel cannot be found

  ▪ Multiple references to the same tModel

  ▪ The specified tModel is not hidden

  ▪ The specified tModel is still referenced

▪ `E_userMismatch`: Signifies that one or more of the tModelKey values passed refers to data that is not owned by the individual publisher who is represented by the authentication token.