



APELON

BECAUSE TERMINOLOGY MATTERS

75 Sgt William B Terry Dr, Suite 2005, Hingham, MA 02043

+1 (203) 431-2530

www.APELON.com

www.ApelondTS.org

DTS 4: Web Services Quick Start Guide

Table of Contents

A. Introduction	3
B. Accessing the DTS WSDL.....	3
C. Generating Java Client Classes (JAX-WS).....	4
D. Creating a DTS Web Service Client	4
E. Getting Further Information.....	6
E.1.1 Web Services Reference Guide	6
E.1.2 Web Service Tutorial	7
E.1.3 Web Service Tutorial Classes	7

A. Introduction

This quick start guide provides basic information on how to connect to an installed DTS4 server using web services. It assumes you have set up DTS4 according to the “dts_InstallationGuide.pdf”. In a client specific production environment, the WSDL URL and the usernames and passwords shown in this document would likely be different.

B. Accessing the DTS WSDL

Start the DTS4 server. You should be able to see the WSDL at this URL:

- <http://localhost:8080/soap/DtsQueryDaows?wsdl>

The URL above assumes the DTS4 server is running on your local machine. If this is not the case, substitute the location of your remote server for `localhost`.

C. Generating Java Client Classes (JAX-WS)

Follow these steps to generate a Java client from the WSDL:

1. Open a terminal or command window and change to your desired working directory.
2. Use the **wsimport** tool (which you can find in the bin directory of your java installation) to generate the JAX-WS portable artifacts:

```
wsimport -keep -Xnocompile http://localhost:8080/soap/DtsQueryDaoWS?wsdl
```

The command generates the .java files that comprise the client.

3. Copy the generated .java files into your IDE.
4. Ignore the .class files.

D. Creating a DTS Web Service Client

To create a functioning web service client you will need a project with the classes generated in the previous step.

1. Create a class called Example like this:

```
import java.util.List;
import java.util.Map;

import javax.xml.ws.BindingProvider;

import com.apelon.dtsserver.types.TConceptAttributeSetDescriptor;
import com.apelon.dtsserver.types.TOntologyConcept;
import com.apelon.dtsserver.types.TProperty;

/**
 * Sample code using a JAX-WS client.
 */
public class Example {

    public static void main(String[] args) {
        try {
            // Instantiate client
            DtsQueryDaoService service = new DtsQueryDaoService();
            System.out.println("Retrieving the port from the following service: "
                + service);
            DtsQueryDaoWS port = service.getDtsQueryDaoPort();

            // Set credentials for JAX-WS.
            Map<String, Object> requestContext =
```

DTS 4: Web Services Quick Start Guide

```
        ((BindingProvider) port).getRequestContext();
requestContext.put(BindingProvider.USERNAME_PROPERTY, "dtsadmin");
requestContext.put(BindingProvider.PASSWORD_PROPERTY, "password");

// This assumes LOINC is loaded into the DTS instance
String conceptCode = "13506-1"; // Urea nitrogen renal clearance
int namespaceId = 5102; // LOINC

// Configure the service to read all property types
TConceptAttributeSetDescriptor attributeSetDescriptor =
    new TConceptAttributeSetDescriptor();
attributeSetDescriptor.setAllPropertyTypes(true);

// Get concept
System.out
    .println("Invoking the findConceptByCode operation on the port.");
TOntologyConcept concept =
    port.findConceptByCode(conceptCode, namespaceId,
        attributeSetDescriptor);

// Print some concept info
String conceptName = concept.getName();
System.out.println("Concept Name: " + conceptName);
System.out.println();

// Print concept properties
List<TProperty> props = concept.getProperties();
System.out.println("Properties\n=====");
for (TProperty prop : props) {
    System.out.println(prop.getPropertyType().getName() + ": "
        + prop.getValue());
}
} catch (Exception ex) {
    ex.printStackTrace();
}
}
```

This example assumes you are running the DTS4 server on the same machine referenced in the `wsimport` call as described in the **Generating Java Client Classes** section above. This is the default behavior. To specify a different host server add a line similar to the below to the example class after the other two `requestContext` puts:

```
requestContext.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY,
    "http://wsserver.MySite.com:8080/soap/DtsQueryDaoWS");
```

2. Run the class as a Java application.
 - a. Make sure DTS is properly deployed to your application server and running.
 - b. If the web services are available, you should be able to see the raw WSDL at the URL at the beginning of this document.

DTS 4: Web Services Quick Start Guide

- c. Make sure the username and password in the “main” method are correct for your deployment
- d. You should see output like the following.

```
Retrieving the port from the following service:
com.apelon.dtsserver.ws.dtsquery.DtsQueryDaoService@4af8d139
Invoking the findConceptByCode operation on the port.
Concept Name: Urea nitrogen renal clearance
```

Properties

=====

```
Component: Urea nitrogen renal clearance
Property: Volume Rate
Time Aspect: Point in time
System: Urine+Ser/Plas
ScaleType: Quantitative
Short Name: UUN Cl Ur+SerPl-vRate
Code in Source: 13506-1
Change Type: MAJ
Date Last Changed: 20070612
Units Required: Y
Order/Observation: Both
Source: LCA
Long Common Name: Urea nitrogen renal clearance
UMLS CUI: C0550556
UMLS Semantic Type: Clinical Attribute
Status: ACTIVE
Class: CHEM
Classtype: 1
Example Unit: mL/min
Example UCUM Units: mL/min
Common SI Test Rank: 0
Synthesized LOINC Name: Urea nitrogen renal clearance:VRat:Pt:Urine+Ser/Plas:Qn:
Related Names: ; Urea nit; UN; Blood urea nitrogen; N2; Nit; N; vRate; Volume rate;
Flow; Point in time; Random; Ur+SerPl; Ur+SerPl; UR; Urn; UA; SerPl; SerPlas; SerP;
Serum; SR; Plasma; Pl; Plsm; Quantitative; QNT; Quant; Quan; Cl; Kidney; Clear;
Clearanc; Clearances; Clr; Chemistry; BUN; UUN
Multiaxial Path To Root: 2 | LP31388-9.LP31398-8.LP14492-0.LP131080-6
```

E. Getting Further Information

DTS provides three additional resources to support web service development:

E.1.1 Web Services Reference Guide

The **DTS Web Services Reference Guide** documents all the DTS web services methods. This guide is available under the *Documentation* menu item in the *DTS Start Menu*.

E.1.2 Web Service Tutorial

The **DTS Web Service Programming Tutorial** can be accessed from a link at the bottom of the DTS Javadoc *Overview* page. This page is also available from *Documentation* menu item in the *DTS Start Menu*. The tutorial follows the same outline as the **DTS Java API Tutorial**.

E.1.3 Web Service Tutorial Classes

Sample Java code that implements the chapters of the **DTS Web Service Programming Tutorial** can be found in the `\samples\wstutorial` folder of the DTS install directory.