Delegated Regulation (EU) Nº 885/2013 of 15 May 2013 with regard to the provision of information services for safe and secure parking places for trucks and commercial vehicles

European Access Point for Truck Parking Data

User Manual: Web Service description to push static DATEX II data related to safe and secure Truck Parking Areas to the European Access point

V1.1
March 2016

EUROPEAN COMMISSION
Directorate-General for Mobility and Transport
Directorate C — Innovative & sustainable mobility
Unit C3 — Intelligent Transport Systems
Contact: MOVE-ITS@ec.europa.eu
European Commission
B-1040 Brussels
Table of Contents

1 Introduction.................................................................................................................................................. 3
  1.1 Background........................................................................................................................................... 3
  1.2 European Access Point for Truck Parking Data .................................................................................... 3
  1.3 Scope of this document .......................................................................................................................... 4

In case of questions please contact: ............................................................................................................. 4

2 Service Overview ......................................................................................................................................... 5

3 Before you start ........................................................................................................................................... 6
  3.1.1 WSDL.................................................................................................................................................. 6
  3.1.2 Login/password ................................................................................................................................. 6

4 Calling the web service using SOAP-UI ..................................................................................................... 7
  4.1 Download Soap-UI ................................................................................................................................. 7
  4.2 Create a supplierPushService project ................................................................................................. 7
  4.3 Call the web service ............................................................................................................................. 8

5 Calling the web service from a Java client .................................................................................................. 9
  5.1 Generate the portable artifacts ............................................................................................................... 9
  5.2 Implement the service endpoint ........................................................................................................... 11
  5.3 Implement a service client with JAX-WS RI ......................................................................................... 11
  5.4 Implement a service client with Apache CXF (alternative JAX-WS implementation) ......................... 13

6 Annexes ..................................................................................................................................................... 15
1 Introduction

1.1 Background

The ITS Directive (2010/40/EU) provides a legal framework in order to support the coordinated and interoperable deployment and use of ITS across Europe. The Directive empowers the Commission to adopt delegated acts. Six priority actions have been defined in the ITS Directive and one of them – Action “e” – concerns the provision of information services for safe and secure parking places for trucks and commercial vehicles.


1.2 European Access Point for Truck Parking Data

Article 5 of delegated regulation 885/2013 requires the public or private parking operators and service providers to share and exchange data related to the parking areas through a national or international access point. The adopted delegated act does not oblige the European Commission to develop an access point. However, the Commission has decided to develop a European Access Point for Truck Parking Data, since this need has been repeatedly expressed by Member States during the preparation phase with Member States experts and after the adoption of delegated regulation 885/2013.

The European Access Point for Truck Parking aims to collect truck parking data according to regulation 885/2013 in order to make the collected data available as DATEX II export. At this moment the Access Point will only support static data as defined in Article 4(1) and 4(2) of the delegated regulation.

DATEX II


DATEX II – Part 6

[http://www.datex2.eu/content/datex-ii-parking-submitted-standardisation](http://www.datex2.eu/content/datex-ii-parking-submitted-standardisation)


The sole responsibility for data quality and the necessary updating/maintenance obligations remain with the Member States that are submitting the data.
Web Service description to push static DATEXII data related to safe and secure Truck Parking Areas to the European Access point

1.3 Scope of this document

A web-based data collection tool of the European Access Point enables Member States to directly enter data according to delegated regulation 885/2013 through a web application. Access to the data entry/collection tool will be granted by the EC to registered Member State representatives only. The entered data will be stored in the database of the European Commission. The tool can further be used to maintain and update the data. The web application tool provides functionalities in order to enable Member States to edit and maintain/update their truck parking data prior to submitting them for publication in DATEX II format on the open data portal of the EC ("export").

This web tool also proposes to Member States to submit data directly in Datex II format compliant with the Truck Parking profile as submitted to standardisation either by:
  - Uploading a Datex II file from the web application
  - Calling a SOAP web service to push Datex II.

The purpose of this document is to provide all the information that is needed for integrating this web service in Responsible Authorities local systems. This document also provides examples of client requests using SOAP-UI (open-source web service testing application) as generic client as well as Java code. C#.Net source code won’t be provided in this document but a client can be easily derived from the example shared on the Datex II forum at: http://www.datex2.eu/content/datex-ii-interfaces-building-soap-and-rest-web-services-java-and-cnet

In case of questions please contact:
MOVE-ITS@ec.europa.eu
2 Service Overview

The web service provides the ability to push static data on Truck Parking Areas to the European Access point by implementing the Datex II truck parking profile.

Please note that using the web service is similar than using the interactive upload function from the truck parking web application. Only the latest valid data submitted by a responsible authority will be published on the EU Open Data Portal. This web service cannot be used for Incremental /partial pushes.

On demand, and to deal with specific cases of federal countries, a single user account (e.g. at federal government level) can be used to push data on behalf of other responsible authorities of the same country (e.g. at regional level).

The web service will then associate files to the target responsible authorities based on the following rules:

- ParkingTable identifiers must start with the country code of the user account supplying data. (e.g. DE for Germany)

- ParkingTable identifiers must be equal to the identifier of a known responsible authority (e.g. DE-BE-ITP / DE-HH-ITP / DE-HB-IT with BE, HH and HB being respectively acronyms for German regions Berlin, Freie und Hansestadt Hamburg, Freie Hansestadt Bremen ...)

This service is derived from the one published on the Datex II web portal at

http://www.datex2.eu/content/datex-ii-exchange-psm-3

and discussed on the forum:

http://www.datex2.eu/content/datex-ii-interfaces-building-soap-and-rest-web-services-java-and-cnet

Difference with Datex II Exchange PSM (in relation to links provided above):

- Only the push service is implemented. Data is made not accessible from a pull web service but only from the EU Open Data Portal.

- Payload must validate the Datex II Truck Parking profile as submitted to standardisation.

- A user/password authentication is required in order to clearly identified the submitter
3 Before you start

Before you start working with our web service, you need the Web Service Description and an applicative user account.

3.1.1 WSDL

SOAP web services are described through Web Services Description Language files (WSDL).

WSDL can be retrieved directly from the targeted environment:

- Acceptance
  

- Production
  

- Or directly with the "supplierPushService.zip" archive provided with this document (includes xjb binding customisation files):

3.1.2 Login/password

A basic authentication to access our web service is required. A login and password can be requested to your contact point at the European Commission: MOVE-ITS@ec.europa.eu

The web services are authenticated by "usenetname:token" without HTTP basic authentication (pure wsse). Confidentiality is ensured by using HTTPS.

If you do not set the authentication information in the SOAP header, or if an incorrect login/password is used, you will get the following responses:

```xml
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Body>
    <env:Fault>
      <env:Sender></env:Sender>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

Or

```xml
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Body>
    <env:Fault>
      <env:Sender></env:Sender>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

The username password must be provided in compliance with the Web Services Security UsernameToken Profile 1.1 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
Web Service description to push static DATEXII data related to safe and secure Truck Parking Areas to the European Access point

4 Calling the web service using SOAP-UI

4.1 Download Soap-UI
The open source version of Soap-UI can be downloaded at


4.2 Create a supplierPushService project

- Create a new SoapUI Project

- Set information about the WSDL.

- A new project is created, that includes all the operations you can call on the specified web service
4.3 Call the web service

- Set the login, password and password type (passwordText)

- Fill in the payload with a data structure compliant with the Truck Parking Profile.
Web Service description to push static DATEXII data related to safe and secure Truck Parking Areas to the European Access point

- Send the request to the server by clicking on

- If the operation terminates successfully, the server responds by an acknowledge:

5 Calling the web service from a Java client

5.1 Generate the portable artefacts

Use the tool wsimport, included in the java development kit to generate the portable artefacts.

- Unzip the archive provided in § 3.1.1
Web Service description to push static DATEXII data related to safe and secure Truck Parking Areas to the European Access point

- Your working directory should be structured this way:

  ![Image of working directory structure]

- Run the following command line

  `%JAVA_HOME%/bin/wsimport .\PushStaticData.wsdl -b .\truckParkingTablePublication\TruckParkingTablePublication.xjb -Xnocompile -XadditionalHeaders`
Web Service description to push static DATEXII data related to safe and secure Truck Parking Areas to the European Access point

5.2 Implement the service endpoint

- Access the default client stub generated by wsimport in package
  
  `eu.datex2.wsdl.supplierpush._2_0`

- Change the default wsdl location to either the acceptance or production environment (can also be define at construction time)

```java
package eu.datex2.wsdl.supplierpush._2_0;
import java.net.UnknownHostException;
import java.net.URL;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.Holder;
import javax.xml.ws.handler.Handler;
import eu.datex2.schema._2._2_0.CountryEnum;
import eu.datex2.schema._2._2_0.D2LogicalModel;
import eu.datex2.schema._2._2_0.Exchange;
import eu.datex2.schema._2._2_0.GenericPublication;
import eu.datex2.schema._2._2_0.InternationalIdentifier;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushInterface;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushService;
public class ServiceConsumer {
    public static void main(String[] args) {
        //Get web service instance
        SupplierPushService pushService = new SupplierPushService();
        SupplierPushInterface supplierService = pushService.getSupplierPushSoapEndPoint();
        //Authentication
        BindingProvider bp = (BindingProvider) supplierService;
        consumerHandler = new ArrayList<Handler>();
        consumerHandler.add(new SecurityHandler());
        bp.getBinding().setHandlerChain(consumerHandler);
        //Build D2Model object
        ObjectFactory factory = new ObjectFactory();
        D2LogicalModel dModel = factory.createD2LogicalModel();
        dModel.setModelBaseVersion("2");
        //Build Exchange Information
        Exchange exchangeInfo = factory.createExchange();
        InternationalIdentifier supplierInfo = factory.createInternationalIdentifier();
        supplierInfo.setCountry(CountryEnum.BE);
    }
}
```

5.3 Implement a service client with JAX-WS RI

For this first basic draft implementation, a simple JRE (6 or above) without any additional library is required (Assumption: the SSL certificate for webgate.ec.europa.eu is trusted).

- Service Consumer

```java
package eu.test;
import java.util.ArrayList;
import java.util.List;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.Holder;
import javax.xml.ws.handler.Handler;
import eu.datex2.schema._2._2_0.CountryEnum;
import eu.datex2.schema._2._2_0.D2LogicalModel;
import eu.datex2.schema._2._2_0.Exchange;
import eu.datex2.schema._2._2_0.GenericPublication;
import eu.datex2.schema._2._2_0.InternationalIdentifier;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushInterface;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushService;
public class ServiceConsumer {
    public static void main(String[] args) {
        //Get web service instance
        SupplierPushService pushService = new SupplierPushService();
        SupplierPushInterface supplierService = pushService.getSupplierPushSoapEndPoint();
        //Authentication
        BindingProvider bp = (BindingProvider) supplierService;
        consumerHandler = new ArrayList<Handler>();
        consumerHandler.add(new SecurityHandler());
        bp.getBinding().setHandlerChain(consumerHandler);
        //Build D2Model object
        ObjectFactory factory = new ObjectFactory();
        D2LogicalModel dModel = factory.createD2LogicalModel();
        dModel.setModelBaseVersion("2");
        //Build Exchange Information
        Exchange exchangeInfo = factory.createExchange();
        InternationalIdentifier supplierInfo = factory.createInternationalIdentifier();
        supplierInfo.setCountry(CountryEnum.BE);
    }
}
```
Web Service description to push static DATEXII data related to safe and secure Truck Parking Areas to the European Access point

References to wsse policies have been removed from the published WSDL to prevent undesired Basic authorization being added automatically to HTTP requests.

Please note that usage of bindingProvider (to set the user/password) is not supported due to HTTP Basic authorization being added automatically to HTTP requests.

References to wsse policies have been removed from the published WSDL to prevent undesired initiatives from JAX-WS RI.

```java
package eu.test;

import javax.xml.ws.handler.soap.SOAPHandler;
import javax.xml.ws.handler.soap.SOAPMessageContext;
import javax.xml.soap.SOAPHeader;
import javax.xml.soap.SOAPFactory;
import javax.xml.soap.SOAPEnvelope;
import javax.xml.soap.SOAPElement;
import java.util.Set;

public class SecurityHandler implements SOAPHandler<SOAPMessageContext> {

    public boolean handleMessage(SOAPMessageContext context) {
        // This method is auto-generated method stub
        return true;
    }

    public boolean handleFault(SOAPMessageContext context) {
        return true;
    }

    public void close(MessageContext context) {
        // This method is auto-generated method stub
        return;
    }

    public Set<QName> getHeaders(SOAPMessageContext context) {
        final HashSet<QName> headers = new HashSet<QName>();
        headers.add(new QName("http://docs.oasis-open.org/wsa/2004/01/owsa-200401-wss-username-token-profile-1.0#UsernameToken", "UsernameToken"));
        headers.add(new QName("http://docs.oasis-open.org/wsa/2004/01/owsa-200401-wss-username-token-profile-1.0#PasswordText", "PasswordText"));
        return headers;
    }

    public void close(MESSAGE_OUTBOUND_PROPERTY property) {
        // This method is auto-generated method stub
        return;
    }

    public void handleMessage(WrappedMessageContext context) {
        // This method is auto-generated method stub
    }

    public void handleFault(WrappedMessageContext context) {
        // This method is auto-generated method stub
    }

    public void close(WrappedMessageContext context) {
        // This method is auto-generated method stub
    }
}
```

• Security Handler
5.4 Implement a service client with Apache CXF (alternative JAX-WS implementation)

- **Prerequisites**
  
  Apache CXF libraries must be added in the project dependencies (Maven group id org.apache.cxf / artefact cxf-bundle)

  Find more information on how to install cxf on your application server at http://cxf.apache.org/docs/application-server-specific-configuration-guide.html

- **Service Consumer**

```java
package eu.test.client;
import java.util.HashMap;
import java.util.Map;
import javax.xml.ws.Holder;
import org.apache.cxf.endpoint.Client;
import org.apache.cxf.endpoint.Endpoint;
import org.apache.cxf.ws.security.wss4j.WSS4JOutInterceptor;
import org.apache.ws.security.WSConstants;
import org.apache.ws.security.handler.WSHandlerConstants;
import eu.datex2.schema._2._2_0.CountryEnum;
import eu.datex2.schema._2._2_0.D2LogicalModel;
import eu.datex2.schema._2._2_0.Exchange;
import eu.datex2.schema._2._2_0.GenericPublication;
import eu.datex2.schema._2._2_0.InternationalIdentifier;
import eu.datex2.schema._2._2_0.ObjectFactory;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushInterface;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushService;

public class ServiceConsumer {
    @SuppressWarnings("unchecked")
    public static void main(String[] args) {
        SupplierPushService pushService = new SupplierPushService();
        SupplierPushInterface supplierService = pushService.getSupplierPushSoapEndPoint();
        // Build d2Model object
        ObjectFactory factory = new ObjectFactory();
        D2LogicalModel d2Model = factory.createD2LogicalModel();
        // Datex Exchange information
        Exchange exchangeInfo = factory.createExchange();
        InternationalIdentifier supplierInfo = factory.createInternationalIdentifier();
        supplierInfo.setCountry(CountryEnum.BE);
        supplierInfo.setNationalIdentifier("test");
        exchangeInfo.setSupplierIdentification(supplierInfo);
        d2Model.setExchange(exchangeInfo);
        d2Model.setModelBaseVersion("2");
        d2Model.setPayloadPublication(new GenericParkingTablePublication());
        Holder<D2LogicalModel> holder = new Holder<D2LogicalModel>(d2Model);
        Client client = org.apache.cxf.frontend.ClientProxy.getClient(supplierService);
        Endpoint cxfEndpoint = client.getEndpoint();
        // can be replaced by a SpringFramework cxf.xml configuration file
        Map outProps = new HashMap<String, Object>();
        outProps.put(WSHandlerConstants.ACTION, WSHandlerConstants.USERNAME_TOKEN);
        outProps.put(WSHandlerConstants.USER, "nmthieol");
        outProps.put(WSHandlerConstants.PASSWORD_TYPE, WSConstants.PW_TEXT);
        outProps.put(WSHandlerConstants.PW_CALLBACK_CLASS, ClientPasswordCallback.class.getName());
        WSS4JOutInterceptor wssOut = new WSS4JOutInterceptor(outProps);
        cxfEndpoint.getOutInterceptors().add(wssOut);
        // call eu web service
        try {
            supplierService.putDatex2Data(holder);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```java
/**
 * Generate the truck parking table publication Object Structure
 */
private static GenericPublication GenericParkingTablePublication() {
    ObjectFactory factory = new ObjectFactory();
    GenericPublication publication = factory.createGenericPublication();
    // TODO build the payload here
    // ....
    // ....
    return publication;
}
```
ClientPasswordCallback

```java
package eu.test.client;
import java.io.IOException;
import javax.security.auth.callback.Callback;
import javax.security.auth.callback.CallbackHandler;
import javax.security.auth.callback.UnsupportedCallbackException;
import org.apache.ws.security.WSPasswordCallback;

public class ClientPasswordCallback implements CallbackHandler {
    public void handle(Callback[] callbacks) throws IOException,
            UnsupportedCallbackException {
        WSPasswordCallback pc = (org.apache.ws.security.WSPasswordCallback) callbacks[0];
        if ("myUsername".equals(pc.getIdentity())) {
            pc.setPassword("myPassword");
        } // else {... - can add more users, access DB, etc.
    }
}
```
6 Annexes

- Example of soap request used for tests in paragraph 4:

```xml
  <soapenv:Body>
    <ParkingTablePublication instance="2015-11-24T16:41:56.194010+00:00" version="166">
      <parkingLocation id="FR-282901512236386604809551631469246161" xmlns="http://datex2.eu/schema/2/2_0" xsi:type="InterUrbanParkingSite">
        <parkingName>
          <values lang="ENGLISH">Test parking site</values>
        </parkingName>
        <parkingDescription>
          <values lang="ENGLISH">Parking site for testing</values>
        </parkingDescription>
        <parkingNumberOfSpaces>100</parkingNumberOfSpaces>
        <parkingTypeOfGroup>InterUrbanParkingSite</parkingTypeOfGroup>
        <parkingTableVersionTime>2015-11-24T16:41:56.194010+00:00</parkingTableVersionTime>
        <parkingRecord publicationTime="2015-11-24T16:41:56.194010+00:00" xmlns="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ContactDetails">
          <contactOrganisationName>
            <value lang="ENGLISH">Test organisation</value>
          </contactOrganisationName>
          <contactDetailsAddress>
            <values>
              <value lang="ENGLISH">Test address</value>
            </values>
          </contactDetailsAddress>
          <contactDetailsCity>
            <values>
              <value lang="ENGLISH">Test city</value>
            </values>
          </contactDetailsCity>
          <contactDetailsTelephoneNumber>
            <values>
              <value lang="ENGLISH">Test phone number</value>
            </values>
          </contactDetailsTelephoneNumber>
          <contactDetailsEmail>
            <values>
              <value lang="ENGLISH">Test email</value>
            </values>
          </contactDetailsEmail>
          <contactOrganisationName>
            <values>
              <value lang="ENGLISH">Test organisation</value>
            </values>
          </contactOrganisationName>
          <contactDetailsTelephoneNumber>
            <values>
              <value lang="ENGLISH">Test phone number</value>
            </values>
          </contactDetailsTelephoneNumber>
        </parkingRecord>
      </parkingLocation>
      <parkingLocation id="FR-282901512236386604809551631469246161" xmlns="http://datex2.eu/schema/2/2_0" xsi:type="InterUrbanParkingSite">
        <parkingName>
          <values lang="ENGLISH">Test parking site</values>
        </parkingName>
        <parkingDescription>
          <values lang="ENGLISH">Parking site for testing</values>
        </parkingDescription>
        <parkingNumberOfSpaces>100</parkingNumberOfSpaces>
        <parkingTypeOfGroup>InterUrbanParkingSite</parkingTypeOfGroup>
        <parkingTableVersionTime>2015-11-24T16:41:56.194010+00:00</parkingTableVersionTime>
        <parkingRecord publicationTime="2015-11-24T16:41:56.194010+00:00" xmlns="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ContactDetails">
          <contactOrganisationName>
            <values>
              <value lang="ENGLISH">Test organisation</value>
            </values>
          </contactOrganisationName>
          <contactDetailsAddress>
            <values>
              <value lang="ENGLISH">Test address</value>
            </values>
          </contactDetailsAddress>
          <contactDetailsCity>
            <values>
              <value lang="ENGLISH">Test city</value>
            </values>
          </contactDetailsCity>
          <contactDetailsTelephoneNumber>
            <values>
              <value lang="ENGLISH">Test phone number</value>
            </values>
          </contactDetailsTelephoneNumber>
          <contactDetailsEmail>
            <values>
              <value lang="ENGLISH">Test email</value>
            </values>
          </contactDetailsEmail>
          <contactOrganisationName>
            <values>
              <value lang="ENGLISH">Test organisation</value>
            </values>
          </contactOrganisationName>
          <contactDetailsTelephoneNumber>
            <values>
              <value lang="ENGLISH">Test phone number</value>
            </values>
          </contactDetailsTelephoneNumber>
        </parkingRecord>
      </parkingLocation>
    </ParkingTablePublication>
  </soapenv:Body>
</soapenv:Envelope>
```
Web Service description to push static DATEXII data related to safe and secure Truck Parking Areas to the European Access point

```xml
  <soap:Body>
    <d2LogicalModel:d2ParkingPublication>
      <d2:parkingTable>
        <d2:parkingRecord>
          <d2:parkingAccess>
            <d2:primaryRoad>
              <d2:location xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Point">
                <d2:pointByCoordinates>
                  <d2:pointCoordinates>
                    <d2:latitude>50.83695</d2:latitude>
                    <d2:longitude>4.388394</d2:longitude>
                  </d2:pointCoordinates>
              </d2:pointByCoordinates>
            </d2:location>
          </d2:parkingAccess>
          <d2:parkingStandardsAndSecurity>
            <d2:labelSecurityLevel>none</d2:labelSecurityLevel>
            <d2:labelServiceLevel>none</d2:labelServiceLevel>
            <d2:labelSecurityLevelSelfAssessment>unknown</d2:labelSecurityLevelSelfAssessment>
            <d2:labelServiceLevelSelfAssessment>unknown</d2:labelServiceLevelSelfAssessment>
            <d2:parkingSecurity>none</d2:parkingSecurity>
            <d2:parkingAdditionalSecurity>
              <d2:values>
                <d2:value lang="ENGLISH"></d2:value>
              </d2:values>
            </d2:parkingAdditionalSecurity>
            <d2:parkingSupervision>none</d2:parkingSupervision>
            <d2:parkingSecurityNationalClassification>
              <d2:values>
                <d2:value lang="ENGLISH"></d2:value>
              </d2:values>
            </d2:parkingSecurityNationalClassification>
            <d2:interUrbanParkingSiteLocation>motorway</d2:interUrbanParkingSiteLocation>
          </d2:parkingStandardsAndSecurity>
        </d2:parkingRecord>
      </d2:parkingTable>
    </d2LogicalModel:d2ParkingPublication>
  </soap:Body>
</soap:Envelope>
```