



CENTRICO

Special Working Group 4

MIP2004+

DATEX II Low Cost profile exchange specification

Deliverable D5-3

Document version:	1.0
Status:	Final
Document nature:	Technical Report (Deliverable)
Dissemination level:	SWG4
Date of preparation:	2006-11-20
Activity:	Activity 5.3: Tailor DATEX II exchange options
Author(s):	Centrico SWG4
Editor:	J. Kaltwasser
Contact person:	
Name	J. Kaltwasser
Function	SWG4 chairman
Organisation	Heusch / Boesefeldt GmbH
Address	Tempelhofer Str. 4-6 Aachen
Postcode	52068
Tel	+49 241 9669 203
Mobile	+49 172 9031112
Fax	+49 241 9669 177
e-mail	e-mail: Josef.Kaltwasser@heuboe.de

Abstract: This report aims at tailoring the exchange options provided by the abstract DATEX II exchange model for a DATEX II Low Cost Profile based on the HTTP protocol, which the Centrico SC has tasked SWG4 to develop as part of their 2005 work plan.

Keywords: ITS, Centrico, AD2, Traffic Centres, TICS, Data Exchange



Table of Contents

1	<i>Executive Summary</i>	3
2	<i>Introduction</i>	4
3	<i>Normative References</i>	5
4	<i>General exchange</i>	6
5	<i>Use of HTTP (normative)</i>	8
5.1	Basic request / response pattern	8
5.2	Authentication	12
6	<i>Describing payload and interfaces (normative)</i>	14
7	<i>Metadata for link management (normative)</i>	15
8	<i>Appendix A: Glossary</i>	21

1 Executive Summary (informative)

DATEX had been developed in the nineties as a European standard for centre to centre exchange in ITS. A wider take-up of DATEX had been prevented by two major obstacles:

- Weak parts of the DATEX specifications left room for interpretations by the developers and thus led to interoperability problems.
- DATEX was built on early nineties technology and thus failed to find acceptance by a wider developer community after the Internet boom, that wanted a standard based on up-to-date technology.

Research projects in the late nineties showed up a path for migration of DATEX towards today's ICT standards, and the European Commission launched the D2 project to elaborate on this after successful validation of these principles in the scope of Centrico's Open Travel data Access Protocol (OTAP) initiative.

The D2 project created an initial specification that was taken up and continuously improved by the DATEX Technical Committee (TC). The new *DATEX II* specification is mainly characterised by:

- A clear split between traffic engineering content and communication related specifications around that, which allow exchanging this (serialised) content.
- The application of model driven architecture to separate abstract and persistent specifications from platform dependent implementation with potentially much lower lifetime expectations.

The latter principle also has the advantage that the same – and thus basically interoperable – abstract specification can be mapped to more than one implementation platforms.

Whilst at this moment all stakeholders interested in DATEX II seem to agree on a single target platform for content encoding – XML plus XML schema definitions (XSD) – there are at least two instances of exchange target platforms under discussion, named as *Low Cost Profile* and *Regular Profile*. While the latter targets sophisticated exchange scenarios with complex interfaces and services, the former aims at providing a minimum entry threshold interface into DATEX technology (essentially as easy as setting up a website). Both profiles have successfully been demonstrated to the public at the i2tern conference in June 2006 in Barcelona, and both profiles have been proven to provide basic interoperability, thus actually confirming the benefit claims of the DATEX' overall model driven methodology approach!

This document provides the technical specifications for the DATEX II Low Cost Profile that have been developed inside the Centrico project by Special Working Group 4 *Data Exchange Forum*. The document is fully self-contained, and although the document itself is not part of the official DATEX II documentation, its content has been fully incorporated into the DATEX II *Platform Specific Mapping* (PSM) specification. Implementers interested in implementing D2LCP only can use this document to implement software and feel confident that the result is interoperable with any other DATEX II system!



2 Introduction (informative)

The Directorate-General for Energy and Transport of the European Commission (DG-TREN) had commissioned the *D2* project to evolve the current CEN DATEX prestandards¹, a set of specifications for data exchange between traffic centres and from traffic centres to service providers. The project team faced many unexpected obstacles during this task, and at the end of the project duration in February 2005 not all deliverables were available in the expected form. Especially for the work package 5 on a *DATEX II Low Cost Profile* (D2LCP), only a high level guideline on how to proceed further could be delivered. In response to a request from DG-TREN, the Centrico Steering Committee has tasked its Special Working Group 4 (SWG4 – *Data Exchange Forum*) to take up this activity and to fully develop and support a trial for such a low cost profile.

This document contains results of this work. The D2LCP had been built primarily based on the input from an earlier Centrico project called *Open Travel data Access Protocol* (OTAP). OTAP is an initiative to stimulate access of public and private sector service providers to the real-time traffic and travel related information held in the road operators' databases in the Centrico area. It became clear very quickly that OTAP could not use DATEX with EDIFACT encoding for this purpose, and after reviewing initial work in this field (from the COURIER and TRIDENT research projects and from a DATEX Technical Committee analysis based on this) the choice was to code the DATEX content in XML (with an XML schema) and provide an HTTP/1.1 profile to provide a stateless client-pull access to this data. The outcome of this work has been substantially tested since 2003 in various operational services and has only slightly been adapted to cope with some extra requirements for DATEX II, in particular the interoperability with DATEX II Regular Profile suppliers / clients implemented with Web Services toolkits and using the SOAP protocol on top of HTTP.

This document – besides some informative information – provides essentially three normative sections for the D2LCP:

- One section stipulating the use of HTTP in the context of D2LCP systems.
- Another section providing rules that define the structure of D2LCP data feeds.
- And finally one section devoted to end-to-end acknowledgements, which are required in the case of intermediate Webservers being used to publish the payload data.

The D2LCP allows for active servers (e.g. Java servlet containers) as well as file based, standard WWW servers to publish data, and D2LCP compliant feeds can easily be accessed – for debug purposes – using any standard Web browser.

¹ CEN ENVs 13106:2000 and 13777:2000

3 Normative References (normative)

- [AUTH] J. Franks et. al., *HTTP Authentication: Basic and Digest Access Authentication*, RFC 2617, <http://www.ietf.org/rfc/rfc2616.txt>, June 1999
- [BASE64] N. Freed, N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, RFC2045, <http://www.ietf.org/rfc/rfc2045.txt>, November 1996
- [D5-1] Centrico SWG4, Deliverable D5-1, *DATEX II Low Cost Profile Requirements*, version 1.0, 4 May 2005
- [HTTP] R. Fielding et. al., *Hypertext Transfer Protocol -- HTTP/1.1*, RFC 2616, <http://www.ietf.org/rfc/rfc2616.txt>, June 1999
- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, RFC 2119, <http://www.ietf.org/rfc/rfc2119.txt>, March 1997
- [URI] T. Berners-Lee et. al., *Uniform Resource Identifiers (URI): Generic Syntax*, RFC 2396, <http://www.ietf.org/rfc/rfc2396.txt>, August 1998
- [UTF-8] F. Yergeau, *UTF-8, a transformation format of ISO 10646*, RFC 2279, <http://www.ietf.org/rfc/rfc2279.txt>, January 1998
- [XML] T. Bray et. al., *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation, <http://www.w3.org/TR/2004/REC-xml-20040204>, 4 February 2004

4 General exchange (informative)

The data access interface of the D2LCP is based on stateless client/server (C/S) communication, or – more precisely – on using the most widely spread transport mechanism for this type of interaction, the HTTP/1.1 protocol as specified in RFC 2616 [HTTP]. In reality, there will be a whole stack of protocols below DATEX II, i.e. most probably HTTP will run on top of TCP, which runs on top of IP, which runs on top of a bearer specific Link Layer protocol. Nevertheless, DATEX II will neither know about nor depend upon these lower layers. The important assumption is that the HTTP services are available, and DATEX II will define its interaction with the outside world via the invocation of HTTP services².

This assumption has two main implications on DATEX II LCP:

- Being able to perform a successful HTTP Request / Response dialog initiated by the client is the main mandatory prerequisite for DATEX II LCP data exchange.
- DATEX II data exchange can benefit from any standard security mechanism that is transparent from the perspective of invoking HTTP services, opening up a wide range of security support, from the simple (SSH or SSL/TLS) up to the most sophisticated (IPsec). Specification of the use and configuration of these mechanisms is beyond the scope of this document, although the DATEX II community expresses its will to support users with suitable guideline documentation for this task.

It should be noted that although the restriction to stateless C/S solution will not be shared with the DATEX II Regular Profile (D2RP) – which will use WSDL/SOAP based, potentially stateful peer-to-peer communication instead – the HTTP transport constitutes the common platform for both profiles. This commonality is also the foundation for cross-profile interoperability between all DATEX II systems!

This specification has been developed based on the exchange requirements for the DATEX II Low Cost Profile, as specified in the according deliverable of the Centrico SWG4 [D5-1].

The next sections should be understood as providing mainly an HTTP/1.1 profile, plus some additional regulations that provide essential link management functionality.

² Quote from RFC 2616: “*HTTP communication usually takes place over TCP/IP connections. The default port is TCP 80 [19], but other ports can be used. This does not preclude HTTP from being implemented on top of any other protocol on the Internet, or on other networks. HTTP only presumes a reliable transport; any protocol that provides such guarantees can be used; the mapping of the HTTP/1.1 request and response structures onto the transport data units of the protocol in question is outside the scope of this specification.*”

Section 5 stipulates the use of HTTP/1.1 for DATEX II LCP communication. Section 6 describes how individual DATEX II publications are described and made known to potential clients. Finally, section 7 describes the additional metadata (beyond the payload) that is required to establish a link and information management that suits the D2LCP's requirements.

Important Note:

The normative sections of this specification are marked up by the term “(normative)” in their heading. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119, see <http://www.ietf.org/rfc/rfc2119.txt> ([RFC2119]).

5 Use of HTTP (normative)

D2LCP exchange uses HTTP Request (GET or POST) / Response dialogs to convey payload and status data from the supplier to the client. Note though that D2LCP supports POST requests only for interoperability to other request/response exchange systems, in particular the DATEX II Regular profile based on SOAP over HTTP exchange. D2LCP itself does not process the information potentially included in the body of an HTTP POST request!

This section stipulates how to use HTTP options in the context of the D2LCP.

5.1 Basic request / response pattern

[C.1] Suppliers and clients SHALL use the HTTP/1.1 protocol for D2LCP access. D2LCP clients and suppliers SHALL fully comply with the HTTP/1.1 protocol specification in RFC 2616, as of June 1999.

This protocol is essentially based upon a request / response pattern, where the request can take one of several possible forms, among them the GET and POST methods for retrieving data. The GET and POST differ essentially in how the parameters of a request can be conveyed to the supplier. While these parameters are conveyed as part of the URL in the HTTP GET request, the POST request allows specifying an “entity” (i.e. a message body) that contains these parameters, thus enabling less restricted parameters. POST requests were originally intended for server upload functionality.

As the D2LCP specification foresees no complex request parameters, D2LCP prefers that HTTP GET requests are used. Since other exchange systems sometimes require HTTP POST requests, D2LCP also accepts these requests. Nevertheless, it is not the intention of D2LCP to establish another exchange protocol layer on top of HTTP, and thus the D2LCP systems are not obliged to process the content of the body of an HTTP POST request.

NOTE: D2LCP systems MAY not process the body of HTTP POST requests!

[C.2] Clients SHALL use the HTTP GET or POST method of the HTTP REQUEST message to request data from the supplier

The HTTP GET or POST request is served by the supplier by generating an HTTP response message. The payload data – if any – conveyed in this response is passed in the entity-body. The payload data has to conform to the DATEX II data serialisation rules. The exact structure of the DATEX II content payload is beyond the scope of this specification! It should be noted though that for interoperability reasons (in particular with DATEX II Regular profile clients that require a SOAP wrapper around the XML payload) the D2LCP does not stipulate the DATEX II content payload to be the root element of the XML content. It only requires the existence of **exactly one** DATEX II content payload instance to be available as a subtree of the whole XML content tree.

[C.3] Suppliers SHALL use an HTTP RESPONSE message to respond to requests. If applicable, payload data according to DATEX II payload encoding specifications is contained in the entity-body. The DATEX II payload element MAY be embedded into other XML elements ('wrapper') in the contained XML document, but this XML document MUST contain exactly one DATEX II payload element.

[C.4] D2LCP suppliers SHALL NOT respond to HTTP REQUEST messages using the GET or POST methods by responding with 405 (Method Not Allowed) or 501 (Not Implemented) return codes.

D2LCP is following a client-pull, polling paradigm, as all communication is initiated by the client. Any data flow from supplier to client can only happen as the supplier's response to a client's request. When requesting data, the client is not able to know whether the data he would receive would be exactly the same as the one he had received in response to his last request. This would lead to a serious amount of redundant network traffic, with potential undesired impact on communication charges and supplier/client work load. HTTP supports avoiding this by letting the client specify the modification time of the last received update of a resource in an HTTP header field (If-Modified-Since). If no newer data is available, the response message will consist only of a header without an entity, stating that no new data is available. Clients are therefore recommended to set this header field in case they already hold reasonably recent information from the accessed URL.

[C.5] Suppliers MUST set the 'Last-Modified' header field in HTTP RESPONSE messages that provide payload data (response code 200) to the value that the information product behind the URL was last updated.

[C.6] Clients SHOULD set the 'If-Modified-Since' header field in all HTTP REQUEST messages if they already hold a consistent set of data from a particular URL in their database and the last modification time of that data is known from the 'Last-Modified' header field of the HTTP header of the HTTP RESPONSE message within which the payload data was received.

It must be understood that the semantics of the timestamps used within the If-Modified-Since header field are calculated in the server. Therefore, times generated by the client according to his own system clock may not be used here but must be filled using the content of the Last-Modified header field of the most recently received HTTP RESPONSE message. If clients connect to a resource for the first time or want to resynchronise, they simply don't set this header field.

[C.7] When setting the 'If-Modified-Since' header field, the client SHALL copy the value of the Last-Modified header field received within the last



successful HTTP RESPONSE containing payload (response code 200) message into this field.

[C.8] D2LCP suppliers SHOULD provide XML coded DATEX II payload as “text/xml” media type. D2LCP suppliers SHOULD state the used character set via the “charset” parameter; D2LCP suppliers SHOULD use the UTF-8 character set, i.e. the “Content-Type” response-header field SHOULD state “text/xml; charset=utf-8”.

Character sets, media types, etc. are a vast area that is notoriously underestimated as a source of potential interoperability problems. This clause aims at recommending the most widely used set of options, namely the use of the UTF-8 character set [UTF-8] for XML payload. The “text/xml” is preferred to “application/xml” following a recommendation in RFC 3023 (“If an XML document -- that is, the unprocessed, source XML document -- is readable by casual users, text/xml is preferable to application/xml.”). Although in principle D2LPC is solely devoted to B2B communication, readability of the exchange payload has often proved to be beneficial for testing and educational purposes.

It should be noted that omitting the “charset” attribute in HTTP/1.1 for “text/*” type content implies the use of “ISO-8859-1” which is different from the UTF family (UTF-8, UTF-16) that are the minimum requirement for XML processors (see [XML], section 4.3.3), and can thus be seen as the de-facto standard for XML. Consequently, sending typical XML payload like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<d2lm:situationPublication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="D2LogicalModel D2LogicalModel_TiS_48.3_ext3.xsd">
[...]
```

via HTTP/1.1 actually requires the use of the “charset” attribute:

```
HTTP/1.x 200 OK
[...]
```

Content-Type: text/xml; charset=UTF-8

[...]

If “text/xml” without parameter would be sent, HTTP/1.1 would be violated. (Quote from RFC 2616: “When no explicit charset parameter is provided by the sender, media subtypes of the “text” type are defined to have a default charset value of “ISO-8859-1” when received via HTTP. Data in character sets other than “ISO-8859-1” or its subsets MUST be labeled with an appropriate charset value.”)

[C.9] D2LCP clients MUST accept “identity” content-coding; D2LCP clients SHOULD (and if they do, prefer to) accept “gzip” content-coding; D2LCP clients MAY accept other “content-coding” values registered by the

Internet Assigned Numbers Authority (IANA) in their content-coding registry³ as long as they also accept “identity” and “gzip” content-coding.

[C.10] When including an “Accept-Encoding” request-header field in an HTTP REQUEST message, the client MUST NOT exclude acceptance of “identity” content-coding.

[C.11] D2LCP suppliers MUST provide “identity” content-coding of the payload; D2LCP suppliers SHOULD provide “gzip” content-coding of the payload; D2LCP suppliers MAY provide other “content-coding” values registered by the Internet Assigned Numbers Authority (IANA) in their content-coding registry as long as they also provide “identity” and “gzip” content-coding.

This set of clauses essentially ensures that a confused situation where the supplier is not able to provide payload in a content-coding that the client understands can not exist, as all suppliers are enforced to support “identity” (which in case of D2LCP means unmodified text/xml content) content-coding, and clients are enforced to understand this content-coding.

Furthermore, these clauses include a policy that recommends the use of compression and ensures that compression is always interoperable because it requires all clients/suppliers that do support compression to support “gzip” at least as an option. This means that:

- All client/supplier interaction will work at least with “identity”
- Clients/suppliers supporting compression will always be able to agree on “gzip”
- Clients can request preferred other compression (“deflate” or “compress”), and suppliers will respond accordingly if they support these content-codings.

Implementers should be aware that non-transparent web caches may perform media type transformations on behalf of their clients. Thus client running over such a cache might notice that compressed response content is automatically decompressed. However this is only problematic if there is a low bandwidth connection between the client and the cache, such as a dial up access point. If a D2LCP service has a significant number of such users then the addition of the no-transform directive to the Cache-Control header field of the generated responses should be considered. For more details on the use of the Cache-Control field, please consult Section 14.9 of RFC 2616.

[C.12] Servers MAY use the ‘no-transform’ directive in the ‘Cache-Control’ header field to avoid non-transparent caches from sending non-compressed content.

³ See www.iana.org

5.2 Authentication

D2LCP supports authentication, i.e. only users with explicit permission are allowed to download payload data. The required access credentials have to be provided to the client as part of the outcome of the DATEX II subscription creation with the content supplier, which is an offline process in D2LCP. The D2LCP specifications make use of the simplest and most widely spread authentication scheme for HTTP, i.e. BASIC authentication, as specified in [AUTH].

Note:

This scheme is in itself not seen as sufficiently strong for commercial strength business and safety relevant application, as the password is not encrypted during transmission. Applications that fit into this description will either have to use other DATEX II profiles (e.g. the DATEX II Regular Profile) or they will need to establish a sufficiently secure transport layer below DATEX II!

In essence, the client receives a user name and password together with a URL which identifies a specific publication from the server. During access, the client then builds a single string from this ("`<username>:<password>`") and encodes it according to base64 encoding rules (see [BASE64]). The result is put into the Authorization header field of the HTTP REQUEST message.

[C.13] Clients SHOULD fill access credentials they MAY have received during the subscription negotiation process into the 'Authorization' header field of the HTTP REQUEST message.

[C.14] Server providing access credentials (user name & password) during the subscription negotiation phase MAY respond with response code 401 (Unauthorized) to HTTP REQUESTS that do not contain valid access credentials in the 'Authorization' header field

The regulations in this and the previous section are a clarification on how to use standard features of HTTP/1.1 according to RFC2616 for D2LCP systems. The following sections contain additional regulations that go beyond 'pure' HTTP. Anyway, the regulations presented so far have to be seen as clarifications on top of RFC2616. They are compliant with the standard and have to be used in conjunction with the standard itself. This principle holds especially for the handling of HTTP return codes. The following clause summarises the main return codes as used in D2LCP connections and refers to the standard for the handling of further codes.

[C.15] D2LCP servers SHALL produce and D2LCP clients SHALL process the following return codes:

- 200 (OK), in responses carrying payload,***
- 304 (Not Modified), if no payload is send because of the specification in the 'If-Modified-Since' header,***



- **503 (Service Unavailable)**, if an active HTTP server is disconnected from the content feed,

- **404 (Not Found)**, if a file based HTTP server does not have a proper payload document stored in the place associated to the URL.

D2LCP servers SHOULD produce and D2LCP clients SHOULD process the following return codes:

- **401 (Unauthorised)**, if authentication is required but not presented in the request, or if invalid authentication is presented in the request,

- **403 (Forbidden)**, if the requested operation is not successful for any other reason.

D2LCP servers & client SHALL apply an RFC2616 compliant regime for producing / handling all other return code.

6 Describing payload and interfaces (normative)

D2LCP does not support server side client filtering! It should also be noted that although such a feature could in principle be incorporated into the D2LCP, it would require substantial processing power inside the server and also client specific information products, which would increase the implementation complexity on the server side substantially. Users aiming at applications based on server side client filtering are thus advised to consider using the DATEX II Regular Profile instead!

Nevertheless, it was decided that servers should (statically and equally for all clients) split their publications into different ‘information products’ (e.g.. roadwork information, incidents, x-urgent messages). Information products SHOULD be used to split the different DATEX II publications, but can go further in adding particular filters on attributes, locations, etc. In particular, this allows consideration to be given to different amounts of data and typical update cycles / latency requirements for different types of data. An x-urgent message information product will probably hardly contain more than one or two situation elements at a time but clients may want to poll this information product at high frequency. Roadwork information – on the other hand – will probably only change once per day but may contain hundreds of situation records. Even if D2LCP provides various mechanisms to reduce redundant downloads, some clients may decide that updating this data every 2 hours is sufficient.

D2LCP handles information products by assigning a specific URL (potentially plus access credentials) per information product. The information product itself is denoted by all but one path segments in the URL, while the ‘filename’ (i.e. the middle path segment) is “content.xml” per definition. This convention was introduced to allow the definition of related meta-data for this information product in other files in the same directory.

***[C.16] Clause 13: Payload data for D2LCP Information products SHALL be denoted by a URL according to the following convention:
d2lcp_infop = "http://" host [":" port] infop_path "/content.xml" ["?" query]
where “infop_path” is a “path” component as specified in section 3.3 of [URI], but excluding the last path segment.***

The end of this clause may sound awkward. It strives at maintaining all the regulations of the URI RFC, thus not constraining URLs for D2LCP information products, but incorporating the need to have the final path element (the ‘filename’) being “content.xml” by convention.

To support authentication, the servers have to provide the credentials required to perform authentication for any particular information product.

[C.17] Clause 14: Server requiring authentication MUST provide the required access credentials for BASIC authentication (i.e. user name & password) together with the URL for the Information Product.



7 Metadata for link management (normative)

Scenarios exist where the If-Modified-Since mechanism introduced earlier is not preferred to avoid redundant downloads. In particular, this will happen if the server provides information products by updating files on a standard, file-based WWW server (like Apache or Microsoft Internet Information Server). In this option, the server would have two possible update regimes:

1. Periodical update of the information product's payload file independent from any changes in the data.
2. Update the information product's payload file on demand, when changes relevant to this product have occurred in the server's database.

With updating regime 1, the file based WWW server would cyclically send the file content to the clients, as he will derive the Last-Modified value from the file modification times. So the client would receive redundant downloads!

[C.18] D2LCP implementation based upon standard WWW servers and files as information products SHOULD update information product payload files only when their traffic domain content has changed!

Now this will mean that the file may stay around unmodified for some time after an update. This regime has one serious drawback: the client will not be able to determine whether the file remains unchanged because the (road) traffic situation is stable or because the backend server system itself (not the WWW server!) is not operating properly. In fact, the response of the (intermediate) WWW system does not have the quality of an end-to-end acknowledgement.

This is one of the major problems of simple FTP based DATEX links, and certainly has to be avoided for D2LCP! Therefore, an additional mechanism is required for servers that build upon file based WWW servers that give additional explicit meta-information as an end-to-end acknowledgement.

This information is given in a small XML document that is periodically updated, even if the (potentially huge) content file is unchanged. The file is required to refer to an XML Schema that contains an Element called *MetaData* as root element with two required attributes of type *xsd:dateTime* called *confirmationTime* and *confirmedTime*, with *confirmationTime* giving the time the acknowledgement was created and *confirmedTime* giving a value equal to the value the Last-Modified header field would have if the payload file (i.e. *content.xml*) had been requested.

The following XML Schema gives a valid example:



```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!-- Our XML message consists of 1 element, the metadata record!-->
  <xsd:element name="MetaData" type="MetadataType"/>

  <xsd:complexType name="MetadataType">
    <xsd:attribute name="confirmationTime" type="xsd:dateTime" use="required"/>
    <xsd:attribute name="confirmedTime" type="xsd:dateTime" use="required"/>
  </xsd:complexType>
</xsd:schema>
```

The acknowledgement file shall be put in the information product's directory, besides the content.xml file containing the payload, with a filename of metadata.xml. A sample file for the schema above would be:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- D2LCP Demo File for Metadat -->
<MetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="metadata.xsd"
          confirmationTime="2005-05-19T09:40:22+02:00"
          confirmedTime="2005-05-19T09:32:22+02:00"
/>
```

This leads to the following specifications:

[C.19] D2LCP implementation based upon standard WWW servers and files as information products SHOULD provide a cyclically updated acknowledgement, accessible as
*D2LCP_infop_ack = "http:" "://" host [":" port] infop_path
"/metadata.xml" ["?" query]*

[C.20] If a D2LCP acknowledgement is provided, it SHALL be a well-formed XML document with a XML Schema reference. The acknowledgement SHALL validate successfully against the referenced XML Schema.

[C.21] The XML Schema referenced by an D2LCP acknowledgement SHALL contain a root element called "MetaData". This element SHALL contain two attributes, one named "confirmationTime" and one named "confirmedTime", both of type "xsd:dateTime".

[C.22] If used, D2LCP acknowledgements SHALL be updated cyclically, with best effort update cycle, but not less than once every three minutes.

[C.23] An acknowledgement update SHALL indicate that the data server is operating properly at the time it is generated and that the content of that



payload file – as last updated with modification time given in the “confirmedTime” attribute of the “MetaData” – element is currently still valid.

[C.24] The “confirmationTime” attribute of the “MetaData” element SHALL contain the current time when an acknowledgement is updated.

[C.25] The “confirmedTime” attribute of the “MetaData” element SHALL contain the same value that a HTTP RESPONSE to an authorised HTTP REQUEST issued at the time of writing the acknowledgement would contain in the “Last-Modified” header field.

[C.26] A D2LCP server that is based upon standard WWW servers and files as information products SHALL indicate in the D2LCP subscription negotiation process whether the acknowledgement option is supported.

[C.27] D2LCP clients SHOULD use the acknowledgement option – if provided – to determine whether payload download is required.

8 Requirements Tracking (informative)

The requirements document [D5-1] was structured in a way that first the high level needs for various data exchange related viewpoints were discussed, these were then compared to the features of OTAP exchange and finally proposals were derived for D2LCP. This section seeks to ensure that these proposals are actually met with the current D2LCP exchange specification as captured in this document.

8.1 General D2LCP Requirements / high level needs

The proposals to match the requirements elicited in this section were provided in section 3.1.3 of this document. The main advice was to stick to OTAP exchange as far as possible. This has been achieved by actually copying the OTAP exchange specification and modifying them only where either

- they appeared to be wrong or incomplete, or
- where additional features had to be introduced for the sake of interoperability with D2RP.

An example of the former point is the clarification of handling content encoding and character set, a good example for the latter principle is the explicit support for an XML wrapper around the pure payload data in [C.3], which was required for proper interoperable exchange with Web Services clients.

The amount of required changes / amendments were very limited and do not break backward compatibility on the server side, i.e. any compliant implementation of a server-side OTAP exchange system is still a valid D2LCP implementation, although they will not be interoperable with D2RP clients if they don't provide a SOAP wrapper. Client side OTAP can not be guaranteed to be compliant since the OTAP specification did not force them to allow for XML wrapper code around the payload, but resources required to adapt existing software should be at very low levels.

8.2 Specific Technical Requirements / Security

Section 3.2.1.3 of the requirement document proposes to separate security issues from the 'ITS aware' part of DATEX specifications. The D2LCP follows this guidance by actually not integrating security features beyond the very minimum support for BASIC authentication following RFC2617. All additional features should – if required – be implemented on transport layer or below (e.g. by using TLS or IPsec) and would thus be beyond the scope of the D2LCP specification. The required negotiations (e.g. to share keys) would then be added to the process that provides the client with access credentials to the required information product. The approach was recently supported by the DATEX TC that has actually stripped security related aspects from the DATEX II exchange PIM, following exactly the same line of argument!

8.3 Specific Technical Requirements / Communication Characteristics

Again, section 3.2.2.3 of the D2LCP requirements document suggests sticking closely to the OTAP protocol regulations based on the positive evaluation results for OTAP feeds. One single area of attention and potential change was identified: the Information Management Layer that – at the time when the requirement document was created – had not been populated in DATEX II. Meanwhile clarification has been provided by the DATEX Technical Committee that all information management related information has been centrally captured in a *Management* package on the top level in the *D2LogicalModel*. It was further clarified that special information management metadata may only be required in server-push exchanges, thus formerly opening up for D2LCP to retain the OTAP management model for *SituationPublication* data, which means that:

1. A *SituationPublication* contains **all** currently valid *SituationRecord* entities that belong to a particular information product. Note that the term *valid* here is not related to the concept of *ValidityPeriod* introduced recently into DATEX II, i.e. a roadwork *SituationRecord* can very well be valid and part of an information product even if the *validityPeriod* specifies that it is currently inactive!
2. An entity received in a payload download with an identity not known yet to the client is considered to be **new**.
3. An entity in a payload download and already known to the client – but with a higher *situationRecordVersion* value than the known one – is considered to be **updated**.
4. An entity known to the client so far but no longer present in a payload download is considered to be **ended**.

Note that the *situationRecordVersion* attribute is mandatory in DATEX II!

Other publications in the current version of the DATEX II model do contain only static entities that do not require online lifecycle management.

One issue where D2LCP goes beyond OTAP is that the use of other compression methods for content encoding are allowed now, given that client and supplier can agree on such a commonly supported method. GZIP is still mandatory and remains the common interoperability layer for all clients/suppliers that do support compression.

Furthermore, a major clarification on the proper use of media types and character sets has been added to remove some non compliant use of these HTTP mechanisms observed in some OTAP feeds.

8.4 Conclusion

D2LCP strictly follows the recommendations given in [D5-1] to retain OTAP exchange rules where possible, while only having to add very few regulations to improve the specifications and to enable cross-profile interoperability with D2RP. Fully compliant OTAP supplier implementations remain compatible and clients either are already compliant or can be modified with minimum effort. Thus, the positive evaluation results for OTAP can be conveyed to D2LCP services and migration efforts are low. The D2LCP demonstration in June 2006 actually partially relied on a simple XSLT translation service that on-the-fly

transformed OTAP feeds into D2LCP feeds. As part of another work package, Special Working Group 4 is working on packaging this converter into a package that can easily be deployed inside Centrico centres providing OTAP feeds, thus effectively providing Centrico OTAP users with a zero cost migration path from OTAP to DATEX II!



9 Appendix A: Glossary

AD2	Activity Domain 2: European <i>Network of Traffic Centres</i> . One of the activity areas in the Centrico project.
B2B	Business-to-business communication (as opposed to B2C – business-to-customer) is the machine-to-machine communication between two businesses (e.g. traffic centres). In ITS, the term centre-to-centre communication (C2C) is also established.
C/S	Client / Server computing – a paradigm for distributed computing that is widely used on the Internet today.
CEN	European Committee for Standardization
D2	A study commissioned by DG-TREN to evolve DATEX
D2LCP	DATEX II Low Cost Profile – a profile that provides a suitable DATEX II subset at minimal total cost of ownership.
D2RP	DATEX II Regular Profile - a profile that provides full centre to centre functionality, sufficiently strong to support tight coupling of centres as required e.g. on national level in France or Italy.
DATEX	European ITS standard for centre-to-centre data exchange. CEN ENVs 13106:2000 and 13777:2000.
DATEX II	A set of specifications developed under the auspices of DG-TREN that is supposed to replace the existing DATEX pre-standards as the major exchange specification between traffic centres on the TERN.
DG-TREN	Directorate-General Energy and Transport of the European Commission
ENV	European Prestandard (today replace by TS – Technical Specification)
FTP	File Transfer Protocol, RFC 959
HTTP	Hypertext Transfer Protocol (IETF RFC 2616).
ICT	Information and Communication Technologies.
IETF	Internet Engineering Task Force
IP	Internet Protocol, RFC 791
IPsec	A standard for securing Internet Protocol (IP) communications on network layer by encrypting and authenticating all IP packets.
ITS	Intelligent Transport Systems
OTAP	Open Travel data Access Protocol (see www.itsproj.com/otap)

PIM	Platform Independent Model – a notion from the model driven architecture paradigm providing a system description that is abstract and not depending on any implementation aspects.
PKI	Public Key Infrastructure – an infrastructure required to provide certified access to organisations’ public keys, required for asymmetric encryption.
RFC	Request For Comment – the established specification mechanism of the Internet Engineering Task Force.
PSM	Platform Specific Model - a notion from the model driven architecture paradigm providing a concrete implementation (‘mapping’) of a platform independent model on a particular implementation platform.
SSH	Secure Shell – an early but still widely used standards, mainly for remote access but also providing port-forwarding features that allow secure connections between systems.
SSL	Secure Socket Layer – a standard developed by the Netscape company to secure transmission of private documents via the Internet.
SOAP	Simple Object Access Protocol – one of the W3C base standards for Web Services
SWG4	<i>(Special Working Group 4)</i> Internal name of Centrico’s <i>Data Exchange Forum</i>
TCC	Traffic Control Centre.
TCP	Transmission Control Protocol, RFC 793
TERN	The Trans European Road Network.
TICS	Traffic Information and Control System
TLS	Transport Layer Security – an non-proprietary successor of SSL, developed by the IETF (RFC 2246).
TTI	Traffic and Travel Information.
UML	Unified Modelling Language – a de-facto standard for system engineering, meanwhile standardised by the OMG (www.omg.org).
URI	Universal Resource Indicator – the global address of any resource on the World Wide Web (RFC 2396).
URL	Universal Resource Locator – an URI that acts as a unique resource address on the Internet.
Web Service(s)	The programmatic interfaces developed for the use of the World Wide Web as a platform for B2B application-to-application.
WSDL	Web Services Definition Language – one of the W3C base standards for Web Services

WWW	World Wide Web – sometimes (incorrectly) used as a synonym for the Internet, the WWW is actually – in a technical sense - the part of the Internet that is accessible via HTTP.
XSD	XML Schema Definition – a W3C recommendation offering facilities for describing the structure and constraining the contents of XML 1.0 documents (see www.w3.org)
XML	Extensible Markup Language – according to the W3C, XML is a “very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.”