

INTERNET DRAFT

Donald E. Eastlake 3rd

Motorola Laboratories

Expires April 2005

October 2004

Electronic Commerce Modeling Language (ECML):

Version 2 Specification

[draft-ietf-trade-ecml2-spec-13.txt](mailto:ietf-trade-ecml2-spec-13.txt)

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, or will be disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

Distribution of this document is unlimited. Comments should be sent to the author or the IETF TRADE working group <ietf-trade@lists.elistx.com>.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than a "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

Copyright (C) The Internet Society 2004. All Rights Reserved.

Abstract

Electronic commerce frequently requires a substantial exchange of information in order to complete a purchase or other transaction, especially the first time the parties communicate. A standard set of hierarchically organized payment related information field names in an XML syntax are defined so that this task can be more easily automated. This is the second version of an Electronic Commerce Modeling Language (ECML) and is intended to meet the requirements of [RFC 3505](#).

Acknowledgements

The following, listed in alphabetic order, have contributed to the material herein:

Ray Bellis, Steve Bellovin, Scott Hollenbeck, Russ Housley, Jon Parsons, Lauri Piikivi, David Shepherd, and James J. Peter.

Table of Contents

Status of this Memo.....	1
Abstract.....	1
Acknowledgements.....	2
Table of Contents.....	2
1 . Introduction.....	3
1.2 History and Relationship to Other Standards.....	3
2 . Field Definitions, DTD, and Schema.....	4
2.1 Field List and Descriptions.....	4
2.1.1 The Field List.....	4
2.1.2 Field Foot Notes.....	8
2.2 Exemplar XML Syntax.....	13
2.2.1 ECML v2 XML DTD.....	13
2.2.2 ECML v2 XML Schema.....	19
3 . Usage Notes for ECML v2.....	27
3.1 Presentation of the Fields.....	27
3.2 Methods and Flow of Setting the Fields.....	27
4 . Security and Privacy Considerations.....	28
5 . IANA Considerations.....	29
5.1 ECML v2 Schema Template.....	29
5.2 ECML v2 URN Template.....	30
5.2.1 Subregistration of v2.0.....	30
5.3 IANA Registries.....	30
Normative References.....	32
Informative References.....	33
Appendix: Changes from v1.1 to v2.....	35
Copyright and Disclaimer.....	36
Authors Addresses.....	36
File name and Expiration.....	36

1. Introduction

Numerous parties are conducting business on the Internet using ad hoc fields and forms. The data formats and structure can vary considerably from one party to another. Where forms are filled out manually, some users find the diversity confusing and the process of manually filling in these forms can be tedious and error prone.

Software tools, including electronic wallets, can help this situation. Such tools can assist in conducting online transactions by storing billing, shipping, payment, preference, and similar information and using this information to automatically complete the data sets required by interactions. For example, software that fills out forms has been successfully built into browsers, as proxy servers, as helper applications to browsers, as stand-alone applications, as browser plug-ins, and as server-based applications. But the proliferation of more automated transactions software has been hampered by the lack of standards.

ECML (Electronic Commerce Modeling Language) provides a set of hierarchical payment oriented data structures that will enable automated software, including electronic wallets from multiple vendors, to supply and query for needed data in a more uniform manner.

Version 2.0 extends ECML Version 1.0 [[RFC 2706](#)] and 1.1 [[RFC 3106](#)] as described in the Appendix to this document. These enhancements include support for additional payment mechanisms and transaction information and use of XML as the exemplar syntax.

ECML is designed to provide a simple baseline useful in a variety of contexts. Likely uses for ECML v2 are consumer payment information input and business-to-business transactions. At this time, the first is still likely to occur through HTML forms. The second is more likely to use XML documents.

1.2 History and Relationship to Other Standards

The ECML fields were initially derived from the W3C P3P base data schema [P3P BASE] by the ECML Alliance as described in [RFC 2706, 3106]. Technical development and change control of ECML was then transferred to the IETF. In version 2, ECML is extended by the fields in a W3C P3P Note related to eCommerce [P3P ECOM], by [ISO 8583], and other sources. Its primary exemplar form is now an XML syntax.

2. Field Definitions, DTD, and Schema

ECML v2 is the definition and naming of a hierarchically structured set of fields and the provision of an optional XML syntax for their transmission. These fields can be encoded in other syntaxes. Regardless of the encoding used, they can be transmitted via a variety of protocols.

[Section 2.1](#) below lists and describes the fields, [Section 2.2.1](#) provides an XML DTD for use with the fields, and [Section 2.2.2](#) provides an XML schema.

To conform to this document, field names must be named and hierarchically structured as closely to the structure and naming listed below as practical given the syntax and transaction protocol in use. (NOTE: this does not impose any restriction on human visible labeling of fields, just on their name or names and structure as used in on-the-wire communication.)

2.1 Field List and Descriptions

The fields are listed below. along with the minimum data entry size to allow. Implementations may accept larger data sizes, where that makes sense, and, for some applications, will need to allow for larger data sizes.

Note that these fields are hierarchically organized as indicated in this table by the embedded underscore ("_") characters. Appropriate data transmission mechanisms may use this to request and send aggregates, such as Ecom_Payment_Card_ExpDate to encompass all of a set of card expiry date components or Ecom_ShipTo to encompass all the ship to address components that a consumer is willing to provide. The labeling, marshalling, unmarshalling of the components of such aggregates depends on the data transfer protocol used. The suggested syntax is XML as specified in [section 2.2](#).

2.1.1 The Field List

The table below is the ECML v2 field list.

The NAME column gives the structured string name of each field as explained above. The MIN column below is the minimum data size that MUST be allowed for on data entry. It is NOT the minimum size for valid contents of the field and merchant software should, in many cases, be prepared to receive a longer or shorter value. Merchant

dealing with areas where, for example, the state/province name or

D. Eastlake 3rd

[Page 4]

phone number is longer than the MIN given below must obviously permit longer data entry. In some cases, however, there is a maximum size that makes sense and where this is the case, it is usually documented in a Note for the field.

The following fields are typically used to communicate from the customer to the merchant:

FIELD	NAME	MIN	Notes
ship to title	Ecom_ShipTo_Postal_Name_Prefix	4	(1)
ship to first name	Ecom_ShipTo_Postal_Name_First	15	(54)
ship to middle name	Ecom_ShipTo_Postal_Name_Middle	15	(2)
ship to last name	Ecom_ShipTo_Postal_Name_Last	15	(54)
ship to name suffix	Ecom_ShipTo_Postal_Name_Suffix	4	(3)
ship to company name	Ecom_ShipTo_Postal_Company	20	
ship to street line1	Ecom_ShipTo_Postal_Street_Line1	20	(4)
ship to street line2	Ecom_ShipTo_Postal_Street_Line2	20	(4)
ship to street line3	Ecom_ShipTo_Postal_Street_Line3	20	(4)
ship to city	Ecom_ShipTo_Postal_City	22	
ship to state/province	Ecom_ShipTo_Postal_StateProv	2	(5)
ship to zip/postal code	Ecom_ShipTo_Postal_PostalCode	14	(6)
ship to country	Ecom_ShipTo_Postal_CountryCode	2	(7)
ship to phone	Ecom_ShipTo_Telecom_Phone_Number	10	(8)
ship to email	Ecom_ShipTo_Online_Email	40	(9)
bill to title	Ecom_BillTo_Postal_Name_Prefix	4	(1)
bill to first name	Ecom_BillTo_Postal_Name_First	15	(54)
bill to middle name	Ecom_BillTo_Postal_Name_Middle	15	(2)
bill to last name	Ecom_BillTo_Postal_Name_Last	15	(54)
bill to name suffix	Ecom_BillTo_Postal_Name_Suffix	4	(3)
bill to company name	Ecom_BillTo_Postal_Company	20	
bill to street line1	Ecom_BillTo_Postal_Street_Line1	20	(4)
bill to street line2	Ecom_BillTo_Postal_Street_Line2	20	(4)
bill to street line3	Ecom_BillTo_Postal_Street_Line3	20	(4)
bill to city	Ecom_BillTo_Postal_City	22	
bill to state/province	Ecom_BillTo_Postal_StateProv	2	(5)
bill to zip/postal code	Ecom_BillTo_Postal_PostalCode	14	(6)
bill to country	Ecom_BillTo_Postal_CountryCode	2	(7)
bill to phone	Ecom_BillTo_Telecom_Phone_Number	10	(8)
bill to email	Ecom_BillTo_Online_Email	40	(9)
receipt to			(32)
receipt to title	Ecom_ReceiptTo_Postal_Name_Prefix	4	(1)
receipt to first name	Ecom_ReceiptTo_Postal_Name_First	15	(54)
receipt to middle name	Ecom_ReceiptTo_Postal_Name_Middle	15	(2)
receipt to last name	Ecom_ReceiptTo_Postal_Name_Last	15	(54)
receipt to name suffix	Ecom_ReceiptTo_Postal_Name_Suffix	4	(3)

receipt to company name	Ecom_ReceiptTo_Postal_Company	20	
receipt to street line1	Ecom_ReceiptTo_Postal_Street_Line1	20	(4)

D. Eastlake 3rd

[Page 5]

receipt to street line2	Ecom_ReceiptTo_Postal_Street_Line2	20	(4)
receipt to street line3	Ecom_ReceiptTo_Postal_Street_Line3	20	(4)
receipt to city	Ecom_ReceiptTo_Postal_City	22	
receipt to state/province	Ecom_ReceiptTo_Postal_StateProv	2	(5)
receipt to postal code	Ecom_ReceiptTo_Postal_PostalCode	14	(6)
receipt to country	Ecom_ReceiptTo_Postal_CountryCode	2	(7)
receipt to phone	Ecom_ReceiptTo_Telecom_Phone_Number	10	(8)
receipt to email	Ecom_ReceiptTo_Online_Email	40	(9)
name on card	Ecom_Payment_Card_Name	30	(10)
card type	Ecom_Payment_Card_Type	4	(11)
card number	Ecom_Payment_Card_Number	19	(12)
card verification value	Ecom_Payment_Card_Verification	4	(13)
card issuer number	Ecom_Payment_Card_IssueNumber	2	(53)
card expire date day	Ecom_Payment_Card_ExpDate_Day	2	(14)
card expire date month	Ecom_Payment_Card_ExpDate_Month	2	(15)
card expire date year	Ecom_Payment_Card_ExpDate_Year	4	(16)
card valid date day	Ecom_Payment_Card_ValidFrom_Day	2	(14)
card valid date month	Ecom_Payment_Card_ValidFrom_Month	2	(15)
card valid date year	Ecom_Payment_Card_ValidFrom_Year	4	(16)
card protocols	Ecom_Payment_Card_Protocol	20	(17)
loyalty card name	Ecom_Loyalty_Card_Name	30	(10)
loyalty card type	Ecom_Loyalty_Card_Type	20	(52)
loyalty card number	Ecom_Loyalty_Card_Number	40	(34)
loyalty card verification	Ecom_Loyalty_Card_Verification	4	(13)
loyalty card expire day	Ecom_Loyalty_Card_ExpDate_Day	2	(14)
loyalty card expire month	Ecom_Loyalty_Card_ExpDate_Month	2	(15)
loyalty card expire year	Ecom_Loyalty_Card_ExpDate_Year	2	(16)
loyalty card valid day	Ecom_Loyalty_Card_ValidFrom_Day	2	(14)
loyalty card valid month	Ecom_Loyalty_Card_ValidFrom_Month	2	(15)
loyalty card valid year	Ecom_Loyalty_Card_ValidFrom_Year	4	(16)
consumer order ID	Ecom_ConsumerOrderID	20	(18)
user ID	Ecom_User_ID	40	(19)
user password	Ecom_User_Password	20	(19)
user certificate	Ecom_User_Certificate_URL	128	(55)
user data country	Ecom_UserData_Country	2	(7)
user data language	Ecom_UserData_Language	30	(33)
user data gender	Ecom_UserData_Gender	1	(36)
user data birth day	Ecom_UserData_BirthDate_Day	2	(14)
user data birth month	Ecom_UserData_BirthDate_Month	2	(15)
user data birth year	Ecom_UserData_BirthDate_Year	4	(16)
user data preferences	Ecom_UserData_Preferences	60	(34)

schema version	Ecom_SchemaVersion	30	(20)
wallet id	Ecom_WalletID	40	(21)
wallet URL	Ecom_Wallet_Location	128	(35)
customer device ID	Ecom_Device_ID	20	(37)
customer device type	Ecom_Device_Type	20	(38)
end transaction flag	Ecom_TransactionComplete	-	(22)

The following fields are typically used to communicate from the merchant to the consumer:

FIELD	NAME	Min	Notes
merchant home domain	Ecom_Merchant	128	(23)
processor home domain	Ecom_Processor	128	(24)
transaction identifier	Ecom_Transaction_ID	128	(25)
transaction URL inquiry	Ecom_Transaction_Inquiry	500	(26)
transaction amount	Ecom_Transaction_Amount	128	(27)
transaction currency	Ecom_Transaction_CurrencyCode	3	(28)
transaction date	Ecom_Transaction_Date	80	(29)
transaction type	Ecom_Transaction_Type	24	(30)
transaction signature	Ecom_Transaction_Signature	160	(31)
end transaction flag	Ecom_TransactionComplete	-	(22)

The following fields are used to communicate between the merchant and a processor acting for the merchant (such a processor is commonly called an acquirer and is frequently a bank):

FIELD	NAME	Min	Notes
merchant identifier	Ecom_Merchant_ID	8	
merchant terminal	Ecom_Merchant_Terminal_ID	8	(39)
merchant terminal data	Ecom_Merchant_Terminal_Data	128	
transaction process code	Ecom_Transaction_ProcessingCode	6	(40)
transaction reference	Ecom_Transaction_Reference_ID	12	
transaction acquirer	Ecom_Transaction_Acquire_ID	13	(41)
transaction forward	Ecom_Transaction_Forward_ID	13	(42)
transaction trace	Ecom_Transaction_Trace_Audit	6	(43)
transaction effective date	Ecom_Transaction_Effective_Date	4	(44)
transaction CID	Ecom_Transaction_CID	8	
transaction POS	Ecom_Transaction_POSCode	12	(45)
transaction private use	Ecom_Transaction_PrivateUseData	166	
transaction response	Ecom_Transaction_ResponseData	27	
transaction approval code	Ecom_Transaction_ApprovalCode	12	(46)
transaction retrieval code	Ecom_Transaction_RetrievalCode	128	

transaction response action Ecom_Transaction_ActionCode

13 (47)

D. Eastlake 3rd

[Page 7]

transaction reason	Ecom_Transaction_ReasonCode	4	
transaction AAV	Ecom_Transaction_AAV	3	
transaction settlement date	Ecom_Transaction_Settle_Date	4	(48)
transaction capture date	Ecom_Transaction_Capture_Date	4	(49)
transaction Track 1	Ecom_Transaction_Track1	39	(50)
transaction Track 2	Ecom_Transaction_Track2	39	(51)

2.1.2 Field Foot Notes

- (1) For example: Mr., Mrs., Ms., Dr. This field is commonly omitted.
- (2) May also be used for middle initial.
- (3) For example: Ph.D., Jr. (Junior), 3rd, Esq. (Esquire). This field is commonly omitted.
- (4) Address lines must be filled in the order line1, then line2, and last line3. Thus, for example, it is an error for line1 to be null if lines2 or line3 is not.
- (5) 2 characters are the minimum for the US and Canada, other countries may require longer fields. For the US use 2 character US Postal state abbreviation.
- (6) Minimum field lengths for Postal Code will vary based on international market served. Use 5 character or 5+4 ZIP for the US and 6 character postal code for Canada. The size given, 14, is believed to be the maximum required anywhere in the world.
- (7) Use [ISO 3166] standard two letter country codes.
- (8) 10 digits are the minimum for numbers within the North American Numbering Plan (<<http://www.nanpa.com>>: US, Canada and a number of Caribbean and smaller Pacific nations (but not Cuba)), other countries may require longer fields. Telephone numbers are complicated by differing international access codes, variant punctuation of area/city codes within countries, etc. While it is desirable for telephone numbers to be in standard international format [[E.164](#)], it may be necessary to use heuristics or human examination based on the telephone number and addresses given to figure out how to actually call a customer since people may enter local formatted numbers without area/access codes. It is recommend that an "x" be placed before extension numbers and that the "x" and extension number appear after all other parts of the number.
- (9) For example: jsmith@example.com

(10) The name of the cardholder as it appears on the card.

(11) Case insensitive. Use up to the first 4 letters of the association name (see also note 102):

AMER	American Express
BANK	Bankcard (Australia)
DC	DC (Japan)
DINE	Diners Club
DISC	Discover
JCB	JCB
MAST	Mastercard
NIKO	Nikos (Japan)
SAIS	Saison (Japan)
UC	UC (Japan)
UCAR	UCard (Taiwan)
VISA	Visa

(12) Includes the check digit at the end but no spaces or hyphens [ISO 7812]. The min given, 19, is the longest number permitted under the ISO standard.

(13) An additional cardholder verification number printed on the card (but not embossed or recorded on the magnetic stripe) such as the American Express CIV, MasterCard CVC2, and Visa CVV2 values.

(14) The day of the month. Values: 1-31. A leading zero is ignored so, for example, 07 is valid for the seventh day of the month.

(15) The month of the year. Jan - 1, Feb - 2, March - 3, etc.; Values: 1-12. A leading zero is ignored so, for example, 07 is valid for July.

(16) The value in the wallet cell is always four digits, e.g., 1999, 2000, 2001, ...

(17) A space separated list of protocols available in connection with the specified card. Initial list of case insensitive tokens:

none
set
setcert
iotp
echeck
simcard
phoneid

"Set" indicates usable with SET protocol (i.e., is in a SET wallet) but does not have a SET certificate [[SET](#)]. "Setcert" indicates usable with SET and has a set certificate [[SET](#)].

"iotp" indicates the IOTP protocol [[RFC 2801](#)] is supported at

the customer. "echeck" indicates that the eCheck protocol

[eCheck] is supported at the customer. "simcard" indicates ability to use the transaction instrument built into a Cellphone subscriber for identification. "phoneid" indicates use for the transaction of a billable phone number. "None" indicates that automatic field fill is operating but there is no further information.

- (18) A unique order ID string generated by the consumer software.
- (19) The user ID and password fields can be used in cases where the user has a pre-established account with the merchant to which access is authenticated by such values. For that use one would expect an application to require exactly one user ID and one password field be present.
- (20) URI [[RFC 2396](#)] indicating version of this set of fields. Equal to "urn:ietf:params:ecml:v2.0" for this version. See [section 5](#) below. (see also note 101)
- (21) A string to identify the source and version of form fill software that is acting on behalf of a user. Should contain company and/or product name and version. Example "Wallets Inc., SuperFill, v42.7". (see also note 101)
- (22) A flag to indicate that this web-page/aggregate is the final one for this transaction. (see also note 101)
- (23) Merchant domain name [[RFC 1034](#)] such as www.merchant.example. (see also note 101)
- (24) Domain name [[RFC 1034](#)] of the gateway transaction processor that is actually accepting the payment on behalf of the merchant such as www.processor.example. (see also note 101)
- (25) A Transaction identification string whose format is specific to the processor.
- (26) A URL [[RFC 2396](#)] that can be invoked to inquire about the transaction. (see also notes 100)
- (27) The amount of the transaction in ISO currency format [ISO 4217]. This is two integer numbers with a period in between but no other currency marks (such as a \$ dollar sign).
- (28) This is the three letter ISO currency code [ISO 4217]. For example, for US dollars it is USD.
- (29) ISO Transaction date.

(30) The type of the transaction if known. Currently a value from

D. Eastlake 3rd

[Page 10]

the following list:

debit
credit

- (31) A digital signature base64 encoded [[RFC 2045](#)]. (see also note 101)
- (32) The ReceiptTo fields are used when the BillTo entity, location, or address and the ReceiptTo entity, location, or address are different. For example, when using some forms of Corporate Purchasing Cards or Agent Purchasing Cards, the individual card holder would be in the ReceiptTo fields and the corporate or other owner would be in the BillTo fields.
- (33) An IETF Language Tag as defined in [[RFC 3066](#)].
- (34) User preferences as specified by the merchant. (see also note 102)
- (35) Uniform Resource Locator [[RFC 2396](#)] for accessing the customer's "wallet" software. (see also note 100)
- (36) A single capital letter, M=male, F=Female, U=Unknown [ISO 5218].
- (37) An immutable device identification or serial number. (see also note 102)
- (38) User understandable device brand name. (see also note 102)
- (39) [ISO 8583] field "card acceptor terminal identification".
- (40) [ISO 8583] field "processing code".
- (41) [ISO 8583] field "acquiring institution identification code".
- (42) [ISO 8583] field "forwarding institution identification code".
- (43) [ISO 8583] field "system trace audit field".
- (44) [ISO 8583] field "date effective".
- (45) [ISO 8583] field "point of sale date code".
- (46) [ISO 8583] field "approval code".
- (47) [ISO 8583] field "action code".
- (48) [ISO 8583] field "date settlement".
- (49) [ISO 8583] field "date capture".

- (50) [ISO 8583] field "trace 1 data".
- (51) [ISO 8583] field "trace 2 data".
- (52) User recognizable loyalty card brand name. Values for this field are not controlled and there is no IANA or other registry for them. (see also note 102)
- (53) The card issuer number required by the UK based Switch and Solo acquirers.
- (54) The field names "first_name" and "last_name" have been retained for compatibility with earlier versions of ECML. However, "last_name" should be understood to refer to family or inherited names(s) while "first_name" is the first given or non-inherited name and "middle_name" is the subsequent given or non-inherited name or names if any.
- (55) Uniform Resource Locator [[RFC 2396](#)] for accessing the user's X.509v3 certificate encoded as binary DER. (see also note 100)

Meta notes (referenced by other notes):

- (100) ECML, being a basic field naming and structuring convention, does not impose any particular requirements on these URLs. It is to be expected that most applications that make use of ECML will impose such limitations and perform checking to be sure that provided URLs conform to such limitations before attempting to invoke them.
- (101) This is a field which, when presented in a web page, is usually hidden.
- (102) An ASCII [[ASCII](#)] character string with no leading or trailing white space.

2.2 Exemplar XML Syntax

The following sections provide an XML DTD and an XML Schema that express the ECML fields with ECML v2 naming and ECML v2 hierarchical structure. In case of conflict between this DTD and Schema, the Schema should prevail. Note that the ECML v2 naming and structure may be used in non-XML syntaxes.

The ECML v2 XML syntax is deliberately liberal on the assumption that specific applications making use of ECML will impose their own additional constraints.

For internationalization of ECML, use the general XML character encoding provisions [[XML](#)] (which mandate support of UTF-8 and UTF-16 and permit support of other character sets) and the `xml:lang` attribute, which may be used to specify language information.

2.2.1 ECML v2 XML DTD

The following is an XML DTD for ECML v2.

```
<!-- Electronic Commerce Modeling Language v2 -->

<!ELEMENT Ecom ( #PCDATA | ShipTo | BillTo | ReceiptTo | Payment |
                  Loyalty | User | Merchant | Transaction |
                  TransactionComplete )* >
<!ATTLIST Ecom
    id          ID          #IMPLIED
    ConsumerOrderID CDATA #IMPLIED
    Merchant    CDATA      #IMPLIED
    Mode        (Query|Assert) #IMPLIED
    Processor   CDATA      #IMPLIED
    SchemaVersion "urn:ietf:params:ecml:v2.0"
                #IMPLIED
    WalletID    CDATA      #IMPLIED
    WalletLocation CDATA #IMPLIED >

<!ELEMENT ShipTo ( #PCDATA | Postal | Telecom | Online )* >
<!ATTLIST ShipTo
    id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED >

<!ELEMENT BillTo ( #PCDATA | Postal | Telecom | Online )* >
<!ATTLIST BillTo
    id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED >
```

<!ELEMENT ReceiptTo (#PCDATA | Postal | Telecom | Online)* >

```
<!--ATTLIST ReceiptTo
      id          ID          #IMPLIED
      Mode        (Query|Assert) #IMPLIED >

<!--ELEMENT Postal ( #PCDATA | Name | Company |
                      Street | City | StateProv )* >

<!--ATTLIST Postal
      id          ID          #IMPLIED
      PostalCode  NMTOKEN     #IMPLIED
      Mode        (Query|Assert) #IMPLIED
      CountryCode NMTOKEN     #IMPLIED >

<!--ELEMENT Name EMPTY >
<!--ATTLIST Name
      id          ID          #IMPLIED
      Mode        (Query|Assert) #IMPLIED
      Prefix      NMTOKEN     #IMPLIED
      First       NMTOKEN     #IMPLIED
      Middle      NMTOKEN     #IMPLIED
      Last        NMTOKEN     #IMPLIED
      Suffix      NMTOKEN     #IMPLIED >

<!--ELEMENT Street EMPTY >
<!--ATTLIST Street
      id          ID          #IMPLIED
      Mode        (Query|Assert) #IMPLIED
      Line1       CDATA       #REQUIRED
      Line2       CDATA       #IMPLIED
      Line3       CDATA       #IMPLIED >

<!--ELEMENT Company (#PCDATA) >
<!--ATTLIST Company
      Mode        (Query|Assert) #IMPLIED >

<!--ELEMENT City (#PCDATA) >
<!--ATTLIST City
      Mode        (Query|Assert) #IMPLIED >

<!--ELEMENT StateProv (#PCDATA) >
<!--ATTLIST StateProv
      Mode        (Query|Assert) #IMPLIED >

<!--ELEMENT Telecom ( #PCDATA | Phone )* >
<!--ATTLIST Telecom
      Mode        (Query|Assert) #IMPLIED >

<!--ELEMENT Phone EMPTY >
<!--ATTLIST Phone
      id          ID          #IMPLIED
```

Mode (Query|Assert) #IMPLIED

D. Eastlake 3rd

[Page 14]

```

        Number      CDATA      #REQUIRED >

<!ELEMENT Online ( #PCDATA | Email )* >
<!ATTLIST Online
        Mode      (Query|Assert) #IMPLIED >

<!ELEMENT Email EMPTY >
<!ATTLIST Email
        id          ID          #IMPLIED
        Mode      (Query|Assert) #IMPLIED
        Address    CDATA      #REQUIRED >

<!ELEMENT Payment Card >
<!ATTLIST Payment
        Mode      (Query|Assert) #IMPLIED >

<!ELEMENT Card ExpDate ValidDate? >
<!ATTLIST Card
        id          ID          #IMPLIED
        Mode      (Query|Assert) #IMPLIED
        Name        CDATA      #IMPLIED
        Type        NMTOKEN    #IMPLIED
        Number      NMTOKEN    #REQUIRED
        Protocols   NMTOKENS    #IMPLIED
        Verification NMTOKEN    #IMPLIED
        Issuer      NMTOKEN    #IMPLIED >

<!ELEMENT Loyalty ExpDate? ValidDate? >
<!ATTLIST Loyalty
        id          ID          #IMPLIED
        Mode      (Query|Assert) #IMPLIED
        Name        CDATA      #IMPLIED
        Type        NMTOKEN    #IMPLIED
        Number      NMTOKEN    #REQUIRED
        Verification NMTOKEN    #IMPLIED >

<!ELEMENT ExpDate EMPTY >
<!ATTLIST ExpDate
        id          ID          #IMPLIED
        Mode      (Query|Assert) #IMPLIED
        Day        NMTOKEN    #IMPLIED
        Month      NMTOKEN    #REQUIRED
        Year        NMTOKEN    #REQUIRED >

<!ELEMENT ValidDate EMPTY >
<!ATTLIST ValidDate
        id          ID          #IMPLIED
        Mode      (Query|Assert) #IMPLIED
```

Day	NMTOKEN	#IMPLIED
Month	NMTOKEN	#IMPLIED

D. Eastlake 3rd

[Page 15]

```

        Year          NMTOKEN    #REQUIRED >

<!ELEMENT User ( #PCDATA | UserID | Password )* >
<!ATTLIST User
    id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED
    CertificateURL CDATA    #IMPLIED
    DataCountry NMTOKEN    #IMPLIED
    DataLanguage CDATA     #IMPLIED >

<!ELEMENT UserID (#PCDATA) >
<!ATTLIST UserID
    Mode        (Query|Assert) #IMPLIED >

<!ELEMENT Password (#PCDATA) >
<!ATTLIST Password
    Mode        (Query|Assert) #IMPLIED >

<!ELEMENT Merchant Terminal >
<!ATTLIST Merchant
    Mode        (Query|Assert) #IMPLIED
    id          ID          #IMPLIED >

<!ELEMENT Terminal EMPTY >
<!ATTLIST Terminal
    Id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED
    Data        CDATA     #IMPLIED >

<!ELEMENT Transaction ( #PCDATA | Id | Code | Date | Data |
                        Inquiry | Signature )* >
<!ATTLIST Transaction
    Amount      CDATA      #IMPLIED
    Currency    NMTOKEN    #IMPLIED
    Mode        (Query|Assert) #IMPLIED
    Type        NMTOKEN    #IMPLIED >

<!ELEMENT Id EMPTY >
<!ATTLIST Id
    Id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED
    CID        NMTOKEN    #IMPLIED
    Reference   NMTOKEN    #IMPLIED
    Acquire     NMTOKEN    #IMPLIED
    Forward     NMTOKEN    #IMPLIED >

<!ELEMENT Code EMPTY >
<!ATTLIST Code
    Mode        (Query|Assert) #IMPLIED

```

Processing CDTATA #IMPLIED

D. Eastlake 3rd

[Page 16]

Approval	NMTOKEN	#IMPLIED
Retrieval	NMTOKEN	#IMPLIED
Action	NMTOKEN	#IMPLIED
Reason	NMTOKEN	#IMPLIED
POS	NMTOKEN	#IMPLIED >

```
<!ELEMENT Date (Effective?, Settle?, Capture?) >
```

```
<!ATTLIST Date
  Mode      (Query|Assert) #IMPLIED
  id        ID              #IMPLIED >
```

```
<!ELEMENT Effective EMPTY >
```

```
<!ATTLIST Effective
  id        ID              #IMPLIED
  Mode      (Query|Assert) #IMPLIED
  Day       NMTOKEN         #REQUIRED
  Month     NMTOKEN         #REQUIRED
  Year      NMTOKEN         #REQUIRED >
```

```
<!ELEMENT Settle EMPTY >
```

```
<!ATTLIST Settle
  id        ID              #IMPLIED
  Mode      (Query|Assert) #IMPLIED
  Day       NMTOKEN         #REQUIRED
  Month     NMTOKEN         #REQUIRED
  Year      NMTOKEN         #REQUIRED >
```

```
<!ELEMENT Capture EMPTY >
```

```
<!ATTLIST Capture
  id        ID              #IMPLIED
  Mode      (Query|Assert) #IMPLIED
  Day       NMTOKEN         #REQUIRED
  Month     NMTOKEN         #REQUIRED
  Year      NMTOKEN         #REQUIRED >
```

```
<!ELEMENT Data ( #PCDATA | Trace | PrivateUse | Response |
  AAV | Track1 | Track2 )* >
```

```
<!ATTLIST Data
  Mode      (Query|Assert) #IMPLIED >
```

```
<!ELEMENT Trace (#PCDATA) >
```

```
<!ATTLIST Trade
  id        ID              #IMPLIED
  Mode      (Query|Assert) #IMPLIED >
```

```
<!ELEMENT PrivateUse (#PCDATA) >
```

```
<!ATTLIST PrivateUse
  id        ID              #IMPLIED
```

Mode (Query|Assert) #IMPLIED >

D. Eastlake 3rd

[Page 17]

```
<!ELEMENT Response (#PCDATA) >
<!ATTLIST Response
    id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED >

<!ELEMENT AAV (#PCDATA) >
<!ATTLIST AAV
    id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED >

<!ELEMENT Track1 (#PCDATA) >
<!ATTLIST Track1
    id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED >

<!ELEMENT Track2 (#PCDATA) >
<!ATTLIST Track2
    id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED >

<!ELEMENT Inquiry (#PCDATA) >
<!ATTLIST Inquiry
    id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED >

<!ELEMENT Signature (#PCDATA) >
<!ATTLIST Signature
    id          ID          #IMPLIED
    Mode        (Query|Assert) #IMPLIED >

<!ELEMENT TransactionComplete EMPTY >
```


[2.2.2](#) ECML v2 XML Schema

The following is an XML Schema for ECML v2.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Electronic Commerce Modeling Language v2 -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <xs:attribute name="Mode">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Query"/>
        <xs:enumeration value="Assert"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="id" type="xs:ID"/>
  <xs:complexType name="EcomSimpleText">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:element name="Ecom">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="ShipTo"/>
        <xs:element ref="BillTo"/>
        <xs:element ref="ReceiptTo"/>
        <xs:element ref="Payment"/>
        <xs:element ref="Loyalty"/>
        <xs:element ref="User"/>
        <xs:element ref="Merchant"/>
        <xs:element ref="Transaction"/>
        <xs:element ref="TransactionComplete"/>
      </xs:choice>
      <xs:attribute ref="Mode" use="optional"/>
      <xs:attribute ref="id" use="optional"/>
      <xs:attribute name="ConsumerOrderID" use="optional"/>
      <xs:attribute name="Merchant" use="optional"/>
      <xs:attribute name="Processor" use="optional"/>
      <xs:attribute name="SchemaVersion" type="xs:string"
        fixed="urn:ietf:params:ecml:v2.0"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<xs:attribute name="WalletID" use="optional"/>  
<xs:attribute name="WalletLocation" type="xs:anyURI"
```

```
        use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="ShipTo">
    <xs:complexType mixed="true">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="Postal"/>
            <xs:element ref="Telecom"/>
            <xs:element ref="Online"/>
        </xs:choice>
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="BillTo">
    <xs:complexType mixed="true">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="Postal"/>
            <xs:element ref="Telecom"/>
            <xs:element ref="Online"/>
        </xs:choice>
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="ReceiptTo">
    <xs:complexType mixed="true">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="Postal"/>
            <xs:element ref="Telecom"/>
            <xs:element ref="Online"/>
        </xs:choice>
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="Postal">
    <xs:complexType mixed="true">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="Name"/>
            <xs:element ref="Company"/>
            <xs:element ref="Street"/>
            <xs:element ref="City"/>
            <xs:element ref="StateProv"/>
        </xs:choice>
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
        <xs:attribute name="PostalCode" type="xs:NMTOKEN"
```

```
        use="optional"/>  
<xs:attribute name="CountryCode" type="xs:NMTOKEN"
```

```
        use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="Telecom">
    <xs:complexType mixed="true">
        <xs:sequence maxOccurs="unbounded">
            <xs:element name="Phone">
                <xs:complexType>
                    <xs:attribute ref="Mode" use="optional"/>
                    <xs:attribute ref="id" use="optional"/>
                    <xs:attribute name="Number"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute ref="Mode" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="Online">
    <xs:complexType mixed="true">
        <xs:sequence maxOccurs="unbounded">
            <xs:element name="Email">
                <xs:complexType>
                    <xs:attribute ref="Mode" use="optional"/>
                    <xs:attribute ref="id" use="optional"/>
                    <xs:attribute name="Address"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute ref="Mode" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="Payment">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Card">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="ExpDate"/>
                        <xs:element ref="ValidDate" minOccurs="0"/>
                    </xs:sequence>
                    <xs:attribute ref="Mode" use="optional"/>
                    <xs:attribute ref="id" use="optional"/>
                    <xs:attribute name="Name" use="optional"/>
                    <xs:attribute name="Type" type="xs:NMTOKEN"
                        use="optional"/>
                    <xs:attribute name="Number" type="xs:decimal"/>
                    <xs:attribute name="Protocols" type="xs:NMTOKENS"
                        use="optional"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute ref="Mode" use="optional"/>
    </xs:complexType>
</xs:element>
```

```
<xs:attribute name="Verification"
  type="xs:NMTOKEN" use="optional"/>
```

```
        <xs:attribute name="Issuer" type="xs:NMTOKEN"
            use="optional"/>
    </xs:complexType>
</xs:element>
</xs:sequence>
    <xs:attribute ref="Mode" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="Loyalty">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="ExpDate"/>
            <xs:element ref="ValidDate" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
        <xs:attribute name="Name" use="optional"/>
        <xs:attribute name="Type" type="xs:NMTOKEN"
            use="optional"/>
        <xs:attribute name="Number" type="xs:NMTOKEN"/>
        <xs:attribute name="Verification" type="xs:NMTOKEN"
            use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="ExpDate">
    <xs:complexType>
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
        <xs:attribute name="Day" type="xs:positiveInteger"/>
        <xs:attribute name="Month" type="xs:positiveInteger"/>
        <xs:attribute name="Year" type="xs:positiveInteger"/>
    </xs:complexType>
</xs:element>
<xs:element name="ValidDate">
    <xs:complexType>
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
        <xs:attribute name="Day" type="xs:positiveInteger"/>
        <xs:attribute name="Month" type="xs:positiveInteger"/>
        <xs:attribute name="Year" type="xs:positiveInteger"/>
    </xs:complexType>
</xs:element>
<xs:element name="User">
    <xs:complexType mixed="true">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="UserID"/>
            <xs:element ref="Password"/>
        </xs:choice>
```

```
<xs:attribute ref="Mode" use="optional"/>  
<xs:attribute ref="id" use="optional"/>
```

```
<xs:attribute name="CertificateURL" type="xs:anyURI"
  use="optional"/>
<xs:attribute name="DataCountry" type="xs:NMTOKEN"
  use="optional"/>
<xs:attribute name="DataLanguage" type="xs:language"
  use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="Transaction">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="Id"/>
      <xs:element ref="Code"/>
      <xs:element ref="Date"/>
      <xs:element ref="Data"/>
      <xs:element ref="Inquiry"/>
      <xs:element ref="Signature"/>
    </xs:choice>
    <xs:attribute ref="Mode" use="optional"/>
    <xs:attribute name="Currency" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Type" type="xs:NMTOKEN"
      use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="Date">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Effective" minOccurs="0"/>
      <xs:element ref="Settle" minOccurs="0"/>
      <xs:element ref="Capture" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute ref="Mode" use="optional"/>
    <xs:attribute ref="id" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="Data">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="Trace"/>
      <xs:element ref="PrivateUse"/>
      <xs:element ref="Response"/>
      <xs:element ref="AAV"/>
      <xs:element ref="Track1"/>
      <xs:element ref="Track2"/>
    </xs:choice>
    <xs:attribute ref="Mode" use="optional"/>
  </xs:complexType>
```

```
</xs:element>  
<xs:element name="Merchant">
```

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="Terminal">
      <xs:complexType>
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
        <xs:attribute name="Data" use="optional"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute ref="Mode" use="optional"/>
  <xs:attribute ref="id" use="optional"/>
</xs:complexType>
</xs:element>

<xs:element name="AAV" type="EcomSimpleText"/>
<xs:element name="Capture">
  <xs:complexType>
    <xs:attribute ref="Mode" use="optional"/>
    <xs:attribute ref="id" use="optional"/>
    <xs:attribute name="Day" type="xs:NMTOKEN"/>
    <xs:attribute name="Month" type="xs:NMTOKEN"/>
    <xs:attribute name="Year" type="xs:NMTOKEN"/>
  </xs:complexType>
</xs:element>
<xs:element name="City" type="EcomSimpleText"/>
<xs:element name="Code">
  <xs:complexType>
    <xs:attribute ref="Mode" use="optional"/>
    <xs:attribute name="Processing" use="optional"/>
    <xs:attribute name="Approval" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Retrieval" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Action" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Reason" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="POS" type="xs:NMTOKEN"
      use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="Company" type="EcomSimpleText"/>
<xs:element name="Effective">
  <xs:complexType>
    <xs:attribute ref="Mode" use="optional"/>
    <xs:attribute ref="id" use="optional"/>
    <xs:attribute name="Day" type="xs:NMTOKEN"/>
```

```
<xs:attribute name="Month" type="xs:NMTOKEN"/>  
<xs:attribute name="Year" type="xs:NMTOKEN"/>
```

```
</xs:complexType>
</xs:element>
<xs:element name="Id">
  <xs:complexType>
    <xs:attribute ref="Mode" use="optional"/>
    <xs:attribute ref="id" use="optional"/>
    <xs:attribute name="CID" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Reference" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Acquire" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Forward" type="xs:NMTOKEN"
      use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="Inquiry">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:anyURI">
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="Name">
  <xs:complexType>
    <xs:attribute ref="Mode" use="optional"/>
    <xs:attribute ref="id" use="optional"/>
    <xs:attribute name="Prefix" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="First" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Middle" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Last" type="xs:NMTOKEN"
      use="optional"/>
    <xs:attribute name="Suffix" type="xs:NMTOKEN"
      use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="Password" type="EcomSimpleText"/>
<xs:element name="PrivateUse" type="EcomSimpleText"/>
<xs:element name="Response" type="EcomSimpleText"/>
<xs:element name="Settle">
  <xs:complexType>
    <xs:attribute ref="Mode" use="optional"/>
```

```
<xs:attribute ref="id" use="optional"/>  
<xs:attribute name="Day" type="xs:NMTOKEN"/>
```

```
        <xs:attribute name="Month" type="xs:NMTOKEN"/>
        <xs:attribute name="Year" type="xs:NMTOKEN"/>
    </xs:complexType>
</xs:element>
<xs:element name="Signature">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute ref="Mode" use="optional"/>
                <xs:attribute ref="id" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="StateProv" type="EcomSimpleText"/>
<xs:element name="Street">
    <xs:complexType>
        <xs:attribute ref="Mode" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
        <xs:attribute name="Line1"/>
        <xs:attribute name="Line2" use="optional"/>
        <xs:attribute name="Line3" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="Trace" type="EcomSimpleText"/>
<xs:element name="Track1" type="EcomSimpleText"/>
<xs:element name="Track2" type="EcomSimpleText"/>
<xs:element name="TransactionComplete">
    <xs:complexType/>
</xs:element>
<xs:element name="UserID" type="EcomSimpleText"/>

</xs:schema>
```


3. Usage Notes for ECML v2

This section provides a general usage guide for ECML v2.

3.1 Presentation of the Fields

ECML v2 merely names fields and specifies their content and hierarchical organization. It does not constrain the order or completeness of communication of or query for these fields.

Some parties may wish to provide or ask for more information, some less by omitting fields. Some may ask for the information they want in one interaction or web page, others may ask for parts of the information at different times in multiple interactions or different web pages. For example, it is common to ask for "ship to" information earlier, so shipping cost can be computed, before the payment method information. Some parties may require that all the information they request be provided while others make much information optional. Other variations are likely.

Every element may be flagged as a query or assertion by including, when XML syntax is in use, the optional Mode attribute with the value "Query" or "Assert" respectively. The Mode attribute effects all descendant elements until overridden by a lower level element with a Mode attribute. Thus it is easy to indicate that all of the elements in an ECML v2 structure are present as queries or assertions.

Query elements may have data content. Such content SHOULD be interpreted as a default value to be returned if no better value is known.

There is no way with Version 2.0 of ECML to indicate what query fields a party considers mandatory to be answered. From this point of view, all fields queried are optional to complete. However, a party may give an error or re-present a request for information if some field it requires is not completed, just as it may if a field is completed in a manner it considers erroneous.

3.2 Methods and Flow of Setting the Fields

There are a variety of methods of communication possible between the parties by which each can indicate what fields it wants the other to provide. Probably the easiest method for currently deployed mass software is as fields in an [[HTML](#)] form. Other possibilities are to use an [[XML](#)] exchange, the IOTP Authenticate transaction [[RFC 2801](#)],

or proprietary protocols.

D. Eastlake 3rd

[Page 27]

So that browser software can tell what version it is dealing with, it is REQUIRED that the Ecom_SchemaVersion field be included in every transactions when ECML is being used on the web. Ecom_SchemaVersion SHOULD appear on every web page that has any Ecom fields. It is usually a hidden field in HTML Forms.

User action or the appearance of the Ecom_SchemaVersion field are examples of triggers that can be used to initiate a facility capable of providing information in response to an ECML based query or utilizing information from ECML assertions. Because some web software may require user activation, it is RECOMMENDED that these be at least one user visible Ecom field on every web page with any Ecom fields present when ECML is used via the web.

Because, under some circumstances, communications can proceed very slowly, it may not be clear to an automated processing function when it is finished receiving ECML fields on a web page or the like. For this reason, it is RECOMMENDED that the Ecom_SchemaVersion field be the last Ecom field on a web page.

Transfer or requests for information can extend over several interactions or web pages. Without further provision, a facility could either require re-starting on each page or possibly violate or appear to violate privacy by continuing to provide personal data beyond with end of the transaction with a particular business. For this reason the Ecom_TransactionComplete field, which is normally hidden when part of an HTML Form, is provided. It is RECOMMENDED that it appear on the last interaction or web page involved in a transaction, just before an Ecom_SchemaVersion field, so that multi-interaction automated logic receives a hint as to when to stop if it chooses to check for this field.

4. Security and Privacy Considerations

The information called for by many of these fields is sensitive. It should be protected from unauthorized modification and kept confidential if stored in a location or transmitted over a channel where it might otherwise be observed. In addition, the authenticity of the information will be a concern in many systems.

Mechanisms for such protection and authentication are not specified herein but might, depending on circumstances, include object security protocols, such as XMLDSIG [[RFC 3275](#)], XML encryption [[XMLENC](#)], or CMS [[RFC 3852](#)], or channel security such as TLS [[RFC 2246](#)] or IPsec [[RFC 2411](#)]. Systems in which an ECML field or fields are stored and later forwarded will likely find object security to be the most appropriate.

When information is being requested from a user, their control over release of such information is needed to protect their privacy.

Software that is installed on a shared or public terminals should be configurable such that memory of any sensitive or individual identity information is fully disabled. This is vital to protect the privacy of library patrons, students, and customers using public terminals, and children who might, for example, use a form on a public terminal without realizing that their information is being stored.

When sensitive or individual identification information is stored, the operator or user should have an option to protect the information, for example with a password without which the information will be unavailable, even to someone who has access to the file(s) in which it is being stored.

Any multi-page/screen or other multi-aggregate field fill in or data provision mechanism SHOULD check for the Ecom_TransactionComplete field and cease automated fill when it is encountered until fill is further authorized.

It should be remembered that default, hidden, and other values transferred to another party may be maliciously modified before being returned.

5. IANA Considerations

The sections below provide for

1. registration of the ECML v2 XML schema contained in this document,
2. a version URN for ECML versions,
3. the subsidiary registration of particular ECML versions and the specific registration of Version 2.0,
4. three additional IANA registries for elements appearing in three ECML v2 fields.

5.1 ECML v2 Schema Template

The ECML v2 schema give in [section 2.2.2](#) above is registered as follows:

URI: urn:ietf:params:xml:schema:<id-please assign>

Registrant Contact: The IESG <iesg@ietf.org>

XML: The XML Schema in [section 2.2.2](#) above.

5.2 ECML v2 URN Template

As specified by the template below from [[RFC 3553](#)],
urn:ietf:params:ecml is permanently registered with sub registration
via RFC publication.

Registry name: urn:ietf:params:ecml

Specification: RFC XXXX - ([draft-ietf-trade-ecml2-spec](#)-.txt)

Repository: RFC XXXX - ([draft-ietf-trade-ecml2-spec](#)-.txt)

Index value: Values subordinate to urn:ietf:params:ecml are
registered by RFC publication. As provided in [[RFC 3553](#)], once
such a value is registered, it may never change.

5.2.1 Subregistration of v2.0

The subordinate value "v2.0" is hereby permanently registered so that
the URN

urn:ietf:params:ecml:v2.0

is used to indicate an ECML field or fields that conform to this
specification. Although it is not anticipated that deeper values
subordinate to this URN will need to be registered, if necessary they
are registered by IESG approval..

5.3 IANA Registries

There are three fields described in [Section 2.1.2](#) that require the
establishment of IANA registries as described below:

Ecom_Payment_Card_Type

A registry of case insensitive alphanumeric ASCII [[ASCII](#)] card
type designations from one to four characters in length with no
white space. See [section 2.1.2](#), note 11, for the initial 12
designations. Designations are added based on expert approval.
Applicants for registration will normally be required to already
have an ISO Issuer Identification Number (IIN) or set of IINs.

Ecom_Payment_Card_Protocol

This field holds a space separated list of protocols designated
by case insensitive alphanumeric ASCII [[ASCII](#)] tokens from this
registry or the token "none". See [section 2.1.2](#), note 17, for

the initial seven registered tokens (including "none") and

further information. Tokens are added to the registry based on expert approval.

Ecom_Transaction_Type

A case insensitive alphabetic ASCII [[ASCII](#)] value indicating the type of transaction. See [section 2.1.2](#), note 30, for the initial two registered values. Values are added based on expert approval.

Normative References

[ASCII] - USA Standard Code for Information Interchange, X3.4
American National Standards Institute; New York, 1968.

[E.164] - ITU-T Recommendation E.164/I.331 (05/97): The International
Public Telecommunication Numbering Plan. 1997.

[IANA] - Internet Assigned Numbers Authority, Official Names for
Character Sets, ed. Keld Simonsen et al. <[ftp://ftp.isi.edu/in-
notes/iana/assignments/character-sets](http://ftp.isi.edu/in-notes/iana/assignments/character-sets)>.

[ISO 3166] - "Codes for the representation of names of countries and
their subdivisions -- Part 1: Country codes", ISO 3166-1, 1997.

[ISO 4217] - "Codes for the representation of currencies and funds",
ISO 4217, 2001.

[ISO 5218] - "Information interchange -- Representation of human
sexes", ISO 5218, 1977.

[ISO 7812] - "Identification card - Identification of issuers - Part
1: Numbering system", ISO 7812-1, 2000.

[ISO 8583] - "Financial transaction card originated messages -
Interchange message specifications - Part 1: Messages, elements and
code values", ISO 8583-1, 2001.

[RFC 1766] - "Tags for the Identification of Languages", H.
Alvestrand, March 1995.

[RFC 2045] - "Multipurpose Internet Mail Extensions (MIME) Part One:
Format of Internet Message Bodies", N. Freed, N. Borenstein, November
1996.

[RFC 2396] - "Uniform Resource Identifiers (URI): Generic Syntax", T.
Berners-Lee, R. Fielding, L. Masinter, August 1998.

[RFC 3066] - "Tags for the Identification of Languages", H.
Alvestrand, January 2001.

[XML] - Extensible Markup Language (XML) 1.0 (Second Edition),
<<http://www.w3.org/TR/1998/REC-xml>>, T. Bray, J. Paoli, C. M.
Sperberg-McQueen, E. Maler

Informative References

[eCheck] - <<http://www.echeck.org>>

[EMV] - <<http://www.emvco.org/specifications.cfm>>

[HTML] - "HTML 3.2 Reference Specification", <
<http://www.w3.org/TR/REC-html32.html>>, D. Raggett, January 1997.

[P3P BASE] - "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification", L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, December 2000, <<http://www.w3.org/TR/WD-P3P/basedata.html>>.

[P3P ECOM] - "Using P3P for E-Commerce", J. Coco, S. Klien, D. Schutzer, S. Yen, A. Slater, November 1999,
<<http://www.w3.org/TR/P3P-for-ecommerce>>.

[RFC 1034] - "Domain names - concepts and facilities", P.V. Mockapetris, Nov-01-1987.

[RFC 2026] - "The Internet Standards Process -- Revision 3", S. Bradner, October 1996.

[RFC 2246] - "The TLS Protocol: Version 1.0", T. Dierks, C. Allen. January 1999.

[RFC 2411] - "IP Security: Document Roadmap", R. Thayer, N. Doraswamy, R. Glenn, November 1998.

[RFC 2706] - "ECML v1: Field Names for E-Commerce", D. Eastlake, T. Goldstein, September 1999.

[RFC 2801] - "Internet Open Trading Protocol - IOTP Version 1.0", D. Burdett, April 2000.

[RFC 3106] - "ECML v1.1: Field Specifications for E-Commerce", D. Eastlake, T. Goldstein, April 2001.

[RFC 3275] - "(Extensible Markup Language) XML-Signature Syntax and Processing", D. Eastlake 3rd, J. Reagle, D. Solo, March 2002.

[RFC 3553] - "An IETF URN Sub-namespace for Registered Protocol Parameters", M. Mealling, L. Masinter, T. Hardie, G. Klyne, June 2003.

[RFC 3852] - "Cryptographic Message Syntax (CMS)", R. Housley, July 2004.

[SET] - Secure Electronic Transaction,

D. Eastlake 3rd

[Page 33]

<http://www.setco.org/set_specifications.html>

[XMLENC] - "XML Encryption Syntax and Processing", D. Eastlake 3rd,
J. Reagle, <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>,
December 2002.

Appendix: Changes from v1.1 to v2

Substantial rewording of text to change the emphasis from HTML Form Fields to XML Syntax.

Addition of the merchant -> processor fields.

Addition of the Ecom_Wallet_Location and Ecom_User_Certificate_URL fields.

Addition of the "Mode" attribute.

Addition of the ECom_Payment_Card_IssueNumber, Loyalty Card fields, Device ID, Valid From, and User Data fields.

Addition of an XML schema.

Some minor fixes related to telephone numbers.

Addition of IANA Considerations section.

Updating of RFC references for obsoleted RFCs.

Copyright and Disclaimer

Copyright (C) The Internet Society 2004. This document is subject to the rights, licenses and restrictions contained in [BCP 78](#) and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Author's Addresses

Donald E. Eastlake 3rd
Motorola Laboratories
155 Beaver Street
Milford, MA 01757 USA

Phone: 1-508-786-7554 (work)
1-508-634-2066 (home)
EMail: Donald.Eastlake@motorola.com

File name and Expiration

This file is [draft-ietf-trade-ecml2-spec-13.txt](#).

It expires April 2005.

