Developers Guide

Version: 1.24
Publication Date: 09-Oct-2009

Contents

A. Introduction

This document is the DataCash Developers Guide. It is aimed at developers and technical personnal who will be integrating applications. This is reflected in the layout of the document, which covers the submission of data in XML form to DataCash.

As each Service requires particular information to be submitted and this information tends to be grouped within similar areas (parent elements) of the XML schema, the names of the parent elements will first be introduced.

Each parent will then be placed into its context in the schema and its child elements discussed. This includes any restrictions on the format, length and transaction type of each element. Example XML for each parent will be shown.

Complete XML example Requests and successful Responses will then be given for each transaction type.

If the Service contains additional aspects – for example notifications for Standard Direct Debit and Standard Recurring Transactions – this will also be covered.

Changelog	
Version	**Notes**
1.0	Credit and Debit Card and Line Item Detail
1.1	Added Airlines, AVSCV2, Pre-Registered Cards and Smart Voucher
1.2	Added URU®, Capture Method
1.3	Added BatchInput and BACS Services
1.4	Added Click2Pay
1.5	Added Real-time fraud screening, Historic Recurring, “Fire and Forget” Card Payments and Cardholder Present
1.6	URU® phase 2 added
1.7	Added 3-D Secure and Batch Input for Cardholder Present
1.7.1	Updated Chip and PIN, 3DS
1.7.2	Added password expiry, updated URU, plus formatting changes
1.7.3	Added BMS & Omnipay Airlines
1.8	Added PayPal, Pre-Registered cards with 3-D Secure (DataCash MPI and 3rd Party MPI), Pre-Registered cards with CV2AVS, Batched Fraud Screening and VEOD
1.9	PayPal req_billing_address usage clarified
1.10	PayPal expected ACK responses are now correctly specified
1.11	Added MPI Only
1.12	Added MPI Only preregistered card
1.13	Added RBS Gift Card
1.14	Added PPT
1.15	Added CP unattended device

1.16	Added Fexco DCC
1.17	Added Merchant Narrative
1.18	Added acquirer specific notes for Chase
1.19	Added Cash Advance
1.20	Added PayPal Auth and Capture
1.21	Removed Click2Pay and Smart Voucher
1.22	Updated PayPal Auth and Capture
1.23	Added PayPal Airlines

This document includes all Services DataCash currently provides. If you are integrating a service which is still in development, please contact Support for the relevant documents.

This document is available to download from the DataCash Developers Area: https://testserver.datacash.com/software/download.cgi.

A.1. XML Requests and Responses

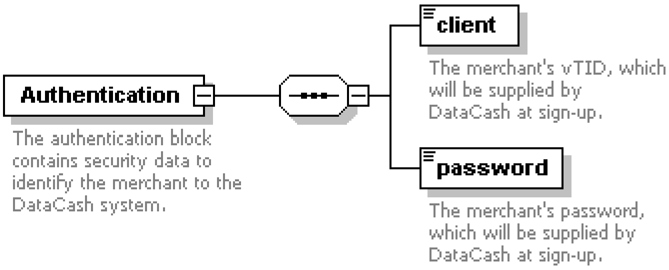
There are certain features of the Request and Response that are global to all DataCash Services – these elements are covered in this section. Others are only used for a particular Service or group of Services – these elements will be covered within the documentation for that Service.

A.1.1. Request

In the Request, only the Authentication element is used for all DataCash Services

A.1.1.1. Authentication

Element Name:	`Authentication`
Position:	`Request`


Authentication
The authentication block contains security data to identify the merchant to the DataCash system.
client
The merchant's vTID, which will be supplied by DataCash at sign-up.
password
The merchant's password, which will be supplied by DataCash at sign-up.

Element Name	Description	Values / limitations
`client`	Specifies the account the transaction will be processed on	As advised by DataCash
`password`	The password for the account.	As advised by DataCash[1]

1. When your account is setup, DataCash will advise you of the password. This password has a maximum lifetime of twelve months and you are also responsible for changing it each time a person who know it leaves your organisation. Further information about generating a new password is available in the Support Centre: http://datacash.custhelp.com/cgi-bin/datacash.cfg/php/enduser/std_adp.php?p_faqid=567 (keyword search "DPG password")

Example XML for Authentication complex elements

```
<Authentication>
  <client>99000001</client>
  <password>mypasswd</password>
</Authentication>
```

A.1.2. Response

There are several elements to the Response that are returned regardless of the Service used. These are:

Element Name	Description
`status`	Numeric return code indicating the result of the transaction
`reason`	A text field expanding on the status of the transaction
`datacash_reference`	The DataCash Reference of the transaction.
`time`	The Unix Timestamp at which the transaction reached our server
`mode`	Indicates the current status of your account – one of: live, test or accreditation

There are also elements that may be returned, depending upon the result of the transaction:

Element Name	Description
`information`	If an error is generated, additional information is often returned to allow the source of the error to be identified
`merchant_reference`	Your reference number

B. Single Card Payments

B.1. Credit and Debit Card Service

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/bankcard/overview.shtml.

B.1.1. Schema Elements for Request

In this section the required fields for each transaction type will be presented, along with example XML for those fields. The XML is presented in italics for those fields that are not required for all transaction types. If the transaction type is specified in the XML, this is highlighted in the cases where other transaction types can be used in its place.

As not all transaction types use the same fields, each field will be labelled with the following key:

- O - Optional
- R - Required, field must be presented
- X - Excluded, presenting this field will cause the transaction to fail
- M - Mandatory if Available, if the information is available, it should be presented

Please refer to the website for definitions of the transaction types and examples of when you may wish to implement them.

Initial Transactions with Card details

The `auth`, `pre`, `refund` and `erp` transaction types all require the same information about the transaction to be provided.
This data is passed in these distinct places in the schema:

- `Request`
  - `Authentication` - section A.1.1.1
  - Transaction
    - `CardTxn` - the type and authorisation code, section B.1.1.2
      - `Card` - this contains all the information about the Card, section B.1.1.1
    - `TxnDetails` - contains details of the transaction, section B.1.1.3

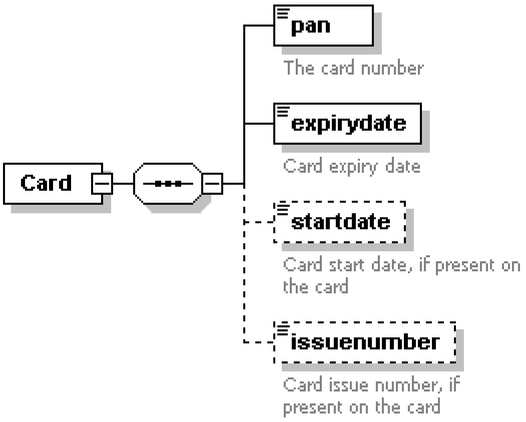
Historic Transactions

The `cancel`, `fulfill` and `txn_refund` transaction types all require similar information about the transaction to be provided.
This data is passed in these distinct places in the schema:

- `Request`
  - `Authentication` - section A.1.1.1
  - `Transaction`
    - `HistoricTxn` - information relating to the original transaction, section B.1.1.4
    - `TxnDetails` - contains details of the new transaction, section B.1.1.3

B.1.1.1. Card

Element Name: `Card`
Position: `Request.Transaction.CardTxn`


Card
pan
The card number
expirydate
Card expiry date
startdate
Card start date, if present on the card
issuenumber
Card issue number, if present on the card

Elements of Card						
Element Name	**description**	**values / limitations**	**auth**	**pre**	**refund**	**erp**
`pan`	card number	must be a valid card number between 13 and 19 digits in length	R	R	R	R
`expirydate`	expiry date for the card	must be passed in `mm/yy` format	R	R	R	R
`startdate`	start date for the card	must be passed in `mm/yy` format	M	M	M	M
`issuenumber`	issue number of the card	must be one or two digits long	M	M	M	M

The CardInfo files can be used to determine whether a startdate or an issuenumber is required for each card.

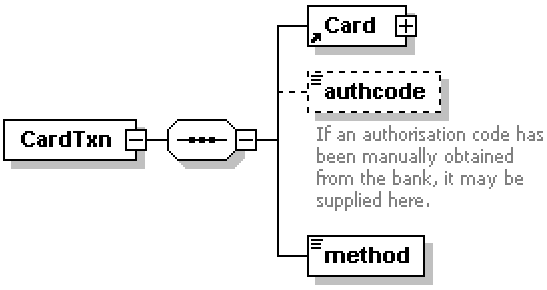
Example XML for Card complex elements

```
<Card>
  <pan>444433*********1</pan>
  <expirydate>04/06</expirydate>
</Card>

<Card>
  <pan>675998*********1</pan>
  <expirydate>04/06</expirydate>
  <startdate>01/99</startdate>
  <issuenumber>01</issuenumber>
</Card>
```

B.1.1.2. CardTxn

For transactions using card details only. In addition to the basic Card details, the transaction type and authorisation code (if applicable) are sent in the CardTxn element:

Element Name: `CardTxn`
Position: `Request.Transaction`


Card
authcode
If an authorisation code has been manually obtained from the bank, it may be supplied here.
CardTxn
method

Elements of CardTxn						
Element Name	**description**	**values / limitations**	**auth**	**pre**	**refund**	**erp**
Card	see section B.1.1.1					
authcode	authorisation code received from the bank	If presented, must be value received from Banks Authorisation centre	O	O	O	O
method	the transaction type	auth pre refund erp	R	R	R	R

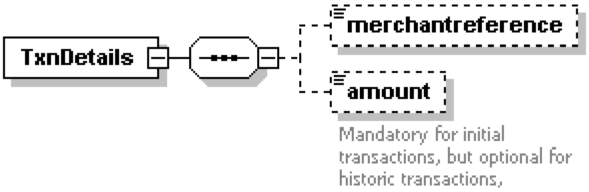
Example XML for CardTxn complex elements

```
<CardTxn>
  <Card>…</Card>
  <authcode>123123</authcode>
  <method>pre<method>
</CardTxn>
```

B.1.1.3. TxnDetails

For all transactions

Element Name: `TxnDetails`
Position: `Request.Transaction`


TxnDetails
merchantreference
amount
Mandatory for initial transactions, but optional for historic transactions,

Elements of TxnDetails										
Element Name	**description**	**values / limitations**	**auth**	**pre**	**refund**	**erp**	**cancel**	**fulfill**	**txn_refund**
`merchantreference`	A unique reference number for each transaction	Minimum 6, maximum 30 alphanumeric characters. Must be unique	R	R	R	R	-	-	-
`amount`	The value of the transaction.		R	R	R	R	-	O	O

Attributes for Elements of TxnDetails									
Attribute Name	**Attribute of element**	**values / limitations**	**auth**	**pre**	**refund**	**erp**	**cancel**	**fulfill**	**txn_refund**
`currency`	`amount`	Currency is passed in the 3 character ISO 4217 Alphabetic format (Default is GBP) e.g. GBP, USD, EUR	O	O	O	O	O	O	X

Example XML for TxnDetails complex elements:

Card Transaction

```
<TxnDetails>
  <merchantreference>myreference123459988</merchantreference>
  <amount currency="EUR">146.99</amount>
</TxnDetails>
```

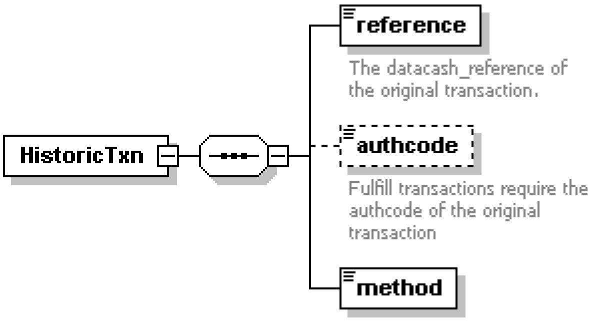
Historic Transaction

```
<TxnDetails>
 <amount>146.99</amount>
</TxnDetails>
```

B.1.1.4. HistoricTxn

This element is the direct equivalent of the `Card` element (for transactions using card details).

Element Name: `HistoricTxn`
Position: `Request.Transaction`


reference
The datacash_reference of the original transaction.
HistoricTxn
authcode
Fulfill transactions require the authcode of the original transaction
method

Elements of HistoricTxn					
Element Name	**description**	**values / limitations**	**cancel**	**fulfill**	**txn_refund**
`reference`	DataCash unique reference of the original transaction	must be a valid transaction	R	R	R
`authcode`	DataCash authorisation code of the original transaction	-	n/a	R	-

`method`	The transaction type	`fulfill`
		`txn_refund`
		`cancel`
			R	R	R

Example XML Request for HistoricTxn complex elements

```
<HistoricTxn>
  <reference>4100200039275407</reference>
  <authcode>641413</authcode>
  <method>fulfill</method>
</HistoricTxn>
```

B.1.1.4.1. Acquirer Specific Notes

For the Chase PaymentTech acquirer, an optional `reversal=”true”` attribute can be supplied within the `method` element when a cancel is being performed. This has the effect of attempting an online transaction reversal with the acquirer, in addition to excluding the transaction from the settlement file.

B.1.2. XML Examples Requests

B.1.2.1. Transactions with Card Details

Example XML Request for refund

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>*****</password>
  </Authentication>
  <Transaction>
    <CardTxn>
      <Card>
        <pan>633300*********1</pan>
        <expirydate>04/06</expirydate>
        <startdate>01/04</startdate>
      </Card>
      <method>refund</method>
    </CardTxn>
    <TxnDetails>
      <merchantreference>1000001</merchantreference>
      <amount currency="GBP">95.99</amount>
    </TxnDetails>
  </Transaction>
</Request>
```

If the `method` is changed, the same XML could also be used for `auth`, `pre` and `erp` transactions.

Example XML Request for pre with an authcode

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>*****</password>
  </Authentication>
  <Transaction>
    <CardTxn>
      <Card>
        <pan>444433*********1</pan>
        <expirydate>12/08</expirydate>
        <startdate>03/05</startdate>
      </Card>
      <authcode>439673</authcode>
      <method>pre</method>
    </CardTxn>
    <TxnDetails>
      <merchantreference>1000023</merchantreference>
      <amount currency="USD">1800.00</amount>
    </TxnDetails>
  </Transaction>
</Request>
```

B.1.2.2. Historic Transactions

Example XML Request for cancel

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <reference>4900200000000001</reference>
      <method>cancel</method>
    </HistoricTxn>
  </Transaction>
</Request>
```

Example XML Request for fulfill

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount>25.00</amount>
    </TxnDetails>
    <HistoricTxn>
      <reference>4900200000000001</reference>
      <authcode>A6</authcode>
      <method>fulfill</method>
    </HistoricTxn>
  </Transaction>
</Request>
```

Example XML Request for a txn_refund

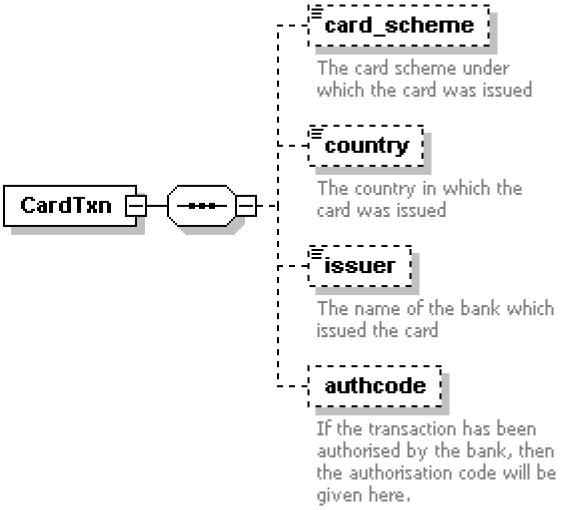
```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount>10.00</amount>
    </TxnDetails>
    <HistoricTxn>
      <reference>3100000088888888</reference>
      <method>txn_refund</method>
    </HistoricTxn>
  </Transaction>
</Request>
```

B.1.3. Schema Elements for Response

In addition to the elements covered in this section, Responses for this Service will also contain the general Response elements, as described in section A.1.2

B.1.3.1. CardTxn

Element Name: `CardTxn`
Position: `Response`


card_scheme
The card scheme under which the card was issued
country
The country in which the card was issued
CardTxn
issuer
The name of the bank which issued the card
authcode
If the transaction has been authorised by the bank, then the authorisation code will be given here.

Element Name	description	Values / limitations
`issuer`	The Card Issuing Bank, if known	
`card_scheme`	The Card Scheme, if known. Current values which may be returned are listed	• American Express • ATM • Carte Blanche • Diners Club • Discover • EnRoute • GE Capital • JCB • Laser • Maestro • Mastercard • Solo • Switch • Unknown • Visa • Visa Delta • VISA Electron • Visa Purchasing
`country`	The Country of Issue, if known	
`authcode`	Authorisation code for successful transactions. Additional information from the bank if declined or referred	

Example XML Response for CardTxn complex elements

```
<CardTxn>
  <authcode>01HD</authcode>
  <card_scheme>Visa</card_scheme>
  <country>France</country>
  <issuer>Credit Lyonnais</issuer>
</CardTxn>
```

B.1.4. XML Example Responses

There are various results that can be returned for transactions. These can be clustered into two groups:

- Bank Responses – the transaction is submitted to the bank
- Error codes – an error occurred which prevented the transaction from being sent to the bank

If the transaction is submitted to the bank, the bank can either accept, decline or refer the transaction. Please refer to the website for additional information on why transactions can be declined or referred.

Status	Meaning
1	The bank has authorised the transaction
7	The bank has declined or referred the transaction
All others	All other responses are error codes

As there are many different return codes, this document only contains examples for status 1 and 7. A complete list of Response Codes for this service is available on the website. The Support Centre also contains extensive examples for most error codes, with illustrations on how they would appear in both Reporting and an XML Response and also contains suggestions to prevent them occurring.

- Support Centre
- Website

B.1.4.1. Initial Transactions

Example XML Response for an authorised initial transaction

```
<Response>
  <CardTxn>
    <authcode>060642</authcode>
    <card_scheme>Switch</card_scheme>
    <country>United Kingdom</country>
    <issuer>HSBC</issuer>
  </CardTxn>
  <datacash_reference>3000000088888888</datacash_reference>
  <merchantreference>1000001</merchantreference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>1071567305</time>
</Response>
```

Example XML Response for a REFERRED transaction

```
<Response>
  <CardTxn>
    <authcode>CALL AUTH CENTRE</authcode>
    <card_scheme>Mastercard</card_scheme>
    <country>United Kingdom</country>
    <issuer>BARCLAYS BANK PLC</issuer>
  </CardTxn>
  <datacash_reference>3100000088888123</datacash_reference>
  <merchantreference>1000113</merchantreference>
  <mode>LIVE</mode>
  <reason>REFERRED</reason>
  <status>7</status>
  <time>1134692433</time>
</Response>
```

Example XML Response for a DECLINED transaction

```
<Response>
  <CardTxn>
    <authcode>DECLINED</authcode>
    <card_scheme>Mastercard</card_scheme>
    <country>United Kingdom</country>
  </CardTxn>
  <datacash_reference>4400200045583767</datacash_reference>
  <merchantreference>AA004630</merchantreference>
  <mode>TEST</mode>
  <reason>DECLINED</reason>
  <status>7</status>
  <time>1169223906</time>
</Response>
```

B.1.4.2. Historic Transactions

Example XML Response for a successful cancel

```
<Response>
  <datacash_reference>4900200000000001</datacash_reference>
  <merchantreference>4900200000000001</merchantreference>
  <mode>TEST</mode>
  <reason>CANCELLED OK</reason>
  <status>1</status>
  <time>1151567456</time>
</Response>
```

Example XML Response for a successful fulfil

```
<Response>
  <datacash_reference>3900200000000001</datacash_reference>
  <merchantreference>3900200000000001</merchantreference>
  <mode>LIVE</mode>
  <reason>FULFILLED OK</reason>
  <status>1</status>
  <time>1071567356</time>
</Response>
```

Example XML for a successful txn_refund

```
<Response>
  <datacash_reference>4000000088889999</datacash_reference>
  <HistoricTxn>
    <authcode>896876</authcode>
  </HistoricTxn>
  <merchantreference>4100000088888888</merchantreference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>1071567375</time>
</Response>
```

B.1.4.2.1. Acquirer Specific Notes

As mentioned in the Acquirer Specific Notes for the XML request, the following is returned as a response to a Chase PaymentTech cancellation where the `method` attribute `reversal=”true”` was supplied:

Example XML Response for a successfully reversed cancellation

```
<Response>
  <datacash_reference>4100900012345675</datacash_reference>
  <merchantreference>4100900012345675</merchantreference>
  <mode>TEST</mode>
  <reason reversal='success'>CANCELLED OK</reason>
  <status>1</status>
  <time>1071567390</time>
</Response>
```

This additional attribute indicates the results of the online reversal. This could be `‘success’`, `‘failed’` or `‘not attempted’`.

B.2. Line Item Detail

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/cpc/index.shtml

This service is utilised by sending a normal Credit and Debit Card Service Request with additional information. This section of documentation assumes the Credit and Debit Card Service has already been integrated and the reader is familiar with it. The Credit and Debit Card Service is described in section B.1

Please bear in mind that the amount field must be reconciled with the line items. The amount field should contain the gross amount, after tax, and (where appropriate) shipping and discounts. In order to compensate for rounding errors, a tolerance of one minor currency unit per line item element is allowed.

When using the Line Item Detail Service, the transaction methods `auth`, `pre`, `refund`, `erp` and `txn_refund` can all be submitted. However, if you are using the `pre` and `erp` methods, the full amount must be fulfilled – partial fulfills cannot be used. The `txn_refund` method can also be used, providing the full value of the original transaction is refunded in one step. If a partial `fulfill`, or `txn_refund` is performed, the amount would not reconcile with the line items.

B.2.1. Schema Elements for Request

In this section, the fields that can be presented with Line Item Detail transactions will be presented along with example XML for those fields

To process Visa or American Express Corporate Purchasing Cards, one can make use of the LineItemDetail schema element:

- `Request`
  - `Authentication` - section A.1.1.1
  - `Transaction`
    - `TxnDetails`
      - `LineItemDetail` - information relating to the whole order, section B.2.1.5
        - `Shipping` – details about the delivery, section B.2.1.2
          - `AddressDetail` – details about the delivery address, section B.2.1.1
        - `Items` –allows detail of each item to be passed, section B.2.1.3
          - `Item` – detailed information about a single item, section B.2.1.4. As the fields which can be provided depend upon the Acquiring Bank you are using, each field will be labelled with the following key:

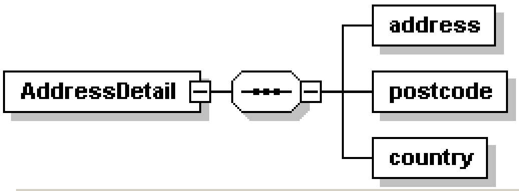
- *O* - Optional
- *R* - Required, field must be presented
- *X* - Excluded, presenting this field will cause the transaction to fail
- *M* - Mandatory if Available, if the information is available, it should be presented

B.2.1.1. AddressDetail

The fields within this element can only be presented for holders of American Express cards when the cardholder is billed in a different country from the supplier (a cross border transaction). If you are submitting such a transaction, it is recommended that these fields are provided.

Submitting this information for a non-American Express transaction, or an American Express non-cross border transaction will cause the transaction to be rejected.

Element Name	`AddressDetail`
Position	`Request.Transaction.TxnDetails.LineItemDetail.Shipping`
Children	`Yes`
Attributes	`Yes`



Elements of AddressDetail					
Attribute Name	**Description**	**Values/Limitations**	**Amex**	**Amex, cross border**	**NWS**
`side`	Designates whether the source or destination address	“source” or “destination”	X	R	X

Elements of AddressDetail					
Element Name	**Description**	**Values/Limitations**	**Amex**	**Amex, cross border**	**NWS**
`address`	A single line of the address. Between one and four such lines may be submitted, distinguished by the line attribute	Alphanumeric, up to 40 characters. line attribute should be a single digit between 1 and 4	X	R	X
`postcode`	The postal code for the address	Alphanumeric, up to 10 characters	X	R	X
`country`	ISO numeric code for the address	Numeric, 3-digit ISO code. For example, UK=826, France=250 etc.	X	R	X

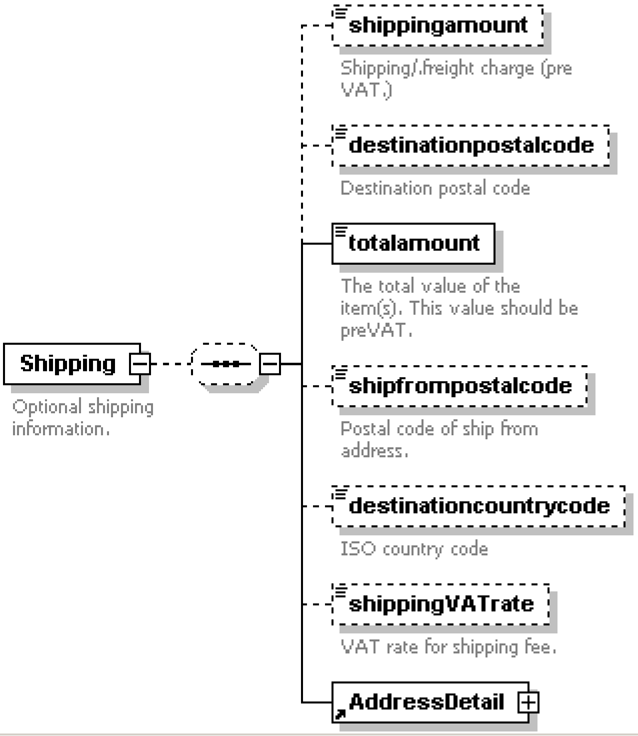
XML Example element for AddressDetail

```
<AddressDetail side=”source”>
  <address line=”1”>My Company</address>
  <address line=”2”>My Street</address>
  <address line=”3”>Mytown</address>
  <postcode>MP1 1AB</postcode>
  <country>826</country>
</AddressDetail>
<AddressDetail side=”destination”>
  <address line=”1”>Ma Maison</address>
  <address line=”2”>Ma Rue</address>
  <postcode>75123</postcode>
  <country>250</country>
</AddressDetail>
```

B.2.1.2. Shipping

Optional shipping information may be submitted for a CPC transaction. In the case of American Express, whilst this information may be submitted, it is not used by their systems at present (with the exception of the AddressDetail information discussed above). In particular, shipping should *not* be used in invoice reconciliation for American Express transactions.

Element Name	`Shipping`
Position	`Request.Transaction.TxnDetails.LineItemDetail`
Children	`Yes`
Attributes	`No`


Shipping
Optional shipping information.
shippingamount
Shipping/.freight charge (pre VAT.)
destinationpostalcode
Destination postal code
totalamount
The total value of the item(s). This value should be preVAT.
shipfrompostalcode
Postal code of ship from address.
destinationcountrycode
ISO country code
shippingVATrate
VAT rate for shipping fee.
AddressDetail

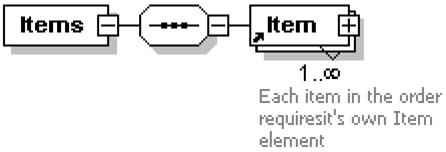
Attributes of Shipping				
Element Name	**Description**	**Values/Limitations**	**Amex**	**NWS**
`AddressDetail`	See section B.2.1.1			
`destinationcountrycode`	Country code of the destination	ISO Alphabetic 3 characters	O	O
`destinationpostalcode`	Postcode to send to	Alphanumeric up to 10 characters	O	O
`shipfrompostalcode`	Postcode of the source	Alphanumeric up to 10 characters	O	O
`shippingamount`	Net (pre-tax) cost of shipping	Numeric	O	O
`shippingVATrate`	Percentage rate at which shipping is taxed	Numeric	O	O

XML Example elements for Shipping

```
<Shipping>
  <destinationcountrycode>GBR</destinationcountrycode>
  <destinationpostalcode>EH1 1AB</destinationpostalcode>
  <shipfrompostalcode>SW19</shipfrompostalcode>
  <shippingamount>5.95</shippingamount>
  <shippingVATrate>17.5</shippingVATrate>
</Shipping>
```

B.2.1.3. Items

Element Name	`Items`
Position	`Request.Transaction.TxnDetails.LineItemDetail`
Children	`Yes`
Attributes	`No`



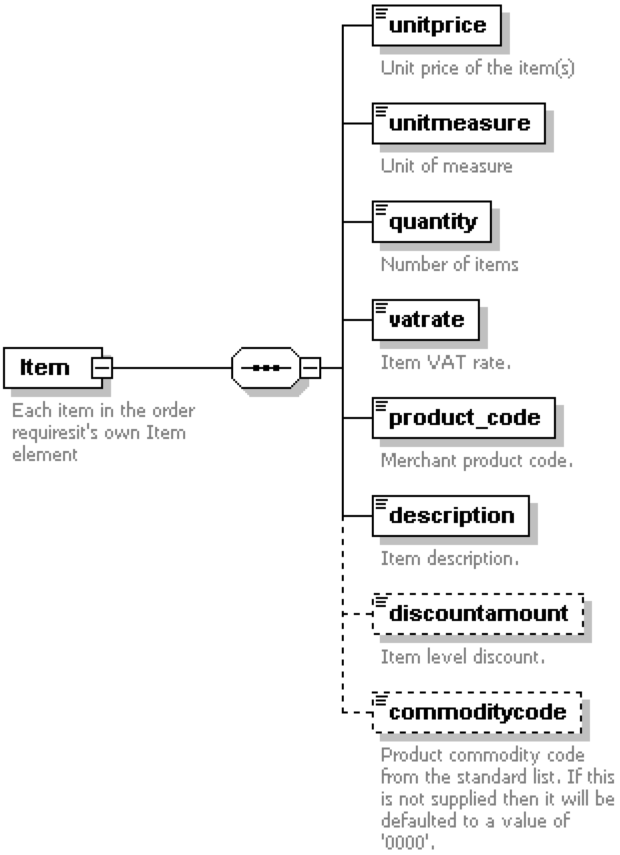
Elements of Items		
Element Name	**Description**	**Values/Limitations**
`Item`	An individual line item, as described below	At least one item element must be submitted per transaction

Example XML is shown in section B.2.1.4.

B.2.1.4. Item

Details of the individual line items are submitted here.

Element Name	`Item`
Position	`Request.Transaction.TxnDetails.LineItemDetail.Items`
Children	`Yes`
Attributes	`No`


Item
Each item in the order requiresit's own Item element
unitprice
Unit price of the item(s)
unitmeasure
Unit of measure
quantity
Number of items
vatrate
Item VAT rate.
product_code
Merchant product code.
description
Item description.
discountamount
Item level discount.
commoditycode
Product commodity code from the standard list. If this is not supplied then it will be defaulted to a value of '0000'.

Attributes of Item			
Element Name	**Description**	**Required**	**Values/Limitations**
commoditycode	The commodity code for this purchase	N	Four-digit commodity code. See section I.1 for a list
description	Description of the item	Y	Alphanumeric up to 26 characters (Barclays, Natwest), or 40 characters (Amex)
discountamount	The item-level discount for this item	N	Numeric. Not supported by American Express
product_code	Merchant's product code	N	Alphanumeric up to 12 characters
quantity	The number of units	Y	Numeric
totalamount	The total Net (pre-VAT) cost of the items	Y	Numeric. Must reconcile with quantity, unitprice and (where appropriate) discountamount
unitcost	The amount for a single unit	Y	Numeric
unitmeasure	The units in which the items are counted	Y	Alphanumeric up to 12 characters
vatrate	The item-level VAT rate for this item	Y	Numeric

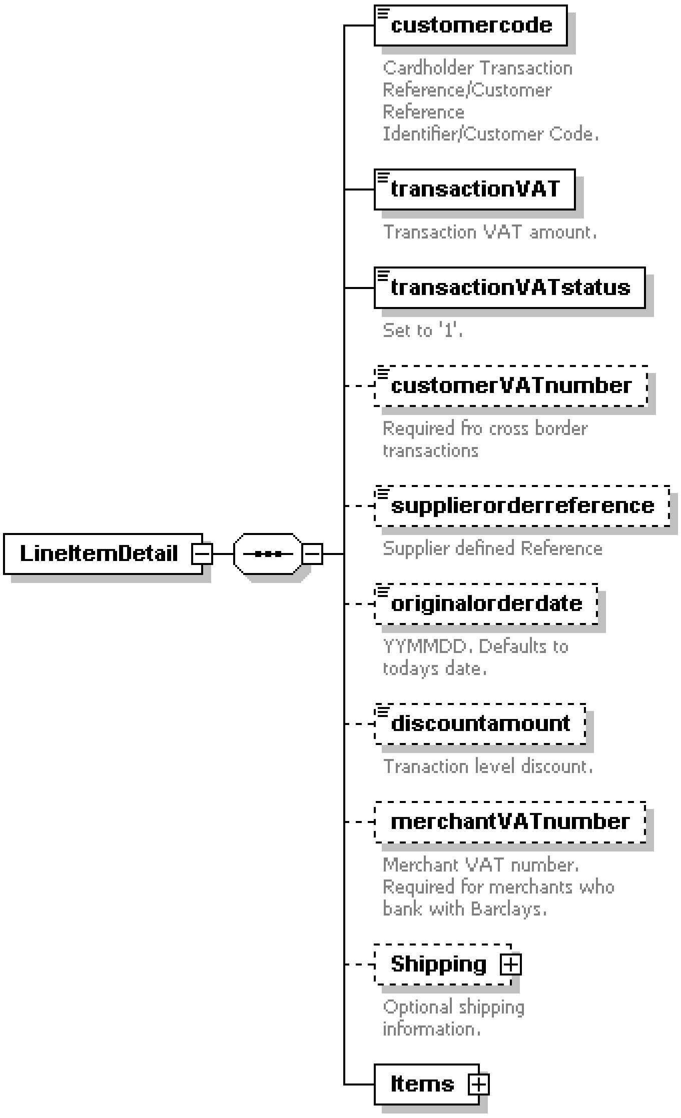
XML Example element for Items and Item

```
<Items>
  <Item>
    <commoditycode>4800</commoditycode>
    <description>Ring Binder</description>
    <unitmeasure>Box</unitmeasure>
    <unitprice>9.99</unitprice>
    <vatrate>17.5</vatrate>
    <quantity>1</quantity>
    <totalamount>9.99</totalamount>
  </Item>
  <Item>
    <commoditycode>4800</commoditycode>
    <description>Photocopier paper</description>
    <unitmeasure>Ream</unitmeasure
    <unitcost>5.00</unitcost>
    <vatrate>17.5</vatrate>
    <quantity>100</quantity>
    <totalamount>500</totalamount>
  </Item>
</Items>
```

B.2.1.5. LineItemDetail

This is where information relating to the order as a whole is submitted. Since each acquiring bank has different requirements, this information is listed with the following key:

Element Name	`LineItemDetail`
Position	`Request.Transaction.TxnDetails`
Children	`Y`
Attributes	`N`


LineItemDetail
customercode
Cardholder Transaction Reference/Customer Reference Identifier/Customer Code.
transactionVAT
Transaction VAT amount.
transactionVATstatus
Set to '1'.
customerVATnumber
Required fro cross border transactions
supplierorderreference
Supplier defined Reference
originalorderdate
YYMMDD. Defaults to todays date.
discountamount
Tranaction level discount.
merchantVATnumber
Merchant VAT number. Required for merchants who bank with Barclays.
Shipping
Optional shipping information.
Items

Elements of LineItemDetail				
Element Name	**Description**	**Values/Limitations**	**NWS**	**Amex**
customercode	Customer-supplied reference field	Alphanumeric, up to 16 characters	R	O
customernumber	Customer-supplied reference field	Alphanumeric, up to 17 characters	N	O
customerref1	Customer-supplied reference field	Alphanumeric, up to 20 characters	N	O
customerref2	Customer-supplied reference field	Alphanumeric, up to 20 characters	N	O
customerVATnumber	The customer's VAT number	Alphanumeric, up to 13 characters	O	O[1]
discountamount	Transaction-level discount amount	Numeric	O	O[2]
merchantVATnumber	The merchant's VAT number	Alphanumeric, up to 13 characters	O	R
originalinvoicenumber	The invoice number of the original transaction (for refunds)	Alphanumeric, up to 12 characters	O	O
originalorderdate	The date on which the order was placed	YYMMDD	O	O
supplierorderreference	The supplier's reference	Alphanumeric, up to 12 characters	O	N
transactionVAT	The total VAT amount for the transaction	Numeric. Must reconcile with the individual amounts and VAT rates of the items	R	R
transactionVATstatus	Should always be '1'	'1'	R	R

Notes:

[1] For American Express transactions, the customer VAT number is required for cross border transactions, and any transaction made by a Belgian merchant.

[2] Whilst the discountamount field may be submitted to American Express, it is currently ignored by their system. In particular, it must *not* be used in invoice reconciliation.

B.2.2. XML Example Request

XML Example Transaction for an order with two items

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <CardTxn>
      <method>auth</method>
      <Card>
        <pan>5700********0001</pan>
        <expirydate>10/05</expirydate>
      </Card>
    </CardTxn>
    <TxnDetails>
      <merchantreference>12345601</merchantreference>
      <amount>600.28</amount>
      <LineItemDetail>
        <customercode>CustCode123</customercode>
        <transactionVAT>90.29</transactionVAT>
        <transactionVATstatus>1</transactionVATstatus>
        <merchantVATnumber>7E6G415</merchantVATnumber>
        <Shipping>
          <shippingamount>5.99</shippingamount>
          <shippingVATrate>17.5</shippingVATrate>
        </Shipping>
        <Items>
          <Item>
            <commoditycode>4800</commoditycode>
            <description>Ring Binder</description>
            <unitmeasure>Box</unitmeasure>
            <unitprice>9.99</unitprice>
            <vatrate>17.5</vatrate>
            <quantity>1</quantity>
            <totalamount>9.99</totalamount>
          </Item>
          <Item>
            <commoditycode>4800</commoditycode>
            <description>Photocopier paper</description>
            <unitmeasure>Ream</unitmeasure
            <unitprice>5</unitprice>
            <vatrate>17.5</vatrate>
            <quantity>100</quantity>
            <totalamount>500</totalamount>
          </Item>
        </Items>
      </LineItemDetail>
    </TxnDetails>
  </Transaction>
</Request>
```

B.3. Airlines Transaction Records

This service is utilised by sending additional information in an otherwise normal Credit and Debit Card transaction. This section assumes that facility for processing Credit and Debit Card transactions has already been integrated and the reader is familiar with it. Details about the Credit and Debit Card Service are available in section B.1

The DPG will also accept Airline data in a `fulfill` transaction when using the two-stage transaction model. The location of the data in the request XML remains the same. At present this only applies to Airline data. See the examples on page 31.

This service is currently available for merchant using The Royal Bank of Scotland Group (inc Natwest Streamline, Natwest IMS, Ulster Bank, Clydesdale Bank, Yorkshire Bank) , Barclaycard Business and Omnipay.

B.3.1. Schema Elements for Request

In this section, the fields associated with Airlines transactions will be presented along with example XML for those fields.

To submit flight itinerary information, one should use the `AirlinesDetails` container element

- `Request`
    - `Authentication` – section A.1.1.1
    - `Transaction`
        - `TxnDetails`
            - `AirlinesDetails` - information relating to the whole order, section B.3.1.1
                - `Passenger` – details about an individual passenger, section B.3.1.2
                - `FlightLeg` – allows detail of each leg of the journey, section B.3.1.3

In the following tables, each field will be labelled with the following key:

- *O* - Optional
- *R* - Required, field must be presented

B.3.1.1. AirlinesDetails

The `AirlinesDetails` elements contains information relating to the whole order.

Element Name	`AirlinesDetails`
Position	`Request.Transaction.TxnDetails`
Children	`Yes`
Attributes	`No`

Elements of AirlinesDetails			
Element Name	**Description**	**Values/Limitations**	**Required**
`agencyname`	The POS agency name	Alphanumeric, up to 26 characters (for submission to DPG). For BMS merchants: the first 25 characters will be used at settlement	R
`agencycode`	The POS IATA number	Numeric up to 8 characters	R
`airlineplaninvoice`	The airline plan invoice	Alpha numeric, up to 6 characters	O
`airlineplannumber`	The airline plan number	Alphanumeric, up to 2 characters	O
`departurepoint`	The departure point	Three-character origination code	R
`Passenger`	See Section B.3.1.2		
`FlightLeg`	See Section B.3.1.3		

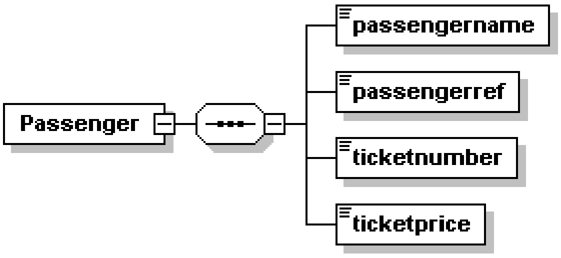
XML Example elements for AirlinesDetails

```
<AirlinesDetails>
  <agencyname>MyAirlinesAgency</agencyname>
  <agencycode>24681357</agencycode>
  <departurepoint>EDI</departurepoint>
  <Passenger>…</Passenger>
  <Passenger>…</Passenger>
  <FlightLeg number=”1”>…</FlightLeg>
  <FlightLeg number=”2”>…</FlightLeg>
  <FlightLeg number=”3”>…</FlightLeg>
</AirlinesDetails>

<AirlinesDetails>
  <agencyname>MyAirlinesAgency</agencyname>
  <agencycode>12345612</agencycode>
  <airplaninvoice>1234AN</airplaninvoice>
  <airplannumber>12</airplannumber>
  <departurepoint>LON</departurepoint>
  <Passenger>…</Passenger>
  <FlightLeg number=”1”>…</FlightLeg>
</AirlinesDetails>
```

B.3.1.2. Passenger

There must be at least one passenger per transaction. Where there is more than one person travelling, supplying multiple `Passenger` elements allows the entire transaction to be authorised at once, whilst maintaining the individual invoice information for each passenger. The sum of ticket prices for each passenger should match the `amount` field that forms part of the normal Credit and Debit Card information. This element may repeat as often as necessary.

Element Name	`Passenger`
Position	`Request.Transaction.TxnDetails.AirlinesDetails`
Children	`Yes`
Attributes	`No`



Elements of Passenger			
Element Name	**Description**	**Values/Limitations**	**Required**
`passengername`	The passenger's name	Alphanumeric, up to 25 characters	R
`passengerref`	A unique reference for the passenger	Alphanumeric, up to 17 characters for submission to DPG. For BMS merchants: the first 12 characters will be used at settlement	R
`ticketnumber`	Ticket number for the passenger. This may optionally be different for each passenger	14 characters. This usually consists of a 3-character prefix, 10 characters of ticket number and a single check digit at the end. The prefix characters and check digit may optionally be replaced by '0'	R
`ticketprice`	The ticket price for the individual passenger.		R

XML Example element for Passenger

```
<Passenger>
  <passengername>Joe Bloggs</passengername>
  <passengerref>12345678</passengerref>
  <ticketnumber>00024680135790</ticketnumber>
  <ticketprice>199.99</ticketprice>
</Passenger>
```

B.3.1.3. FlightLeg

Each transaction may contain details of up to four flight legs. These are each specified in their own `FlightLeg` container element. For a multi-flight leg journey: the point of departure for the first leg will be set to the value supplied in `AirlinesDetails.departurepoint`; the point of departure for subsequent legs will be taken to be the `destination` from the previous leg.

Element Name	`FlightLeg`
Position	`Request.Transaction.TxnDetailsAirlinesDetails`
Children	`Yes`
Attributes	`Yes`

Attributes of FlightLeg			
Attribute Name	**Description**	**Values/Limitations**	**Required**
`number`	The stage of the journey	1, 2, 3 or 4	R

Elements of FlightLeg			
Element Name	**Description**	**Values/Limitations**	**Required**
`carriercode`	Carrier code for this leg	Two character carrier code	R
`class`	Service class of this leg	Single alphanumeric character	R
`departtax`	Depature tax	Numeric, up to 12 digits including decimal point	O
`departuredate`	Date of departure for this leg	Date, as YYYYMMDD	R
`destination`	The destination for this leg	Three-character destination code	R
`farebasiscode`	The fare basis code	Alphanumeric, up to 6 charcters	O
`flight_number`	Flight number for this leg	Alphanumeric, up to 4 characters	O
`stopovercode`	The stop-over code	Alphanumeric, 1 character	O

XML Example elements for FlightLeg

```
<FlightLeg number=”1”>
  <carriercode>BA</carriercode>
  <class>1</class>
  <departuredate>20040730</departuredate>
  <destination>LHA</destination>
  <flight_number>1234</flight_number>
</FlightLeg>

<FlightLeg number=”1”>
  <carriercode>BA</carriercode>
  <class>1</class>
  <departtax>12.99</departtax>
  <departuredate>20040730</departuredate>
  <destination>LHA</destination>
  <farebasiscode>1223AB</farebasiscode>
  <flight_number>1234</flight_number>
</FlightLeg>
```

B.3.2. XML Example Requests

XML Example Transaction for a ticket purchase for single-leg, single-passenger flight

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <CardTxn>
      <method>auth</method>
      <Card>
        <pan>5700********0001</pan>
        <expirydate>10/05</expirydate>
      </Card>
    </CardTxn>
    <TxnDetails>
      <merchantreference>2345678</merchantreference>
      <amount>100.00</amount>
      <AirlinesDetails>
        <agencyname>MyAirlinesAgency</agencyname>
        <agencycode>24681357</agencycode>
        <departurepoint>EDI</departurepoint>
        <Passenger>
          <passengername>Joe Bloggs</passengername>
          <passengerref>12345601</passengerref>
          <ticketnumber>ABC09876543210</ticketnumber>
          <ticketprice>100.00</ticketprice>
        </Passenger>
        <FlightLeg number=”1”>
          <carriercode>BA</carriercode>
          <destination>LHA</destination>
          <departuredate>20040730</departuredate>
          <class>1</class>
        </FlightLeg>
      </AirlinesDetails>
    </TxnDetails>
  </Transaction>
</Request>
```

XML Example Transaction for a ticket purchase for multi-leg, multi-passenger flight

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <CardTxn>
      <method>auth</method>
      <Card>
        <pan>5700********0001</pan>
        <expirydate>10/05</expirydate>
      </Card>
    </CardTxn>
    <TxnDetails>
      <merchantreference>2345679</merchantreference>
      <amount>150.00</amount>
      <AirlinesDetails>
        <agencyname>MyAirlinesAgency</agencyname>
         <agencycode>24681357</agencycode>
        <departurepoint>EDI</departurepoint>
        <Passenger>
          <passengername>Joe Bloggs</passengername>
          <passengerref>12345602</passengerref>
          <ticketnumber>ABC09876543210</ticketnumber>
          <ticketprice>100.00</ticketprice>
        </Passenger>
        <Passenger>
          <passengername>Jane Bloggs</passengername>
          <passengerref>12345603</passengerref>
          <ticketnumber>ABC09876543210</ticketnumber>
          <ticketprice>50.00</ticketprice>
        </Passenger>
        <FlightLeg number=”1”>
          <carriercode>BA</carriercode>
          <destination>LHA</destination>
          <departuredate>20040730</departuredate>
          <class>1</class>
        </FlightLeg>
        <FlightLeg number=”2”>
          <carriercode>AB</carriercode>
          <destination>JFK</destination>
          <departuredate>20040731</departuredate>
          <class>1</class>
        </FlightLeg>
      </AirlinesDetails>
    </TxnDetails>
  </Transaction>
</Request>
```

XML Example Transaction supplying data in a “fulfill” transaction

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <reference>4100200039275407</reference>
      <method>fulfill</method>
    </HistoricTxn>
    <TxnDetails>
      <AirlinesDetails>
        <agencyname>MyAirlinesAgency</agencyname>
        <agencycode>24681357</agencycode>
        <departurepoint>EDI</departurepoint>
        <Passenger>
          <passengername>Joe Bloggs</passengername>
          <passengerref>12345601</passengerref>
          <ticketnumber>ABC09876543210</ticketnumber>
          <ticketprice>100.00</ticketprice>
        </Passenger>
        <FlightLeg number=”1”>
          <carriercode>BA</carriercode>
          <destination>LHA</destination>
          <departuredate>20040730</departuredate>
          <class>1</class>
        </FlightLeg>
      </AirlinesDetails>
    </TxnDetails>
  </Transaction>
</Request>
```

B.4. Merchant Narrative

The Merchant Narrative Service allows merchants to specify the narrative presented on the cardholders statement on a per transaction basis, rather than basing the narrative on static data held by the acquiring bank.

B.4.1. Schema Elements for Request

Element Name: `MerchantNarrative`
Position: `Request.Transaction.CardTxn`

Elements of MerchantNarrative			
Element Name	**description**	**values / limitations**	**required**
`narrative1`	Merchant Narrative element 1	Maximum 26 characters. The following characters are permitted: Alpha numerics, !”%&’()*+,-./:;<=>?_ Spaces permitted however leading and trailing whitespace will be trimmed	M
`narrative2`	Merchant Narrative element 2	Maximum 26 characters. The following characters are permitted: Alpha numerics, !”%&’()*+,-./:;<=>?_ Spaces permitted however leading and trailing whitespace will be trimmed	M

Either `narrative1`, `narrative2`, or both can be present in the parent `MerchantNarrative` element, however an empty `MerchantNarrative` element will not be permitted.

The `MerchantNarrative` element can only be supplied for the following Bank Card transaction methods:

- auth
- refund
- pre
- erp

The Merchant Narrative service is not compatiable with the Airlines Transaction Record service.

B.4.2. XML Example Request

Example Merchant Narrative Request

```
<Request>

  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <merchantreference>12345601</merchantreference>
      <amount currency='GBP'>1000.00</amount>
    </TxnDetails>

    <CardTxn>
      <method>auth</method>
      <Card>
        <pan>XXXXXXXXXXXXXXXX</pan>
        <expirydate>XX/XX</expirydate>
      </Card>
      <MerchantNarrative>
        <narrative1>line1</narrative1>
        <narrative2>line2</narrative2>
      </MerchantNarrative>
    </CardTxn>

  </Transaction>
</Request>
```

C. Repeat Card Payments

C.1. Pre-Registered Cards

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/recurring/pre_reg_cards.shtml

C.1.1. Schema Elements for Request

In this section the required fields for each transaction type will be presented, along with example XML for those fields. If a transaction type is specified in the XML, this is highlighted to indicate that other transaction types can be used in its place.
This document assumes that the Credit and Debit Card Service has already been integrated and the reader is familiar with it.

The data for the Pre-Registered Card Service is passed in these distinct places in the schema:

- `Request`
    - `Authentication` - section A.1.1.1
    - `Transaction`
        - `CardTxn` - the type and datacash_reference of the original transaction, section C.1.1.1.
        - `Secure` – to be used if the 3-D Secure check using a 3rd Party MPI is required, section D.5.1.1
        - `Card`
            - `Cv2Avs` – to be used if the CV2AVS check is required, section D.1.1.1
        - `TxnDetails` - section B.1.1.3
            - `ThreeDSecure` – to be used if the 3-D Secure check using the DataCash MPI is required, section D.4.1.1

C.1.1.1. CardTxn

The details of the initial Credit and Debit Card transaction, plus the method to be used for the Pre-Registered transaction are submitted in the CardTxn element. If the CV2AVS check is being used, this information is also presented within this element::

Element Name:	CardTxn
Position(s)	Request.Transaction

Elements of CardTxn		
Element Name	**description**	**values / limitations**
Card	To be used if the CV2AVS check is required.	Please refer to section D.1.1.1
method	the transaction type	auth pre refund erp
card_details	the datacash_reference of the initial transaction	initial transaction must have traken place within one year ago, and have a status=1

Attributes for Elements of CardTxn		
Attribute Name	**Attribute of element**	**value**
type	card_details	preregistered

Example XML for CardTxn elements

```
<CardTxn>
 <method>pre</method>
 <card_details type="preregistered">2185999900000001</card_details>
</CardTxn>
```

Example XML for CardTxn elements

```
<CardTxn>
 <method>auth</method>
 <card_details type="preregistered">2185999900000008</card_details>
 <Card>…</Card>
</CardTxn>
```

C.1.2. XML Examples Requests

Example XML Request for pre

```
<Request>
  <Authentication>
    <password>******</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123402</merchantreference>
      <amount currency="USD">249.99</amount>
    </TxnDetails>
    <CardTxn>
      <method>pre</method>
      <card_details type="preregistered">2185999900000001
         </card_details>
    </CardTxn>
  </Transaction>
</Request>
```

If the `method` is changed, the same XML could also be used for `auth`, `refund` and `erp` transactions.

For examples with the CV2AVS check, please refer to section D.1.2. Examples using 3-D Secure are available in section D.4.2.1 (DataCash MPI) and D.5.2 (3rd Party MPI).

C.1.3. Schema Elements for Response

The Pre-Registered Card Service will return same structure of Response as the Credit and Debit Card Service. Please refer to section B.1.3 of this document.

C.1.4. XML Examples Responses

The Pre-Registered Card Service will return same structure of Response as the Credit and Debit Card Service. Examples of successful, declined and referred responses are given in the Credit and Debit Card Service section B.1.4 of this document

The initial transaction is older than one year, does not exist or was unsuccessful

```
<Response>
  <datacash_reference>2185999900005678</datacash_reference>
  <merchantreference>123402</merchantreference>
  <reason>Prereg: Invalid reference</reason>
  <status>250</status>
  <time>1071567305</time>
</Response>
```

The account is not currently configured for the Preregistered Card Service

```
<Response>
  <datacash_reference>2185999900005679</datacash_reference>
  <information>This vTID is not configured to process
     pre-registered card transactions.</information>
  <merchantreference>123403</merchantreference>
  <reason>Prereg: Merchant Not Subscribed</reason>
  <status>251</status>
  <time>1074692433</time>
</Response>
```

C.2. Capture Method Recurring Transactions

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/recurring/capturemethod.shtml

This service is utilised by sending a normal Credit and Debit Card Service Request with additional information. This section of the documentation assumes the reader is familiar with it. The Credit and Debit Card Service is described in section B.1

C.2.1. Schema Elements for Request

In this section the required fields for each capture method type will be presented, along with example XML for those fields. If a capture method is specified in the XML, this is highlighted to indicate that other capture method types can be used in its place.

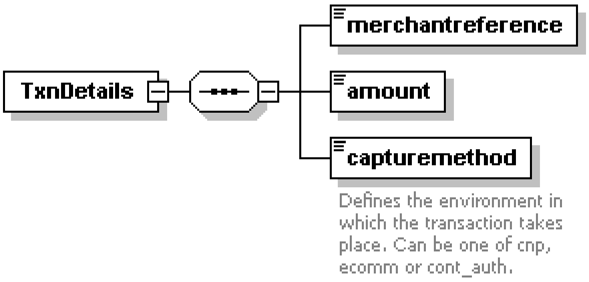
The data for the Capture Method Recurring Transactions Service is passed in three distinct places in the schema:

- `Request`
    - `Authentication` - section A.1.1.1
    - `Transaction`
        - `CardTxn` - - the type and authorisation code, see section B.1.1.2
            - `Card` - this contains all the information about the Card, section B.1.1.1
        - `TxnDetails` - contains details of the transaction, see section C.2.1.1

When using this Service, a single extra element is provided within the `TxnDetails` parent for transactions.

C.2.1.1. TxnDetails

Element Name:	TxnDetails
Position(s)	Request.Transaction


TxnDetails
merchantreference
amount
capturemethod
Defines the environment in which the transaction takes place. Can be one of cnp, ecomm or cont_auth.

Elements of TxnDetails		
Element Name	**description**	**values / limitations**
merchantreference	Please refer to section B.1.1.3	
amount		
capturemethod	Specifies the environment of transaction	ecomm cnp cont_auth

Please note that refund transaction types (including `erp` and `txn_refund`) cannot be processed in a `cont_auth` environment. These should instead be processed using your normal environment flag.

Example XML Elements

```
 <TxnDetails>
   <merchantreference>123401</merchantreference>
   <amount currency="GBP">189.00</amount>
   <capturemethod>cont_auth</capturemethod>
 </TxnDetails>

<TxnDetails>
   <merchantreference>123402</merchantreference>
   <amount currency="GBP">52.82</amount>
   <capturemethod>cnp</capturemethod>
 </TxnDetails>
```

C.2.1.1.1. NatWest Merchants

If you are processing transactions through NatWest Streamline, the following additional information should also be presented:

Attributes of child elements in TxnDetails			
Attribute Name	**Element Name**	**description**	**values / limitations**
`Streamline_CA_environment`	`capturemethod`	Specifies the environment of original transaction	`ecomm` `cnp`

Example XML Elements for NatWest merchants

```
<TxnDetails>
   <merchantreference>123403</merchantreference>
   <amount currency="GBP">249.99</amount>
   <capturemethod>ecomm</capturemethod>
 </TxnDetails>

 <TxnDetails>
   <merchantreference>123403</merchantreference>
   <amount currency="USD">249.99</amount>
   <capturemethod Streamline_CA_environment="ecomm">
       cont_auth</capturemethod>
 </TxnDetails>
```

C.2.2. XML Example Request

C.2.2.1. Ecomm and CNP Transactions

Example XML Request for an initial ecomm transaction

```
<Request>
  <Authentication>
    <password>******</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123404</merchantreference>
      <amount currency="GBP">100.00</amount>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>
    <CardTxn>
      <method>pre</method>
      <Card>
        <expirydate>02/06</expirydate>
        <issuenumber>01</issuenumber>
        <startdate>0199</startdate>
        <pan>444433*********1</pan>
      </Card>
    </CardTxn>
  </Transaction>
</Request>
```

If the `capturemethod` above is changed, the same XML could also be used for `cnp` transactions. When a ‘refund of an existing transaction’ (`txn_refund)` is being performed, the DPG will automatically use the same `capturemethod` as the existing transaction. However, if the existing transaction is a `cont_auth` transaction, the `capturemethod` should be explicitly set to the normal (i.e. `ecomm` or `cnp`) value.

Example XML Request for a txn_refund

```
<Request>
  <Authentication>
    <password>******</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123404</merchantreference>
      <amount currency="GBP">100.00</amount>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>
    <HistoricTxn>
      <method>txn_refund</method>
      <reference>4900200000000001</reference>
    </HistoricTxn>
  </Transaction>
</Request>
```

C.2.2.2. Cont_Auth Transactions

C.2.2.2.1. Non-NatWest Merchants

To send a cont_auth transaction, non-NatWest merchants can use the same XML Request as shown in section C.2.2.1, merely by changing the value of the capturemethod.

C.2.2.2.2. NatWest Merchants

Example XML Request for a NatWest recurring transaction

```
<Request>
  <Authentication>
    <password>******</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123404</merchantreference>
      <amount currency="GBP">100.00</amount>
      <capturemethod Streamline_CA_environment="cnp”>
        cont_auth</capturemethod>
    </TxnDetails>
    <CardTxn>
      <method>pre</method>
      <Card>
        <expirydate>02/06</expirydate>
        <issuenumber>01</issuenumber>
        <startdate>0199</startdate>
        <pan>444433*********1</pan>
      </Card>
    </CardTxn>
  </Transaction>
</Request>
```

C.3. Historic Recurring Transactions

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/recurring/historic.shtml

This section of the documentation assumes the reader is familiar with the Credit and Debit Card Service, which is discussed in section B.1.

C.3.1. Schema Elements for Request

In this section the required fields for each transaction type will be presented, along with example XML for those fields

For Recurring Account Setups

To authorise the first payment and setup an account for a card, information needs to be collected and set in the following places within the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `TxnDetails` – contains details of the transaction, section B.1.1.3
      - `CardTxn`
        - `Card` - this element contains the details about the card, for setups only. These details are the same as for one-off payments, which are covered in section B.1.1.1
    - `ContAuthTxn` – section C.3.1.1

For the Repeat Payments

Once the account has been setup, repeat payments can be taken from it.

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `TxnDetails` – contains details of the transaction, section B.1.1.3
  - `HistoricTxn` – section C.3.1.2

For Cancellations of Payments and Accounts

Cancellations of accounts and payments are performed using the same data as for normal Bank Card cancellations:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `HistoricTxn` – section C.3.1.2

C.3.1.1. ContAuthTxn

This element should be presented with setup and repeat payment Requests. It has one attribute and no children.

Element Name:	`ContAuthTxn`
Position	`Request.Transaction`

ContAuthTxn
Type attribute must be set to either "setup" or "historic"

Attributes of ContAuthTxn			
Attributes	**Attribute of Element**	**description**	**values / limitations**
`type`	`ContAuthTxn`	Indicates whether the transaction is a setup or a repeat payment	`setup` `historic`

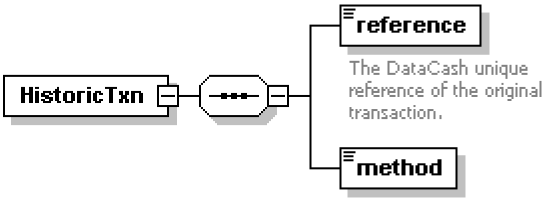
Example XML ContAuthTxn elements

```
<ContAuthTxn type=”setup”></ContAuthTxn>

<ContAuthTxn type=”setup” />

<ContAuthTxn type=”historic” />
```

C.3.1.2. HistoricTxn

This element is required for the repeat payments and cancellations of payments/accounts. It should not be presented with the setup transaction.

Element Name: HistoricTxn
Position Request.Transaction



Elements of HistoricTxn		
Element Name	**description**	**values / limitations**
method	Indicates the transaction type	pre auth cancel
reference	For Payments: the datacash_reference of the account from which the payment is to be taken	Must be a valid account
	For Cancellations: the datacash_reference of the account or payment to be cancelled	Must be a valid account or payment

Example XML HistoricTxn elements

```
<HistoricTxn>
  <method>auth</method>
  <reference>4700200040912890</reference>
</HistoricTxn>

<HistoricTxn>
  <method>cancel</method>
  <reference>4700200040919424</reference>
</HistoricTxn>
```

C.3.2. Example XML Requests

Example XML Request for account setup. The initial transaction will be processed as ecomm

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>mypasswd</password>
  </Authentication>
  <Transaction>
    <ContAuthTxn type="setup" />
    <TxnDetails>
      <merchantreference>385036349305556</merchantreference>
      <amount currency="GBP">1001.00</amount>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>
    <CardTxn>
      <Card>
        <expirydate>01/06</expirydate>
        <issuenumber>1</issuenumber>
        <startdate>0199</startdate>
        <pan>444433*********1</pan>
      </Card>
      <method>auth</method>
    </CardTxn>
  </Transaction>
</Request>
```

Example XML Request for account setup. The initial transaction will be processed as cnp

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>mypasswd</password>
  </Authentication>
  <Transaction>
    <CardTxn>
      <Card>
        <expirydate>01/06</expirydate>
        <pan>444433*********1</pan>
      </Card>
      <method>pre</method>
    </CardTxn>
    <ContAuthTxn type="setup" />
    <TxnDetails>
      <merchantreference>385036</merchantreference>
      <amount currency="AUD">56.52</amount>
      <capturemethod>cnp</capturemethod>
    </TxnDetails>
  </Transaction>
</Request>
```

Example XML Request for a repeat payment

```
<Request>
  <Transaction>
    <ContAuthTxn type="historic" />
    <TxnDetails>
      <merchantreference>3851231</merchantreference>
      <capturemethod>cont_auth</capturemethod>
      <amount currency="GBP">18.50</amount>
    </TxnDetails>
    <HistoricTxn>
      <reference>4500200040925092</reference>
      <method>auth</method>
    </HistoricTxn>
  </Transaction>
  <Authentication>
    <client>99000001</client>
    <password>mypasswd</password>
  </Authentication>
</Request>
```

Example XML Request to cancel a payment or account

```
<Request>
  <Authentication>
    <password>mypasswd</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>cancel</method>
      <reference>4800200040644359</reference>
    </HistoricTxn>
  </Transaction>
</Request>
```

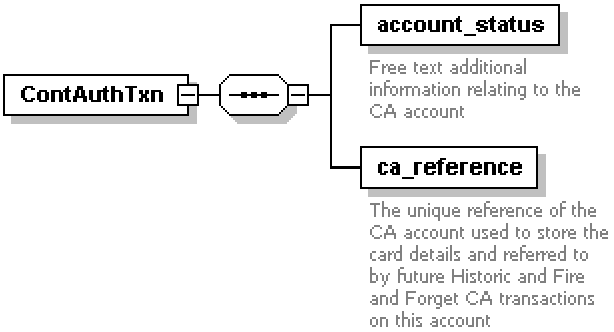
C.3.3. Schema Elements for Response

In addition to the elements covered in this section, responses for this service will contain the general response elements outlined in section A.1.2.

The CardTxn element will also be present, as described in section B.1.3.1.

C.3.3.1. ContAuthTxn

Element Name:	ContAuthTxn
Position	Response


account_status
Free text additional information relating to the CA account
ContAuthTxn
ca_reference
The unique reference of the CA account used to store the card details and referred to by future Historic and Fire and Forget CA transactions on this account

Elements of ContAuthTxn	
Element Name	**description**
ca_reference	The reference number of the account
account_status	Provides additional information about the account status

Example ContAuthTxn element

```
<ContAuthTxn>
  <account_status>Account setup. CNP transaction
            sent successfully</account_status>
  <ca_reference>4500200040151462</ca_reference>
</ContAuthTxn>
```

C.3.4. Example Responses

C.3.4.1. Successful Responses

Example XML Response for a successful setup

```
<Response>
  <CardTxn>
    <authcode>100001</authcode>
    <card_scheme>Mastercard</card_scheme>
    <country>United Kingdom</country>
  </CardTxn>
  <ContAuthTxn>
    <account_status>Account setup. ECOMM transaction sent
                    successfully</account_status>
    <ca_reference>4700200040641889</ca_reference>
  </ContAuthTxn>
  <datacash_reference>4700200040641890</datacash_reference>
  <merchantreference>setupreference12345680</merchantreference>
  <mode>…</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Response for a successful repeat payment

```
<Response>
  <CardTxn>
    <authcode>779099</authcode>
    <card_scheme>VISA</card_scheme>
    <country>United Kingdom</country>
  </CardTxn>
  <ContAuthTxn>
    <account_status>Using account ref 4500200040925092. CONT_AUTH
                    transaction complete</account_status>
  </ContAuthTxn>
  <datacash_reference>4000200040925117</datacash_reference>
  <merchantreference>385036359953704</merchantreference>
  <mode>…</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Response for a successfully cancelled account

```
<Response>
  <datacash_reference>4400200040943162</datacash_reference>
  <information>Continuous authority account cancelled</information>
  <merchantreference>4500200040925092</merchantreference>
  <mode>…</mode>
  <reason>CANCELLED</reason>
  <status>1</status>
</Response>
```

C.3.4.2. Failed or Rejected Responses

Example XML Response for a rejected setup. The initial payment was declined, so the account could not be set up.

```
<Response>
  <CardTxn>
    <authcode>DECLINED</authcode>
    <card_scheme>VISA Delta</card_scheme>
    <country>United Kingdom</country>
  </CardTxn>
  <ContAuthTxn>
    <account_status>Account not setup. CNP transaction was not
                    successfully authed</account_status>
  </ContAuthTxn>
  <datacash_reference>4500200040641890</datacash_reference>
  <merchantreference>…</merchantreference>
  <mode>…</mode>
  <reason>DECLINED</reason>
  <status>7</status>
  <time>…</time>
</Response>
```

Example XML Response where a recurring payment Request was declined

```
<Response>
  <CardTxn>
    <authcode>NOT AUTHORISED</authcode>
    <card_scheme>American Express</card_scheme>
  </CardTxn>
  <ContAuthTxn>
    <account_status>Using account ref 3900200040457927.
              CONT_AUTH transaction complete</account_status>
  </ContAuthTxn>
  <datacash_reference>3800200040465541</datacash_reference>
  <merchantreference>…</merchantreference>
  <mode>LIVE</mode>
  <reason>DECLINED</reason>
  <status>7</status>
  <time>…</time>
</Response>
```

Example XML Response where the account could not be found or has already been cancelled

```
<Response>
  <datacash_reference>4500200040373397</datacash_reference>
  <information>Failed to find live CA account with
                 reference=4700200040579l371</information>
  <merchantreference>historicreference12345690</merchantreference>
  <reason>Failed to find corresponding CA account</reason>
  <status>192</status>
  <time>…</time>
</Response>
```

C.4. Fire and Forget Recurring Transactions

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/recurring/cc_fireandforget.shtml

This section of the documentation assumes the reader is familiar with the Credit and Debit Card Service, which is discussed in section B.1.

C.4.1. Schema Elements for Request

In this section the required fields for each transaction type will be presented, along with example XML for those fields. As not all fields are mandatory, the following key will be used:

- R - Required
- O - Optional

For Recurring Account Setups

The information required to set up an account is passed in several places in the schema:

- `Request`
    - `Authentication` – section A.1.1.1
    - `Transaction`
        - `TxnDetails` – contains details of the transaction, section B.1.1.3
        - `CardTxn`
            - `Card` - contains the details about the card. These details are the same as for one-off payments, which are covered in section B.1.1.1
        - `ContAuthTxn` – contains all the information about the payments which are to be taken from the account, section C.4.1.1
            - `FirstPayment` – enables an initial payment to be specified, section C.4.1.2
            - `LastPayment` – enables a final payment to be specified, section C.4.1.3

For Cancellations of Payments and Accounts

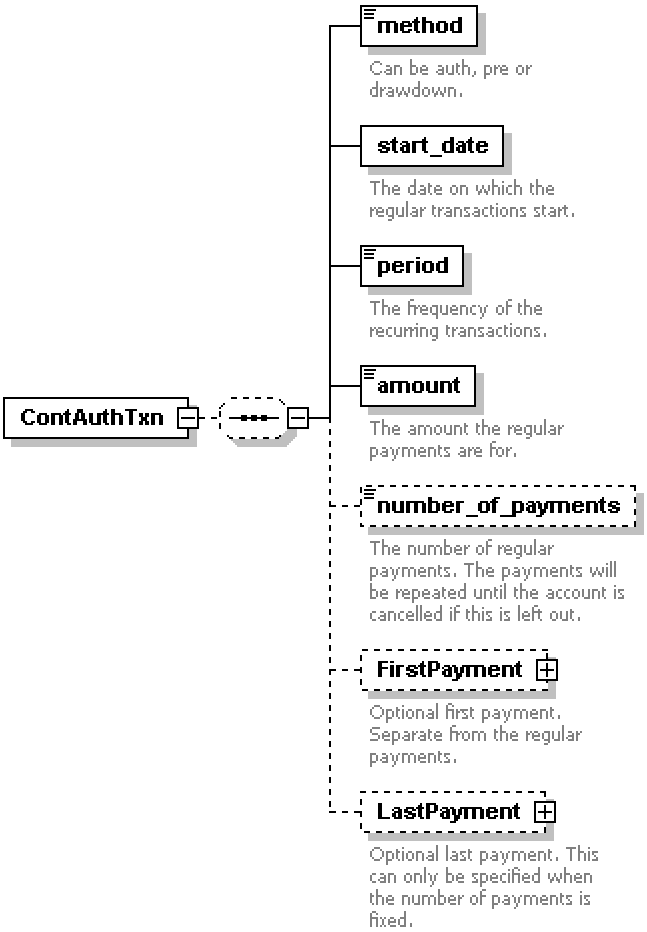
Account and Payment Cancellations are performed using the same data as for normal Bank Card cancellations:

- `Request`
    - `Authentication` – section A.1.1.1
    - `Transaction`
        - `HistoricTxn` – details the transaction method cancel, and the reference number of the payment/account to cancel, section B.1.1.4

C.4.1.1. ContAuthTxn

This element must be presented when setting up an account

Element Name:	`ContAuthTxn`
Position	`Request.Transaction`



method

Can be auth, pre or drawdown.

start_date

The date on which the regular transactions start.

period

The frequency of the recurring transactions.

amount

The amount the regular payments are for.

number_of_payments

The number of regular payments. The payments will be repeated until the account is cancelled if this is left out.

FirstPayment

Optional first payment. Separate from the regular payments.

LastPayment

Optional last payment. This can only be specified when the number of payments is fixed.

Elements of ContAuthTxn

Element Name	description	values / limitations	
method	Indicates the transaction type	pre auth	R
start_date	The date on which the first regular payment will be taken. Subsequent payments will be based from this date. Must be at least the day after the transaction is sent.	dd/mm/yyyy. Must be at least one day ahead of either the current date or the FirstPayment date	R
period	The frequency of payments	weekly monthly quarterly annual	R
amount	The value of each regular payment		R
number_of_payments	Used to restrict the number of regular payments to be collected. Note: this is not the total number of payments if first and/or last payments have been set.	Must be an integer	O
FirstPayment	Details of any initial payment to be made. This is in addition to the regular payments. See section C.4.1.2		O
LastPayment	Details of any final payment to be made. This is in addition to the regular payments. See section C.4.1.3		O

Attributes of ContAuthTxn

Attributes	Attribute of Element	description	values / limitations
currency	amount	Currency in the ISO 4217 Alphabetic format. E.g. GBP, USD, AUD	Will default to GBP if not specified

Example ContAuthTxn element. Monthly payments will be taken until the account is cancelled

```
<ContAuthTxn>
  <method>auth</method>
  <start_date>31/03/2005</start_date>
  <period>monthly</period>
  <amount>25.00</amount>
</ContAuthTxn>
```

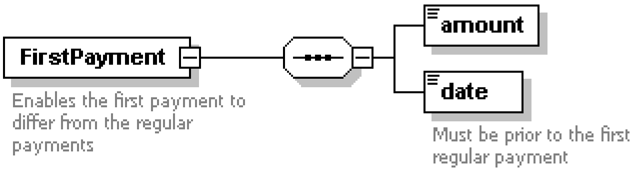
Example ContAuthTxn element with first and last payments

```
<ContAuthTxn>
  <method>auth</method>
  <start_date>10/02/2003</start_date>
  <period>weekly</period>
  <number_of_payments>5</number_of_payments>
  <amount>10.00</amount>
  <FirstPayment>…</FirstPayment>
  <LastPayment>…</LastPayment>
</ContAuthTxn>
```

C.4.1.2. FirstPayment

This optional element enables an initial payment to be taken which varies in both value and timing from the regular transactions. If this element is presented for a transaction, both it’s children must be populated.

Element Name:	`FirstPayment`
Position	`Request.Transaction.ContAuthTxn`


FirstPayment
Enables the first payment to differ from the regular payments
amount
date
Must be prior to the first regular payment

Elements of FirstPayment		
Element Name	**description**	**values / limitations**
`date`	The date the first payment is to be taken	`dd/mm/yyyy` Must be at least one day ahead of the current date
`amount`	The value of the first payment. This may vary from the regular payments. Currency attribute may be specified if required	

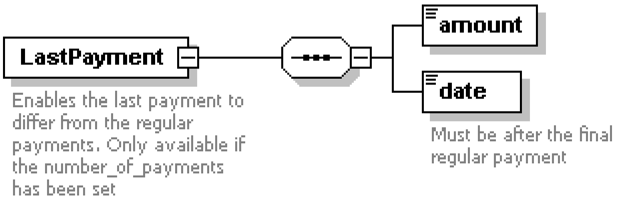
Example FirstPayment element

```
<FirstPayment>
  <amount>30.00</amount>
  <date>29/06/2005</date>
</FirstPayment>
```

C.4.1.3. LastPayment

This optional element enables a final payment to be taken which varies in both value and timing from the regular transactions. The element may only be presented if the `number_of_payments` element has been specified (section C.4.1.1). If this element is presented for a transaction, both it's children must be populated.

Element Name:	`LastPayment`
Position	`Request.Transaction.ContAuthTxn`


LastPayment
Enables the last payment to differ from the regular payments. Only available if the number_of_payments has been set
amount
date
Must be after the final regular payment

Elements of LastPayment		
Element Name	**description**	**values / limitations**
`date`	The date the last payment is to be taken	`dd/mm/yyyy` Must be at least one day after the last regular payment
`amount`	The value of the last payment. This may vary from the regular payments. Currency attribute may be specified if required	

Example LastPayment element

```
<LastPayment>
  <amount>20.00</amount>
  <date>12/01/2016</date>
</LastPayment>
```

C.4.2. Example XML Requests

Example Request for an initial charge of £7.50, followed by regular weekly payments of £5 until a cancellation Request is received

```
<Request>
  <Authentication>
    <client>99106700</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <CardTxn>
      <Card>
        <expirydate>01/06</expirydate>
        <pan>444433*********1</pan>
      </Card>
    </CardTxn>
    <ContAuthTxn>
      <method>auth</method>
      <start_date>15/07/2005</start_date>
      <period>weekly</period>
      <amount>5.00</amount>
      <FirstPayment>
        <amount>7.50</amount>
        <date>09/07/2005</date>
      </FirstPayment>
    </ContAuthTxn>
    <TxnDetails>
      <merchantreference>383914410763889</merchantreference>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>
  </Transaction>
</Request>
```

Example Request for nine quarterly payments of £69.99

```
<Request>
  <Authentication>
    <client>99106700</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <CardTxn>
      <Card>
        <expirydate>06/13</expirydate>
        <pan>3434*********3</pan>
      </Card>
    </CardTxn>
    <ContAuthTxn>
      <method>auth</method>
      <start_date>31/07/2005</start_date>
      <period>quarterly</period>
      <amount currency=”GBP”>69.99</amount>
      <number_of_payments>9</number_of_payments>
    </ContAuthTxn>
    <TxnDetails>
```

```
      <merchantreference>annual0000432432a</merchantreference>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>
  </Transaction>
</Request>
```

Example Request for an initial charge of $30, twelve regular payments of $10 on the first of each month and a final payment of $20

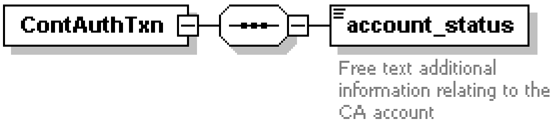
```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>mypasswd</password>
  </Authentication>
  <Transaction>
    <CardTxn>
      <Card>
        <expirydate>01/06</expirydate>
        <pan>444433*********1</pan>
      </Card>
    </CardTxn>
    <ContAuthTxn>
      <method>auth</method>
      <start_date>01/10/2005</start_date>
      <period>monthly</period>
      <amount currency=”USD”>10.00</amount>
      <number_of_payments>12</number_of_payments>
      <FirstPayment>
        <amount currency=”USD”>30.00</amount>
        <date>01/09/2005</date>
      </FirstPayment>
      <LastPayment>
        <amount currency=”USD”>20.00</amount>
        <date>31/10/2006</date>
      </LastPayment>
    </ContAuthTxn>
    <TxnDetails>
      <merchantreference>382684637615741</merchantreference>
      <capturemethod>cnp</capturemethod>
    </TxnDetails>
  </Transaction>
</Request>
```

C.4.3. Schema Elements for Response

In addition to the elements covered in this section, responses for this service will contain the general response elements outlined in section A.1.2.

C.4.3.1. ContAuthTxn

Element Name:	ContAuthTxn
Position	Response



Elements of ContAuthTxn	
Element Name	**description**
account_status	Returns information about the status of the account

Example XML element ContAuthTxn

```
<ContAuthTxn>
  <account_status>Account setup. First transaction due to be
                  batched on 30/06/2005</account_status>
</ContAuthTxn>
```

C.4.4. Example XML Responses

Example XML Response for an account which has been successfully set up

```
<Response>
  <ContAuthTxn>
    <account_status>Account setup. First transaction due to be
                    batched on 09/10/2006</account_status>
  </ContAuthTxn>
  <datacash_reference>4200200040373186</datacash_reference>
  <information>Continuous Authority account set up</information>
  <merchantreference>382684637615741</merchantreference>
  <mode>…</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Response for a card type which is not supported by the service

```
<Response>
  <datacash_reference>4200200040609090</datacash_reference>
  <information>Continuous authority is not supported on
               this card</information>
  <merchantreference>383914408564815</merchantreference>
  <mode>…</mode>
  <reason>CA Not Supported</reason>
  <status>92</status>
  <time>…</time>
</Response>
```

Example XML Response where invalid information has been provided

```
<Response>
  <datacash_reference>4600200040609088</datacash_reference>
  <information>The date of the first payment must be from
               tomorrow onwards</information>
  <merchantreference>12345612</merchantreference>
  <mode>…</mode>
  <reason>Invalid Date</reason>
  <status>94</status>
  <time>…</time>
</Response>
```

C.4.5. Notifications

Whenever a payment is taken from an account, an email will be generated containing the results of the payment(s). The email will contain up to five attachments. Each attachment will take the form of a CSV file and be named one of:

- declined.csv – for declined payments
- error.csv – payments which generated an error message
- expiring.csv – payments on cards which will expire before the next payment date
- referred.csv – for referred payments
- authorized.csv – for successful payments

These notifications can also be downloaded from Reporting

Format for CSV files		
Name		**description**
1	Account Reference	The DataCash reference of the account from which the payment was requested.
2	Merchant Reference	Your reference for the account. As supplied when you setup the account.
3	DataCash Reference	The DataCash reference of the payment that failed. This is not supplied for `referred.csv`

Example CSV file

```
Account Reference,Merchant Reference,DataCash Reference
37707655,acc000045452,4300200040945208
37699077,SMITH0001245432,4100200040945209
37699081,Green458934,4900200040945210
37701359,qwerty12345,4700200040945211
37709097,rtuier49rsw,4500200040945212
37709891,egr3t434,4300200040945213
```

D. Fraud Prevention

D.1. AVSCV2

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/fraud_prevention/avscv2/index.shtml

This section covers the elements to be populated with the cv2 and address details. It also covers the sending of standard and extended policy information within the transaction.

This service is utilised by sending a normal Credit and Debit Card Service Request with additional information. This section of documentation assumes the Credit and Debit Card Service has already been integrated and the reader is familiar with it. The Credit and Debit Card Service is described in section B.1.

It may also be used in conjunction with the Pre-Registered Card Service, which is described in section C.1.

D.1.1. Schema Elements for Request

In this section the fields that can be submitted when using the AVSCV2 service will be presented, along with example XML for those fields.

Each field will be labelled with the following key:

- O - Optional
- R - Required, field must be presented
- C - Checked, the optional field will be checked if presented
- M - Mandatory if available, if the information is available, it should be presented

The following additional schema elements can be presented when performing an AVSCV2 check:

- `Request`
  - `Transaction`
    - `CardTxn`
      - `Card`
        - `Cv2Avs` – see section D.1.1.1
          - `ExtendedPolicy` – see section D.1.1.2

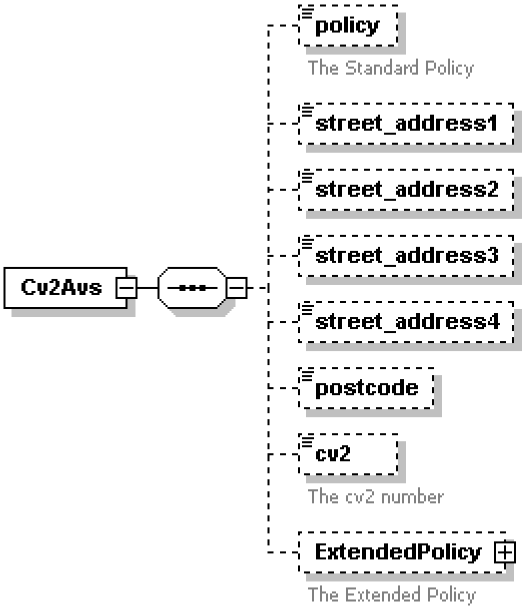
D.1.1.1. Cv2Avs

This parent element should be populated with the information that you require to be checked. When using the either Standard or Extended Policy, this information is also passed here – only one of these elements should be presented. If you wish to use the default policy, neither element should be presented.

Please remember that if you are only presenting a particular subset of the AVSCV2 data to be checked, you must choose a policy that does not require the all fields to be checked – this would otherwise result in all transactions being rejected.

None of the child elements of Cv2Avs are mandatory. If the `cv2` element is presented, the length of its contents will be checked to ensure the number is the correct length for the card type. American Express cards have four digits and all other card types have three. An empty cv2 number will cause the rejection of the transaction.

Element Name: `Cv2Avs`
Position `Request.Transaction.CardTxn.Card.Cv2Avs`



Elements of Cv2Avs			
Element Name	**description**	**values / limitations**	**Required**
`street_address1`	The statement address (excluding postcode) of the customer.	Though only the numeric data can be checked, the full details can be submitted if required. Any non-numeric data will be stripped out and the numeric data will be concatenated prior to checking.	C
`street_address2`			
`street_address3`			
`street_address4`			
`postcode`	The statement postcode	A maximum of 9 alphanumeric characters.	C
`cv2`	The cv2 number from the card	Must be 4 digits for Amex, or 3 digits for all other card types.	C
`policy`	The Standard Policy required for the transaction.	1, 2, 3, 5, 6, 7. Please refer to the website for definitions	O[1]
`ExtendedPolicy`	For use with Extended Policy only. See section D.1.1.2		

[1] Either of the `policy` or `ExtendedPolicy` elements can be presented. To use the default policy configured on the account, neither element should be presented. If a transaction contains both standard and extended policy information, it will be rejected.

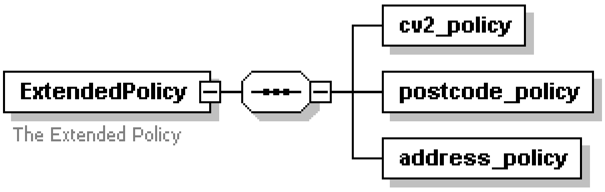
Example Cv2Avs elements for default, standard and extended policy transactions respectively. In each example, the numeric data for the address and postcode is identical and so would return the same result from the bank.

```
<Cv2Avs>
  <street_address1>1 High Street</street_address1>
  <street_address2>This Town</street_address2>
  <street_address3>Somewhere</street_address3>
  <street_address4>United Kingdom</street_address4>
  <postcode>S01 2CD</postcode>
  <cv2>123</cv2>
</Cv2Avs>

<Cv2Avs>
  <street_address1>1 High Street,
     This Town, Somewhere,
     United Kingdom</street_address1>
  <postcode>S01 2CD</postcode>
  <cv2>123</cv2>
  <policy>3</policy>
</Cv2Avs>

<Cv2Avs>
  <street_address1>1</street_address1>
  <postcode>012</postcode>
  <cv2>123</cv2>
  <ExtendedPolicy>…</ExtendedPolicy>
</Cv2Avs>
```

D.1.1.2. ExtendedPolicy

When using the extended policy, each element and attribute must be presented

Element Name: ExtendedPolicy
Position Request.Transaction.CardTxn.Card.Cv2Avs.ExtendedPolicy



Elements of ExtendedPolicy			
Element Name	**description**	**values / limitations**	**Required**
cv2_policy	Each of these elements has five attributes – see below.		R
postcode_policy			R
address_policy			R

Attributes for the child elements of ExtendedPolicy				
Attribute Name	**Element Name**	**Description**	**values / limitations**	**Required**
notprovided	postcode_policy cv2_policy address_policy	Specifies whether to accept or reject each of the five possible bank responses for the three individual elements	accept reject	R
notchecked				R
matched				R
notmatched				R
partialmatch				R

Example Extended policy element.

```
<ExtendedPolicy>
  <cv2_policy notprovided="reject"
     notchecked="accept"
     matched="accept"
     notmatched="reject"
     partialmatch="reject"/>
  <postcode_policy notprovided="reject"
     notchecked="accept"
     matched="accept"
     notmatched="reject"
     partialmatch="accept"/>
  <address_policy notprovided="reject"
     notchecked="accept"
     matched="accept"
     notmatched="reject"
     partialmatch="accept"/>
</ExtendedPolicy>
```

D.1.2. XML Example Requests

D.1.2.1. Default Policy

Example Default Policy Request.

```
<Request>
  <Authentication>
    <client>99000000</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123456</merchantreference>
      <amount currency="EUR">10.00</amount>
    </TxnDetails>
    <CardTxn>
      <Card>
        <pan>4444********1111</pan>
        <expirydate>03/04</expirydate>
        <Cv2Avs>
          <street_address1>Flat 7</street_address1>
          <street_address2>89 Jumble
              Street</street_address2>
          <street_address3>Mytown</street_address3>
          <postcode>AV12FR</postcode>
          <cv2>123</cv2>
        </Cv2Avs>
      </Card>
      <method>auth</method>
    </CardTxn>
  </Transaction>
</Request>
```

Example Default Policy with Pre-Registered Cards.

```
<Request>
  <Authentication>
    <client>99000000</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123456</merchantreference>
      <amount currency="EUR">10.00</amount>
    </TxnDetails>
    <CardTxn>
      <card_details type="preregistered">
          2185999900000001</card_details>
      <Card>
        <Cv2Avs>
          <street_address1>Flat 7</street_address1>
          <street_address2>89 Jumble
              Street</street_address2>
          <street_address3>Mytown</street_address3>
          <postcode>AV12FR</postcode>
          <cv2>123</cv2>
        </Cv2Avs>
      </Card>
      <method>auth</method>
    </CardTxn>
  </Transaction>
</Request>
```

D.1.2.2. Extended Policy

Example Extended Policy Request.

```
<Request>
  <Authentication>
    <client>99000000</client>
    <password>*******</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123456</merchantreference>
      <amount currency="EUR">10.00</amount>
    </TxnDetails>
    <CardTxn>
      <Card>
        <pan>4444********1111</pan>
        <expirydate>03/04</expirydate>
        <Cv2Avs>
          <street_address1>Flat 7</street_address1>
          <street_address2>89 Jumble
                Street</street_address2>
          <street_address3>Mytown</street_address3>
          <postcode>AV12FR</postcode>
          <cv2>123</cv2>
          <ExtendedPolicy>
            <cv2_policy notprovided="reject"
                notchecked="accept"
                matched="accept"
                notmatched="reject"
                partialmatch="reject"/>
            <postcode_policy notprovided="reject"
                notchecked="accept"
                matched="accept"
                notmatched="reject"
                partialmatch="accept"/>
            <address_policy notprovided="reject"
                notchecked="accept"
                matched="accept"
                notmatched="reject"
                partialmatch="accept"/>
           </ExtendedPolicy>
        </Cv2Avs>
      </Card>
      <method>auth</method>
    </CardTxn>
  </Transaction>
</Request>
```

Example Extended Policy Request when using Pre-Registered Cards.

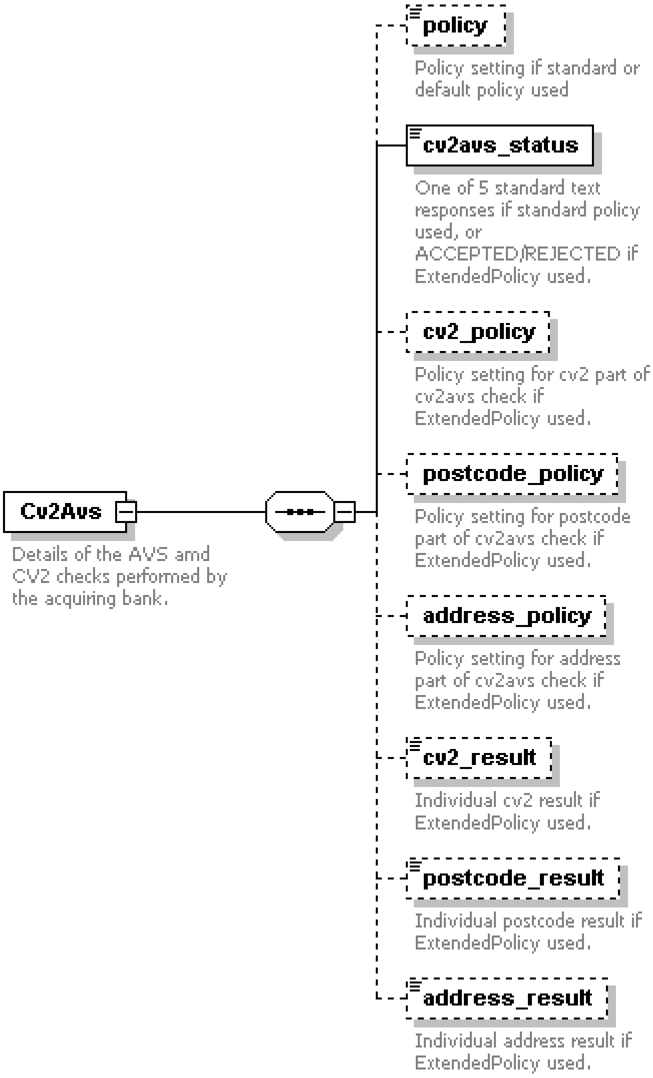
```
<Request>
  <Authentication>
    <client>99000000</client>
    <password>*******</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123456</merchantreference>
      <amount currency="EUR">10.00</amount>
    </TxnDetails>
    <CardTxn>
      <card_details type="preregistered">
         4800200052555666</card_details>
      <Card>
        <Cv2Avs>
          <street_address1>Flat 7</street_address1>
          <street_address2>89 Jumble
                Street</street_address2>
          <street_address3>Mytown</street_address3>
          <postcode>AV12FR</postcode>
          <cv2>123</cv2>
          <ExtendedPolicy>
            <cv2_policy notprovided="reject"
                notchecked="accept"
                matched="accept"
                notmatched="reject"
                partialmatch="reject"/>
            <postcode_policy notprovided="reject"
                notchecked="accept"
                matched="accept"
                notmatched="reject"
                partialmatch="accept"/>
            <address_policy notprovided="reject"
                notchecked="accept"
                matched="accept"
                notmatched="reject"
                partialmatch="accept"/>
           </ExtendedPolicy>
        </Cv2Avs>
      </Card>
      <method>auth</method>
    </CardTxn>
  </Transaction>
</Request>
```

D.1.3. Schema Elements for Response

AVSCV2 results are returned in the `Cv2Avs` element for successfully authorised transaction. If a transaction is DECLINED or REFERRED, the AVSCV2 results will not be available either in the XML response or on the Reporting pages.

D.1.3.1. Cv2Avs

Element Name:	`Cv2Avs`
Position	`Response.CardTxn`


policy
Policy setting if standard or default policy used
cv2avs_status
One of 5 standard text responses if standard policy used, or ACCEPTED/REJECTED if ExtendedPolicy used.
cv2_policy
Policy setting for cv2 part of cv2avs check if ExtendedPolicy used.
postcode_policy
Policy setting for postcode part of cv2avs check if ExtendedPolicy used.
Cv2Avs
Details of the AVS amd CV2 checks performed by the acquiring bank.
address_policy
Policy setting for address part of cv2avs check if ExtendedPolicy used.
cv2_result
Individual cv2 result if ExtendedPolicy used.
postcode_result
Individual postcode result if ExtendedPolicy used.
address_result
Individual address result if ExtendedPolicy used.

Most child elements and attributes of Cv2Avs are dependent upon whether the default/standard or extended policy checks where used.

Attributes of child elements in Cv2Avs			
Attribute Name	**Element Name**	**Description**	**values / limitations**
`reversal`	`cv2avs_status`	The result of the reversal request for transactions failing to meet the chosen policy	`0` – failed `1` - successful

D.1.3.1.1. Standard and Default Policies

Elements of Cv2Avs		
Element Name	**description**	**values / limitations**
`cv2avs_status`	The overall result of the transaction	NO DATA MATCHES ADDRESS MATCH ONLY SECURITY CODE MATCH ONLY ALL MATCH DATA NOT CHECKED
`policy`	The policy the transaction was checked against.	

Example Response elements for a transaction checked using either the default or standard policy

```
<Cv2Avs>
  <cv2avs_status reversal="1">SECURITY CODE MATCH
      ONLY</cv2avs_status>
  <policy>3</policy>
</Cv2Avs>

<Cv2Avs>
  <cv2avs_status reversal="0">ADDRESS MATCH ONLY</cv2avs_status>
  <policy>2</policy>
</Cv2Avs>

<Cv2Avs>
  <cv2avs_status>ALL MATCH</cv2avs_status>
  <policy>6</policy>
</Cv2Avs>
```

D.1.3.1.2. For Extended Polices

Elements of Cv2Avs		
Element Name	**description**	**values / limitations**
`cv2avs_status`	The overall result of the transaction	`ACCEPTED` `REJECTED`

`address_result`	The result of the check for that particular element	One of:
		`notprovided`
		`notchecked`
		`matched`
		`notmatched`
		`partialmatch`

`cv2_result`		
`postcode_result`		
`address_policy`	The policy the transaction was checked against. These have the same format as in the Request – see section D.1.1.2	
`cv2_policy`		
`postcode_policy`		

Attributes of child elements within Cv2Avs			
Attribute Name	**Element Name**	**description**	**values / limitations**

`numeric`	`address_result`
	`cv2_result`
	`postcode_result`
		The numerical result returned by the bank for each element.	`0` – not provided
			`1` – not checked
			`2` – matched
			`4` - not matched
			`8` – partial match

Example Response Elements for extended policy transactions

```
<Cv2Avs>
  <address_policy notprovided="reject"
              notchecked="accept"
              matched="accept"
              notmatched="reject"
              partialmatch="accept"/>
  <address_result numeric='0'>notprovided</address_result>
  <cv2_policy notprovided="reject"
              notchecked="accept"
              matched="accept"
              notmatched="reject"
              partialmatch="reject"/>
  <cv2_result numeric='2'>matched</cv2_result>
  <cv2avs_status reversal='1'>REJECTED</cv2avs_status>
  <postcode_policy notprovided="reject"
              notchecked="accept"
              matched="accept"
              notmatched="reject"
              partialmatch="accept"/>
  <postcode_result numeric='4'>notmatched</postcode_result>
</Cv2Avs>

<Cv2Avs>
  <address_policy notprovided="reject"
              notchecked="accept"
              matched="accept"
              notmatched="reject"
              partialmatch="accept"/>
  <address_result numeric='2'>matched</address_result>
  <cv2_policy notprovided="reject"
              notchecked="accept"
              matched="accept"
              notmatched="reject"
              partialmatch="reject"/>
  <cv2_result numeric='1'>not checked</cv2_result>
  <cv2avs_status>ACCEPTED</cv2avs_status>
  <postcode_policy notprovided="reject"
              notchecked="accept"
              matched="accept"
              notmatched="reject"
              partialmatch="accept"/>
  <postcode_result numeric='2'>matched</postcode_result>
</Cv2Avs>
```

D.1.4. XML Example Responses

D.1.4.1. Successful AVSCV2 Responses

An example response for transactions successfully passing the standard or default policy AVSCV2 check

```
<Response>
  <CardTxn>
    <authcode>3956</authcode>
    <card_scheme>VISA</card_scheme>
    <Cv2Avs>
       <cv2avs_status>SECURITY CODE MATCH ONLY</cv2avs_status>
       <policy>2</policy>
    </Cv2Avs>
  </CardTxn>
  <datacash_reference>4000000098765888</datacash_reference>
  <merchantref>…</merchantref>
  <mode>…</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>…2</time>
</Response>
```

An example response for transactions successfully passing the extended policy AVSCV2 check

```
<Response>
  <CardTxn>
    <authcode>4382</authcode>
    <card_scheme>VISA</card_scheme>
    <Cv2Avs>
      <address_policy notprovided="reject"
         notchecked="accept" matched="accept"
         notmatched="reject" partialmatch="accept"/>
      <address_result numeric='2'>matched</address_result>
      <cv2_policy notprovided="reject"
         notchecked="accept" matched="accept"
         notmatched="reject" partialmatch="reject"/>
      <cv2_result numeric='1'>not checked</cv2_result>
      <cv2avs_status>ACCEPTED</cv2avs_status>
      <postcode_policy notprovided="reject"
         notchecked="accept" matched="accept"
         notmatched="reject" partialmatch="accept"/>
      <postcode_result numeric='2'>matched</postcode_result>
    </Cv2Avs>
  </CardTxn>
  <datacash_reference>4000000098765433</datacash_reference>
  <merchantref>…</merchantref>
  <mode>…</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

D.1.4.2. AVSCV2 Declined Responses

An example response for transactions failing to match the standard or default policy chosen

```
<Response>
  <CardTxn>
    <authcode>3956</authcode>
    <card_scheme>VISA</card_scheme>
    <Cv2Avs>
      <cv2avs_status reversal=”1”>ADDRESS MATCH ONLY
         </cv2avs_status>
      <policy>3</policy>
    </Cv2Avs>
  </CardTxn>
  <datacash_reference>4000000098765889</datacash_reference>
  <merchantref>123998</merchantref>
  <mode>LIVE</mode>
  <reason>CV2AVS DECLINED</reason>
  <status>7</status>
  <time>1234567812</time>
</Response>
```

An example response for transactions failing to match the extended policy chosen

```
<Response>
  <CardTxn>
    <authcode>2893</authcode>
    <card_scheme>VISA</card_scheme>
    <country>United Kingdom</country>
    <Cv2Avs>
      <address_policy notprovided="reject"
         notchecked="accept" matched="accept"
         notmatched="reject" partialmatch="accept"/>
      <address_result numeric='1'>notchecked</address_result>
      <cv2_policy notprovided="reject"
         notchecked="accept" matched="accept"
         notmatched="reject" partialmatch="reject"/>
      <cv2_result numeric='2'>matched</cv2_result>
      <cv2avs_status reversal='1'>REJECTED</cv2avs_status>
      <postcode_policy notprovided="reject"
         notchecked="accept" matched="accept"
         notmatched="reject" partialmatch="accept"/>
      <postcode_result numeric='4'>notmatched</postcode_result>
    </Cv2Avs>
  </CardTxn>
  <datacash_reference>4000000098765432</datacash_reference>
  <merchantref>123456</merchantref>
  <mode>LIVE</mode>
  <reason>CV2AVS DECLINED</reason>
  <status>7</status>
  <time>1234567890</time>
</Response>
```

D.1.4.3. Declined and Referred Transactions

If a transaction is not authorised by the bank, no AVSCV2 information will be available. The transaction Response will therefore be the same as a declined or referred transaction with no AVSCV2 checking. Please refer to section B.1.4 for examples

D.2. URU®

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/fraud_prevention/URU/overview.shtml

This Service allows an individual's identity and personal information to be authenticated against a number of databases, and the results of previous authentications to be retrieved.

Some data elements used in this service use the Universal Unique ID format (UUID). This format is `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` and unless otherwise indicated is hexadecimal.

D.2.1. Schema Elements for Request

In this section the required fields for URU® requests will be presented, along with example XML for those fields.
As the two transaction types do not use the same fields, each field will be labelled with the following key:

- O – Optional
- R – Required
- X – Excluded, presenting this field for the transaction type will cause the transaction to fail.

It should be noted that these indicate the status of the field within the DataCash Payment Gateway only. If you are integrating this service, we strongly advise that you discuss with URU® the fields that must be used in order to gain optimum results from the service.

Authentications

The information required to process an identity authentication - `authenticate` - is passed in several distinct places in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `TxnDetails` - contains the merchantreference, section D.2.1.1
    - `URUTxn` - contains the data relating to the individual being authenticated, along with the method authenticate, section D.2.1.2
      - `Basic` – basic details about the customer, section D.2.1.3
      - `UKData` – customer details for UK customers, sectionD.2.1.4:
      - `Address1`, `Address2`, `Address3` & `Address4` – details for up to four addresses, section D.2.1.5
      - `Passport` – details for UK passports, section D.2.1.6
      - `Electric` – details of the electricity bill, section D.2.1.7
      - `Telephone` – details about the telephone number, section D.2.1.8
      - `Driver` – details of the driving licence, section D.2.1.9
      - `Birth` – details about the place of birth of the customer and mothers maiden name, section D.2.1.10
      - `USData` - customer details for US customers. Includes social security number, driving licence and telephone number. Section D.2.1.11
      - `USAddress1`, `USAddress2`, `USAddress3` & `USAddress4` – details for up to four addresses, section D.2.1.12
      - `Employment` – employment history, section D.2.1.13
      - `CreditDebitCard` – credit and debit card details, section D.2.1.14
      - `InternationalPassport` – details of the machine-readable international passport, section D.2.1.15
      - `Authenticate` – details of the URU account to be used for the transaction, section D.2.1.16

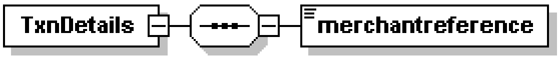
Log Requests

The log of a previous authentication may be retrieved using this method, `get_log_by_authentication_id`

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `TxnDetails` - contains the merchantreference, section D.2.1.1.
    - `URUTxn` - contains the query reference for the authentication being retrieved, along with the method `get_log_by_authentication_id`, section D.2.1.2

D.2.1.1. TxnDetails

Element Name: `TxnDetails`
Position(s) `Request.Transaction`



Elements of TxnDetails				
Element Name	**description**	**values / limitations**	**authenticate**	**log request**
`merchantreference`	A unique reference number for each transaction	Between six and thirty alphanumeric characters	R	R

Example XML for TxnDetails elements

```
<TxnDetails>
      <merchantreference>123402ABCDE12</merchantreference>
</TxnDetails>
```

D.2.1.2. URUTxn

This is the base element for submission of URU specific information. As well as holding information about an individual for authentication, it contains the transaction type and, in the case of log requests, the original authentication ID.

In the case of authentication transactions, only the method element is mandatory. However, at least one of the other user data sub-elements must be present.

Element Name:	URUTxn
Position(s)	Request.Transaction

Elements of URUTxn				
Element Name	**description**	**values / limitations**	**authenticate**	**log request**
method	Identifies the transaction type as an authentication, or a log request	authenticate get_log_by_authentication_id	R	R
Basic	Basic information about an individual's name and date of birth	See section D.2.1.3	O	X
Employment	Information about an individual's employment	See section D.2.1.13	O	X
CreditDebitCard	Information about an individual's Payment Card	See section D.2.1.14	O	X
InternationalPassport	Information about an individual's international machine-readable passport	See section D.2.1.15	O	X
guid	The query reference of the original transaction	UUID	X	R
UKData	Information for UK residents	See section D.2.1.4	O	X
USData	Information for US Residents	See section D.2.1.11	O	X
Authenticate	Information about the URU account which is to be used for the transaction	See section D.2.1.16	O	X

Example XML Element URUTxn for an authentication

```
<URUTxn>
     <method>authenticate</method>
     <Basic>...</Basic>
     <UKData>...</UKData>
     <Employment>...</Employment>
     <CreditDebitCard>...</CreditDebitCard>
     <InternationalPassport>...</InternationalPassport>
</URUTxn>
```

Example XML Element URUTxn for a log request

```
<URUTxn>
     <method>get_log_by_authentication_id</method>
     <guid>01234567-89ab-dcef-0123-456789abcdef</guid>
</URUTxn>
```

D.2.1.3. Basic

This element holds information about an individual's name and date of birth. Unless otherwise specified, all string fields accept up to 256 characters. All sub-fields are optional.

Element Name:	Basic
Position(s)	Request.Transaction.URUTxn

Elements of Basic		
Element Name	**description**	**values / limitations**
dob_day	The day of the month of the individual's birthday	int 1-31
dob_month	The month of the year of the individual's birthday	int 1-12
dob_year	The year of the individual's birth	int 0-9999
forename	The individual's forename	string
gender	The individual's gender	Male Female Unknown Unspecified
middle_initial	The individual's middle initial(s)	string
surname	The individual's surname	string
title	The title of the individual (eg Mr, Miss)	string

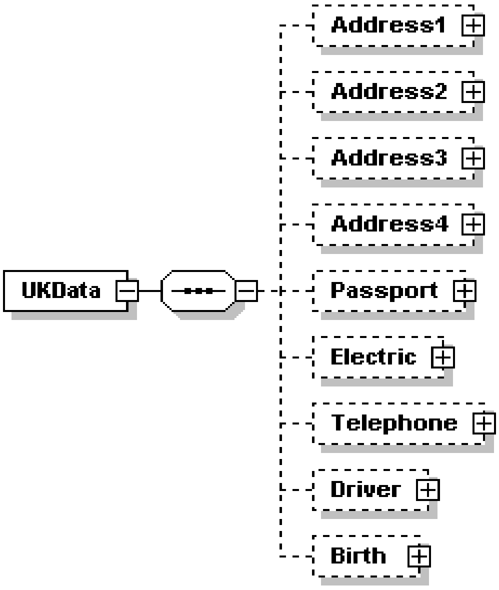
Example XML Element Basic

```
<Basic>
  <dob_day>14</dob_day>
  <dob_month>12</dob_month>
  <dob_year>1972</dob_year>
  <forename>Joe</forename>
  <gender>Male</gender>
  <middle_initial>F</middle_initial>
  <surname>Bloggs</surname>
  <title>Mr</title>
</Basic>
```

D.2.1.4. UKData

This contains a number of complex elements – `Address1` through to `Address4`, `Electric`, `Telephone`, `Passport`, `Driver` and `Birth`. All elements are complex (ie. contain sub elements) and optional.

Element Name:	`UKData`
Position(s)	`Request.Transaction.URUTxn`



Elements of UKData		
Element Name	**description**	**values / limitations**
`Address1`	See section D.2.1.5	
`Address2`		
`Address3`		
`Address4`		
`Passport`	See section D.2.1.6	
`Electric`	See section D.2.1.7	
`Telephone`	See section D.2.1.8	
`Driver`	See section D.2.1.9	
`Birth`	See section D.2.1.10	

Example XML Element UKData

```
<UKData>
     <Address1>...</Address1>
     <Address2>...</Address2>
     <Passport>...</Passport>
     <Electric>...</Electric>
     <Telephone exdirectory='no'>
        ...
     </Telephone>
     <Driver>...</Driver>
</UKData>
```

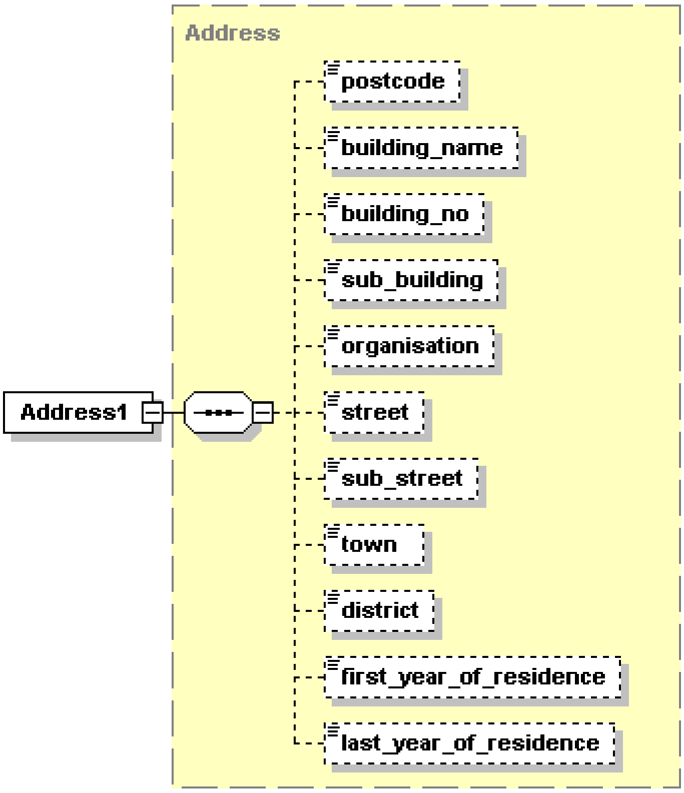
Example XML Element UKData

```
<UKData>
     <Passport>...</Passport>
     <Telephone exdirectory='yes'>...</Telephone>
     <Driver>...</Driver>
</UKData>
```

D.2.1.5. Address#*n*

Up to four addresses may be specified for addresses in the UK, labelled Address1, Address2, Address3 and Address4. Each takes the same format. Unless otherwise specified, all string fields accept up to 256 characters. All sub-fields are optional.

Element Name:	Address#n
Position(s)	Request.Transaction.URUTxn.UKData



Elements of Address#n		
Element Name	**description**	**values / limitations**
postcode	The postcode of the address	string
building_name	The building name of the address	string
building_no	The building number of the address	string
sub_building	The sub-building name/number of the address	string
organisation	The organisation name of the address	string
street	The street name of the address	string
sub_street	The sub-street name of the address	string
town	The town name of the address	string
district	The district name of the address	string
first_year_of_residence	The first year the individual was at this address	int 0-9999

`last_year_of_residence`	The last year theindividual was at this address	int 0-9999

Example XML Element Address1

```
<Address1>
  <postcode>AB1 2CD</postcode>
  <building_name>Duncodin</building_name>
  <building_no>1</building_no>
  <sub_building>3F2</sub_building>
  <organisation>Company Ltd</organisation>
  <street>Main Street</street>
  <sub_street>East End</sub_street>
  <town>Littleton</town>
  <district>Central</district>
  <first_year_of_residence>2003</first_year_of_residence>
  <last_year_of_residence>2004</last_year_of_residence>
</Address1>
```

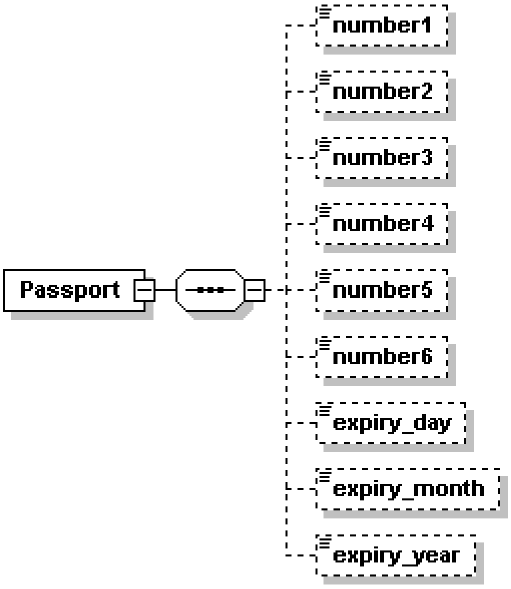
Example XML Element Address2

```
<Address2>
  <postcode>AB1 2CD</postcode>
  <building_no>8</building_no>
  <street>High Street</street>
  <town>My Town</town>
</Address2>
```

D.2.1.6. Passport

This block holds information about an individual's UK passport. International passports should use the `InternationalPassport` element, section D.2.1.15. The passport number is split into six parts, and submitted as `number1` to `number6`. All sub-fields are optional.

Element Name:	`Passport`
Position(s)	`Request.Transaction.URUTxn`



Elements of Passport		
Element Name	**description**	**values / limitations**
`number1`	The first 10 digits of the passport number (9 digit number plus checksum)	string, 10 characters
`number2`	The next 3 digits of the passport number (Issuing State code)	string, 3 characters
`number3`	The next 7 digits of the passport number, (date of birth plus checksum)	string, 7 characters
`number4`	The next 1 character of the passport number (gender)	string, 1 character
`number5`	The next 7 characters of the passport number (passport expiry date plus checksum)	string, 7 characters
`number6`	The final 2 digits of the passport number (checksum digits)	string, 2 characters
`expiry_day`	The day of the month of the passport expiry	int, 1-31
`expiry_month`	The month of the year of the passport expiry	int, 1-12
`expiry_year`	The year of the passport expiry	int, 0-9999

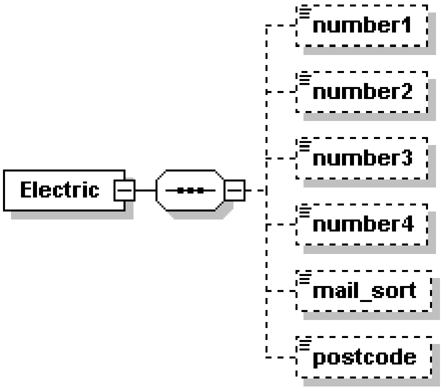
Example XML Passport elements

```
<Passport>
   <number1>1234567890</number1>
   <number2>GBR</number2>
   <number3>6201014</number3>
   <number4>M</number4>
   <number5>1231237</number5>
   <number6>12</number6>
   <expiry_day>12</expiry_day>
   <expiry_month>5</expiry_month>
   <expiry_year>2010</expiry_year>
</Passport>

<Passport>
   <number3>7506083</number3>
   <number4>F</number4>
   <number6>12</number6>
   <expiry_day>23</expiry_day>
   <expiry_month>9</expiry_month>
   <expiry_year>2008</expiry_year>
</Passport>
```

D.2.1.7. Electric

This block holds information about an individual's electricity bill. The bill number is split into four parts and submitted as `number1` to `number4`. Unless otherwise specified strings accept a maximum 256 characters. All sub-fields are optional.

Element Name:	`Electric`
Position(s)	`Request.Transaction.URUTxn`



Elements of Electric		
Element Name	**description**	**values / limitations**
number1	The first 2 digits of the bill number	string, 2 characters
number2	The next 4 digits of the bill number	string, 4 characters
number3	The next 4 digits of the bill number	string, 4 characters
number4	The final 3 digits of the bill number	string, 3 character
mail_sort	The mail sort code of the bill address	string
postcode	The postcode the bill is sent to	string

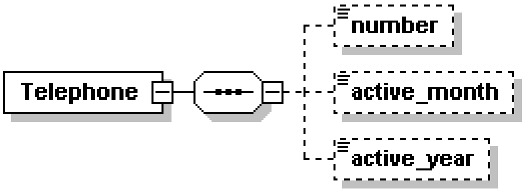
Example XML Electric elements

```
<Electric>
   <number1>12</number1>
   <number2>1234</number2>
   <number3>1234</number3>
   <number4>123</number4>
   <mail_sort>AB123</mail_sort>
   <postcode>AB12 3CD</postcode>
</Electric>

<Electric>
   <number1>56</number1>
   <number2>8318</number2>
   <number3>1234</number3>
</Electric>
```

D.2.1.8. Telephone

This block holds information about an individual's telephone number. Accepts an optional attribute indicating whether or not the number is ex-directory. Unless otherwise specified strings accept a maximum 256 characters. All sub-fields are optional.

Element Name:	`Telephone`
Position(s)	`Request.Transaction.URUTxn`



Attributes of Telephone		
Attribute Name	**description**	**values / limitations**
`exdirectory`	Whether or not the telephone is ex-directory	`yes` `no`

Elements of Telephone		
Element Name	**description**	**values / limitations**
`number`	The telephone number, including STD code	string. May be specified with or without space between STD code and number
`active_month`	The month the telephone number became active	int, 1-12
`active_year`	The year the telephone number became active	int, 0-9999

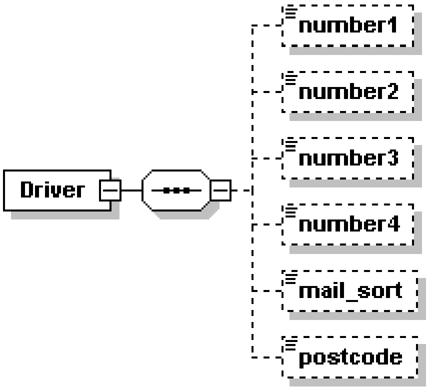
Example XML Telephone elements

```
<Telephone ex-directory=”yes”>
   <number>0123 4567890</number>
   <active_month>12</active_month>
   <active_year>2000</active_year>
</Telephone>

<Telephone ex-directory=”no”>
   <number>0123 4567892</number>
</Telephone>
```

D.2.1.9. Driver

This block holds information about an individual's driving licence. The licence number is split into four parts and submitted as `number1` to `number4`. Unless otherwise specified strings accept a maximum 256 characters. All sub-fields are optional.

Element Name:	`Driver`
Position(s)	`Request.Transaction.URUTxn`



Elements of Driver		
Element Name	**description**	**values / limitations**
`number1`	The first 5 digits of the licence number	string, 5 characters
`number2`	The next 6 digits of the licence number	string, 6 characters
`number3`	The next 3 digits of the licence number	string, 3 characters
`number4`	The last 2 digits of the licence number	string, 2 characters
`mail_sort`	The mail sort code of the licence address	string
`postcode`	The postcode of the licence address	string

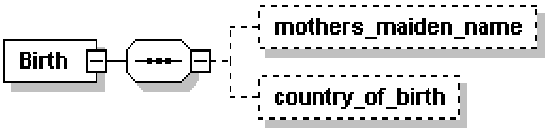
Example XML Driver elements

```
<Driver>
   <number1>HENDE</number1>
   <number2>123456</number2>
   <number3>123</number3>
   <number4>12</number4>
   <mail_sort>AB123</mail_sort>
   <postcode>AB1 2CD</postcode>
</Driver>

<Driver>
   <number2>589231</number2>
   <number4>48</number4>
</Driver>
```

D.2.1.10. Birth

This block holds information about an individual's birth details. Both elements are optional

Element Name:	Birth
Position(s)	Request.Transaction.URUTxn.UKData



Elements of Birth		
Element Name	**description**	**values / limitations**
mothers_maiden_name	Mother's maiden name	String, max 256 characters
country_of_birth	Country of birth	ENGLANDWALES OTHER UNSPECIFIED

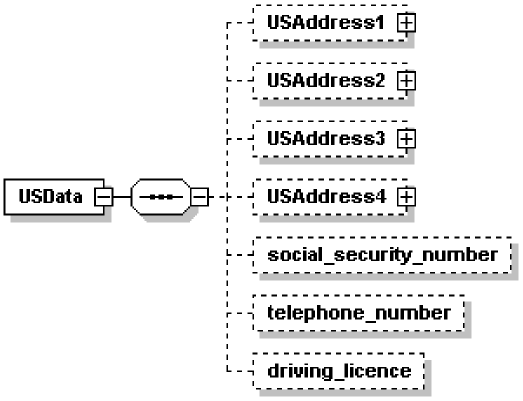
Example XML Birth elements

```
<Birth>
   <mothers_maiden_name>Smith</mothers_maiden_name>
   <country_of_birth>UNSPECIFIED</country_of_birth>
</Birth>
```

D.2.1.11. USData

This contains various details for customers based in the USA.

Element Name:	USData
Position(s)	Request.Transaction.URUTxn



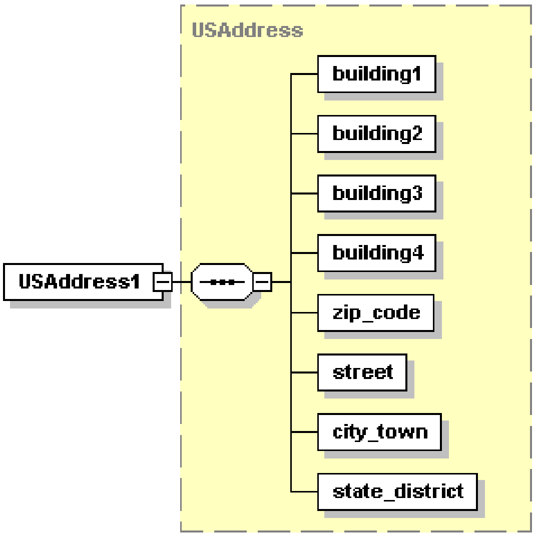
Elements of USData		
Element Name	**description**	**values / limitations**
USAddress1	See D.2.1.12	
USAddress2		
USAddress3		
USAddress4		
social_security_number	Social security number	string
telephone_number	Telephone number	string
driving_licence	The driver’s licence number	string

```
Example XML USData elements

<USData>
  <USAddress1>...</USAddress1>
  <USAddress2>...</USAddress2>
  <USAddress3>...</USAddress3>
  <USAddress4>...</USAddress4>
  <social_security_number>string</social_security_number>
  <telephone_number>string</telephone_number>
  <driving_licence>string</driving_licence>
</USData>
```

D.2.1.12. USAddress#n

Up to four addresses may be specified for addresses in the USA, labelled `USAddress1, USAddress2, USAddress3 and USAddress4.` Each takes the same format. Unless otherwise specified, all string fields accept up to 256 characters. All sub-fields are optional.

Element Name:	`USAddress#n`
Position(s)	`Request.Transaction.URUTxn.USData`



Elements of Address#n		
Element Name	**description**	**values / limitations**
`building1`	Address line 1	string
`building2`	Address line 2	string
`building3`	Address line 3	string
`builidng4`	Address line 4	string
`zip_code`	Postal code of the builing	string
`street`	The street name of the building	string
`city_town`	Town or City	string
`state_district`	The state or district	string

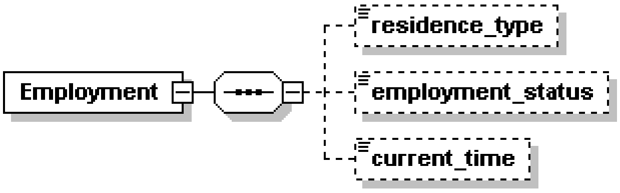
Example XML USAddress#n elements

```
<USAddress1>
     <building1>Apt 312</building1>
     <building2>Block B</building2>
     <building3>Manhattan Heights</building3>
     <building4>44532</building4>
     <zip_code>CHF6666</zip_code>
     <street>Fake Street</street>
     <city_town>New York</city_town>
     <state_district>New York</state_district>
</USAddress1>
```

D.2.1.13. Employment

This block holds information about an individual's employment status and residence. .

Element Name:	`Employment`
Position(s)	`Request.Transaction.URUTxn`



Elements of Employment		
Element Name	**description**	**values / limitations**
residence_type	The residence status of the individual	HomeOwnerOutright HomeOwnerMortgage Tenant LivingWithRelatives
employment_status	The employment status of the individual	FTPerm PTPerm SelfEmployed Retired Homemaker Unemployed Student ArmedForces
current_time	The length of time the individual has been in their current employment	Years0to2 Years2to5 Years5Plus

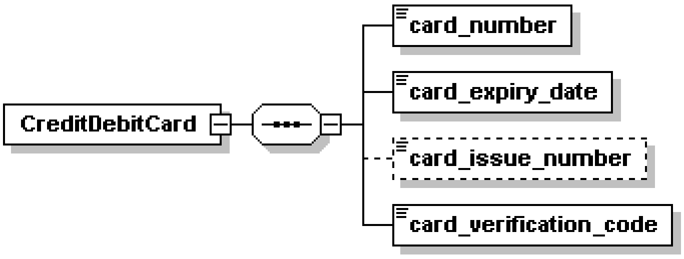
Example XML Employment elements

```
<Employment>
   <residence_type>HomeOwnerMortgage</residence_type>
   <employment_status>FTPerm</employment_status>
   <current_time>Years0to2</current_time>
</Employment>

<Employment>
   <residence_type>Tenant</residence_type>
   <employment_status>Student</employment_status>
</Employment>
```

D.2.1.14. CreditDebitCard

This block holds information about an individual's payment card. If this block is present, it must take an attribute specifying the card type. Although the block is optional as a whole, if it is present then many of its sub-fields become mandatory as indicated below.

Element Name:	CreditDebitCard
Position(s)	Request.Transaction.URUTxn



Attributes of CreditDebitCard			
Attribute Name	**description**	**values / limitations**	**Mandatory**
cardtype	The card type	VISA MASTERCARD DELTA SWITCH AMEX JCB MAESTRO DINERS SOLO ELECTRON	R

Elements of CreditDebitCard			
Element Name	**description**	**values / limitations**	**Mandatory**
card_number	The card number	string, up to 19 characters	R
card_expiry_date	The card expiry date	MMYY	R
card_issue_number	The card issue number	int, 1-9	O
card_verification_code	The CV2 number	int, 3 digits for non-Amex, 4 for Amex	R

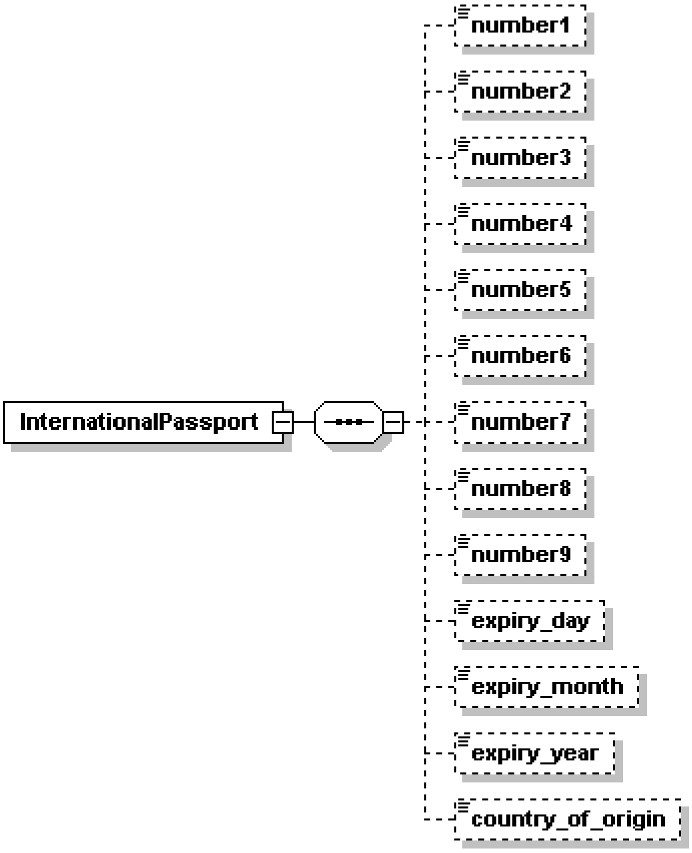
Example XML Element CreditDebitCard

```
<CreditDebitCard cardtype=”SWITCH”>
   <number>6759000000000000</number>
   <card_expiry_date>1209</card_expiry_date>
   <card_issue_number>3</card_issue_number>
   <card_verification_code>123</card_verification_code>
</CreditDebitCard>
```

D.2.1.15. InternationalPassport

This block holds information about an individual's international machine-readable passport. This element can be used for UK machine-readable passports if required, instead of the `Passport` element - section D.2.1.6.
The passport number is split into nine parts and entered as `number1` to `number9`. All sub-fields are optional.

Element Name:	`InternationalPassport`
Position(s)	`Request.Transaction.URUTxn`



Elements of InternationalPassport

Element Name	description	values / limitations
`number1`	The first 9 digits of the passport number	string, 9 characters
`number2`	The next 1 digit of the passport number, a checksum digit for `number1`	string, 1 character
`number3`	The next 3 digits of the passport number, Issuing State code	string, 3 characters
`number4`	The next 7 digits of the passport number, date of birth plus checksum	string, 7 characters
`number5`	The next 1 digit of the passport number, gender	string, 1 character
`number6`	The next 7 digits of the passport number, passport expiry date and checksum	string, 7 characters
`number7`	The next 14 digits of the passport number	string, 14 characters
`number8`	The next 1 digit of the passport number	string, 1 character
`number9`	The last 1 digit of the passport number, checksum of entire passport number	string, 1 character
`expiry_day`	The day of the month of the passport expiry	int, 1-31
`expiry_month`	The month of the year of the passport expiry	int, 1-12
`expiry_year`	The year the passport expires	int, 0-9999
`country_of_origin`	The country of origin of the passport	string, a maximum of 256 characters

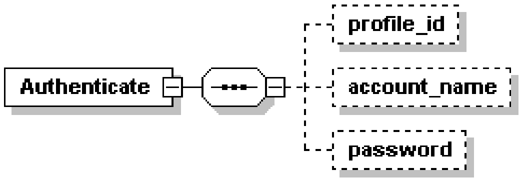
Example XML InternationalPassport elements

```
<InternationalPassport>
   <number1>123456789</number1>
   <number2>1</number2>
   <number3>123</number3>
   <number4>1234567</number4>
   <number5>1</number5>
   <number6>1234567</number6>
   <number7>12345678901234</number7>
   <number8>1</number8>
   <number9>1</number9>
   <expiry_day>21</card_expiry_date>
   <expiry_month>12</expiry_month>
   <expiry_year>2010</expiry_year>
   <country_of_origin>UK</country_of_origin>
</InternationalPassport>

<InternationalPassport>
   <number7>12345678901234</number7>
   <country_of_origin>Germany</country_of_origin>
</InternationalPassport>
```

D.2.1.16. Authenicate

This block holds information about the URU account that is to be used for this transaction. This allows the specification of a URU account other than the default account that may be set up. The block itself is optional, but if it is presented all sub fields are required.

Element Name:	`Authenticate`
Position(s)	`Request.Transaction.URUTxn`



Elements of Authenticate		
Element Name	**description**	**values / limitations**
`profile_id`	The profile identifier – the UUID of the profile	String
`account_name`	The email address that acts as the account user name	String
`password`	The URU login password	String

Example XML Authenticate element

```
<Authenticate>
  <profile_id>332c87be-aac8-42e3-9039-6b93ce1a04b8
    </profile_id>
  <account_name>no.one@nowhere.com</account_name>
  <password>password</password>
</Authenticate>
```

D.2.2. XML Example Requests

Example XML Request for an authentication, with full details

```
<Request>
  <Authentication>
    <password>password</password>
    <client>99000000</client>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>12345678</merchantreference>
    </TxnDetails>
    <URUTxn>
      <method>authenticate</method>
      <Basic>
        <forename>Jane</forename>
        <middle_initial>R</middle_initial>
        <surname>Smith</surname>
        <dob_day>18</dob_day>
        <dob_month>12</dob_month>
        <dob_year>1972</dob_year>
        <gender>Female</gender>
        <title>Ms</title>
      </Basic>
      <UKData>
        <Address1>
          <postcode>AB1 2CD</postcode>
          <building_name>Duncodin</building_name>
          <building_no>1</building_no>
          <sub_building>3F2</sub_building>
          <organisation>Company Ltd</organisation>
          <street>Main Street</street>
          <sub_street>East End</sub_street>
          <town>Littleton</town>
          <district>Central</district>
          <first_year_of_residence>2003</first_year_of_residence>
        </Address1>
        <Address2>
          <postcode>EF3 4GH</postcode>
          <building_no>1</building_no>
          <street>North Street</street>
          <town>Bigton</town>
          <first_year_of_residence>2002</first_year_of_residence>
          <last_year_of_residence>2003</last_year_of_residence>
        </Address2>
        <Address3>
          <postcode>IJ5 6KL</postcode>
          <building_no>1</building_no>
          <street>South Street</street>
          <town>Upper Bigton</town>
          <first_year_of_residence>2001</first_year_of_residence>
          <last_year_of_residence>2002</last_year_of_residence>
        </Address2>
        <Address4>
          <postcode>MN7 8OP</postcode>
          <building_no>1</building_no>
          <street>Market Street</street>
          <town>Lower Littleton</town>
          <first_year_of_residence>2000</first_year_of_residence>
          <last_year_of_residence>2001</last_year_of_residence>
        </Address2>
        <Passport>
          <number1>1234567890</number1>
          <number2>USA</number2>
          <number3>1234567</number3>
          <number4>M</number4>
          <number5>1234567</number5>
          <number6>12</number6>
          <expiry_day>31</expiry_day>
          <expiry_month>12</expiry_month>
          <expiry_year>2010</expiry_year>
        </Passport>
        <Electric>
          <number1>12</number1>
          <number2>1234</number2>
          <number3>1234</number3>
          <number4>123</number4>
          <mail_sort>12345</mail_sort>
          <postcode>AB1 2CD</postcode>
        </Electric>
        <Telephone exdirectory='no'>
          <number>0123 456 7890</number>
          <active_month>12</active_month>
          <active_year>2000</active_year>
        </Telephone>
        <Driver>
          <number1>12345</number1>
          <number2>123456</number2>
          <number3>123</number3>
          <number4>1234</number4>
          <mail_sort>6789</mail_sort>
          <postcode>AB1 2CD</postcode>
        </Driver>
        <Birth>
          <mothers_maiden_name>Smith</mothers_maiden_name>
          <country_of_birth>ENGLANDWALES</country_of_birth>
        </Birth>
      </UKData>
      <Employment>
        <residence_type>Tenant</residence_type>
        <employment_status>FTPerm</employment_status>
        <current_time>Years0to2</current_time>
      </Employment>
      <CreditDebitCard cardtype='VISA'>
        <card_number>5521480000000003</card_number>
        <card_expiry_date>1009</card_expiry_date>
        <card_issue_number>1</card_issue_number>
        <card_verification_code>321</card_verification_code>
      </CreditDebitCard>
      <InternationalPassport>
        <number1>123456789</number1>
        <number2>1</number2>
        <number3>USA</number3>
        <number4>1234567</number4>
        <number5>M</number5>
        <number6>1234567</number6>
        <number7>12345678901234</number7>
        <number8>1</number8>
        <number9>1</number9>
        <expiry_day>21</expiry_day>
        <expiry_month>2</expiry_month>
        <expiry_year>2009</expiry_year>
        <country_of_origin>Scotland</country_of_origin>
      </InternationalPassport>
    </URUTxn>
  </Transaction>
</Request>
```

Example XML Request for an authentication, with partial details

```
<Request>
  <Authentication>
    <password>password</password>
    <client>99000000</client>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123456789</merchantreference>
    </TxnDetails>
    <URUTxn>
      <method>authenticate</method>
      <Basic>
        <title>Dr</title>
        <forename>Ian</forename>
        <surname>Smith</surname>
        <dob_day>29</dob_day>
        <dob_month>11</dob_month>
        <dob_year>1968</dob_year>
      </Basic>
      <UKData>
        <Address1>
          <postcode>AB1 2CD</postcode>
          <building_no>46</building_no>
          <street>Middle Road</street>
          <town>Leeds</town>
          <first_year_of_residence>1992</first_year_of_residence>
        </Address1>
        <Telephone exdirectory='no'>
          <number>0123 456 7890</number>
          <active_month>12</active_month>
          <active_year>1992</active_year>
        </Telephone>
      </UKData>
    </URUTxn>
  </Transaction>
</Request>
```

Example XML Request for a log request

```
<Request>
  <Authentication>
    <client>99000000</client>
    <password>password</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>1357924680</merchantreference>
    </TxnDetails>
    <URUTxn>
      <method>get_log_by_authentication_id</method>
      <guid>12345678-9abc-def0-1234-56789abcdef0</guid>
    </URUTxn>
  </Transaction>
</Request>
```

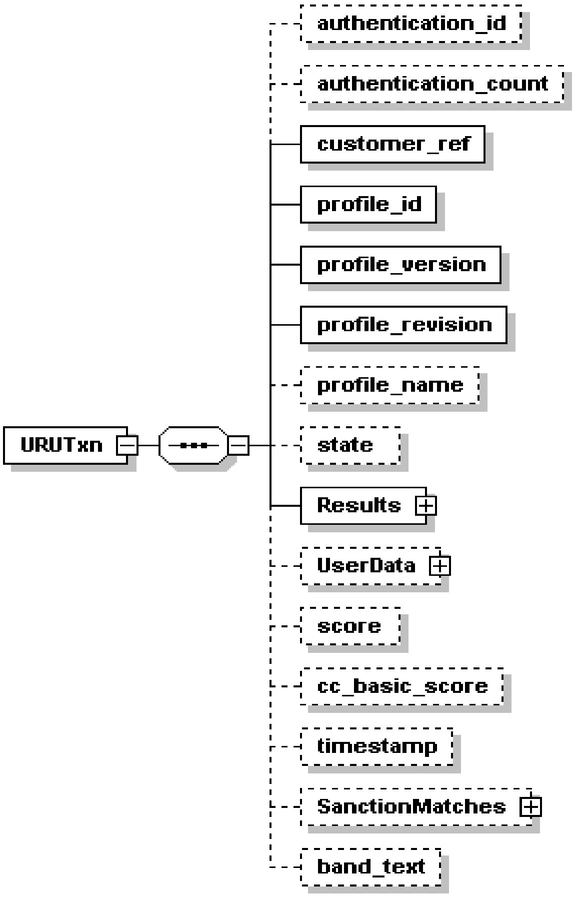
D.2.3. Schema Elements for Response

In addition to the elements covered in this section, responses for this service will contain the general response elements, covered in section A.1.2

D.2.3.1. URUTxn

This element holds all URU specific data. Elements may be relevant only to authentications, to log requests, or to both.

Element Name:	URUTxn
Position(s)	Response



Elements of URUTxn				
Element Name	**description**	**values / limitations**	**authentication**	**log request**
authentication_id	The query reference of the authentication being retrieved	UUID	N	Y
authentication_count	The number of authentications submitted on the URU account	int	Y	Y
customer_ref	The merchantreference element submitted with the authentication	string	Y	Y
profile_id	The profile id submitted with the authentication	UUID	Y	Y
provile_version	The major version number of the profile used for the authentication	int	Y	Y
profile_revision	The minor version number of the profile used for the authentication	int	Y	Y
profile_name	The name of the profile used for the authentication	string	N	Y
query_ref	An authentication reference for future log requests	UUID	Y	N
state	The state the profile was in at time of authentication	PS_TST PS_PRE PS_EFF PS_RET	N	Y
Results	The results of the authentication	See section D.2.3.2	Y	Y
UserData	The original user data supplied with the authentication		N	Y*
score	The score for this request based on values set by the URU account holder	int	Y	Y
cc_basic_score	Crecit card basic score – based on values set by the URU account holder	int	Y	Y
timestamp	Value indicating the date/time of the authentication	string	Y	Y
SanctionMatches	Match information returned from the Sanctions Items check (if processed)	See Section D.2.3.4	Y	Y
band_text	Values returned when scoring and banding are included in the URU profile	string	Y	Y

* The URU service definition provides the facility to return the original user data. However, it is not clear whether this functionality has been implemented at the URU side. If the DataCash Payment Gateway receives this data in the URU response, it will be returned in the DataCash response also.

Example XML Element URUTxn for an authentication Response

```
<URUTxn>
  <authentication_count>2</authentication_count>
  <authentication_id>332c87be-aac8-42e3-9039-6b93ce1a04b8
    </authentication_id>
  <cc_basic_score>5</cc_basic_score>
  <customer_ref>123401</customer_ref>
  <profile_id>332c87be-aac8-42e3-9039-6b93ce1a04b8</profile_id>
  <profile_revision>0</profile_revision>
  <profile_version>1</profile_version>
  <Results>
    ...
  </Results>
  <score>2</score>
  <timestamp>_SOAP_DATE_</timestamp>
</URUTxn>
```

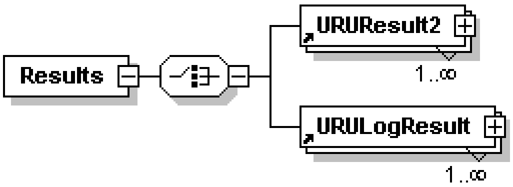
Example XML Element URUTxn for a log request Response

```
<URUTxn>
  <authentication_id>0bcb541d-8b7a-46a0-86d2-1e6a521fef5a
     </authentication_id>
  <customer_ref>1234501</customer_ref>
  <profile_id>332c87be-aac8-42e3-9039-6b93ce1a04b8
     </profile_id>
  <profile_name>Default</profile_name>
  <profile_revision>0</profile_revision>
  <profile_version>1</profile_version>
  <Results>
    ...
  </Results>
  <state>PS_EFF</state>
</URUTxn>
```

D.2.3.2. Results

This element holds a collection of result blocks. In an authentication response, these results will each be contained in `URUResult2` blocks, whilst in the response from a log request they will be in `URULogResult` blocks.

Element Name: `Results`
Position(s) `Response.URUTxn`



Elements of Results				
Element Name	**description**	**values / limitations**	**authentication**	**log request**
`URUResult2`	Information about a single aspect of one of the checks performed by the URU server during an authentication	See section D.2.3.3	Y	N
`URULogResult`	Information about a single aspect of one of the checks performed by the URU server during an authentication	See section D.2.3.3	N	Y

Example XML Element Results for a authentication transaction Response

```
<Results>
  <URULogResult2>
    ...
  </URULogResult2>
  <URULogResult2>
    ...
  </URULogResult2>
  <URULogResult2>
    ...
  </URULogResult2>
  <URULogResult2>
    ...
  </URULogResult2>
</Results>
```

Example XML Element Results for a log request Response

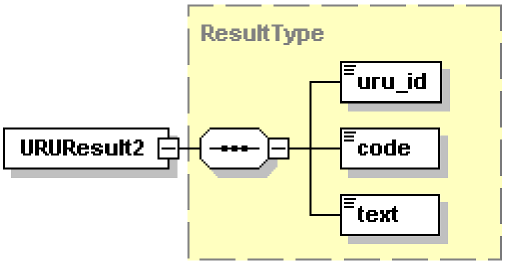
```
<Results>
  <URULogResult>
    ...
  </URULogResult>
  <URULogResult>
    ...
  </URULogResult>
</Results>
```

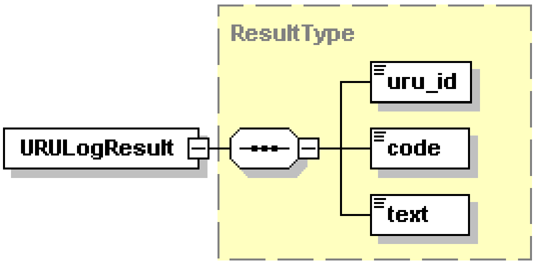
D.2.3.3. URUResult2, URULogResult

Both the `URUResult2` element, found in the response from an authentication, and the `URULogResult` element, found in the response from a log request, have the same basic format, and so are dealt with together.

Element Name:	`URUResult2`
Position(s)	`Response.URUTxn.Results`

Element Name:	`URULogResult`
Position(s)	`Response.URUTxn.Results`





Elements of URUResult2, URULogResult		
Element Name	**description**	**values / limitations**
`uru_id`	A number representing the check to which this part-result belongs	int
`code`	A numeric code describing one aspect of the check referred to by the `uru_id`	int
`text`	A textual description of the code within the context of the check referred to by the `uru_id`	string

Example XML Element URUResult2 for a log request

```
<URUResult2>
  <code>1001</code>
  <text>Address #1 is valid</text>
  <uru_id>1</uru_id>
</Results>
```

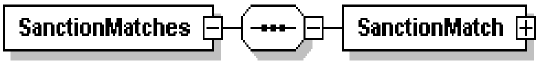
Example XML Element URULogResult for a log request

```
<URULogResult>
  <code>1001</code>
  <text>Address #1 is valid</text>
  <uru_id>1</uru_id>
</Results>
```

D.2.3.4. SanctionMatches

This element holds a collection of `SanctionMatch` blocks.

Element Name:	`SanctionMatches`
Position(s)	`Response.URUTxn`



Elements of SanctionMatches		
Element Name	**description**	**values / limitations**
`SanctionMatch`	Information about each of the sanction match checks performed	See section D.2.3.5

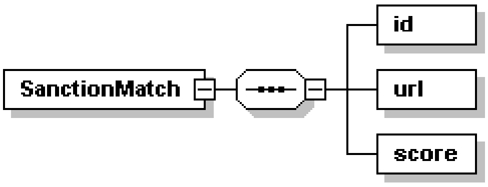
Example XML Element SanctionMatches

```
<SanctionMatches>
  <SanctionMatch>…</SanctionMatch>
  <SanctionMatch>…</SanctionMatch>
</SanctionMatches>
```

D.2.3.5. SanctionMatch

Element containing information regarding each of the sanction match checks performed. Will only appear where the URU account is configured for perform these checks

Element Name:	`SanctionMatch`
Position(s)	`Response.URUTxn.SanctionMatches`



Elements of SanctionMatch		
Element Name	**description**	**values / limitations**
id	A value representing the check to which this part-result belongs	string
`url`	The URL of the URU transaction	string
`score`	A numeric score describing the check refered to by the `id`	int

Example XML Elements SanctionMatch

```
<SanctionMatch>
  <id>Second result</id>
  <score>4</score>
  <url>https://localhost:8180/axis</url>
</SanctionMatch>

<SanctionMatch>
  <id>First result</id>
  <score>2</score>
  <url>https://localhost:8180/axis</url>
</SanctionMatch>
```

D.2.4. XML Example Responses

Example XML Response for an authentication

```
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <merchantreference>sanction_match</merchantreference>
  <mode>TEST</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>1134662619</time>
  <URUTxn>
    <authentication_count>2</authentication_count>
    <authentication_id>5C64C6F0-6D84-11DA-AC9A-9A8DBF8F5DE4
    </authentication_id>
    <cc_basic_score>5</cc_basic_score>
    <customer_ref>sanction_match</customer_ref>
    <profile_id>332c87be-aac8-42e3-9039-6b93ce1a04b8</profile_id>
    <profile_revision>0</profile_revision>
    <profile_version>1</profile_version>
    <Results>
      <URUResult2>
        <code>260</code>
        <text>Part 6 was not sufficiently supplied by user</text>
        <uru_id>3</uru_id>
      </URUResult2>
      <URUResult2>
        <code>1001</code>
        <text>Address #1 is valid</text>
        <uru_id>4</uru_id>
      </URUResult2>
      <URUResult2>
        <code>110</code>
        <text>Telephone number not specified by user</text>
        <uru_id>6</uru_id>
      </URUResult2>
      <URUResult2>
        <code>101</code>
        <text>Drivers license number was not specified in
          full</text>
        <uru_id>7</uru_id>
      </URUResult2>
    </Results>
    <SanctionMatches>
      <SanctionMatch>
        <id>Second result</id>
        <score>4</score>
        <url>https://localhost:8180/axis</url>
      </SanctionMatch>
      <SanctionMatch>
        <id>First result</id>
        <score>2</score>
        <url>https://localhost:8180/axis</url>
      </SanctionMatch>
    </SanctionMatches>
    <score>2</score>
    <timestamp>2005-12-15T16:03:41.568Z</timestamp>
  </URUTxn>
</Response>
```

Example XML Response for a log response

```
<Response>
  <datacash_reference>4600900012345673</datacash_reference>
  <merchantreference>123403</merchantreference>
  <mode>TEST</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>1134662157</time>
  <URUTxn>
    <authentication_id>26238d77-02fe-4177-8752-029a0d384e2a
      </authentication_id>
    <customer_ref>customerRef_datacash1@uru.com</customer_ref>
    <profile_id>332c87be-aac8-42e3-9039-6b93ce1a04b8</profile_id>
    <profile_name>Default</profile_name>
    <profile_revision>0</profile_revision>
    <profile_version>1</profile_version>
    <Results>
      <URULogResult>
        <code>110</code>
        <text>Telephone number not specified by user</text>
        <uru_id>6</uru_id>
      </URULogResult>
      <URULogResult>
        <code>101</code>
        <text>Drivers license number was not specified in
            full</text>
        <uru_id>7</uru_id>
      </URULogResult>
      <URULogResult>
        <code>260</code>
        <text>Part 6 was not sufficiently supplied by user</text>
        <uru_id>3</uru_id>
      </URULogResult>
      <URULogResult>
        <code>1001</code>
        <text>Address #1 is valid</text>
        <uru_id>4</uru_id>
      </URULogResult>
    </Results>
    <state>PS_EFF</state>
    <UserData>
      <Basic>
        <dob_day>10</dob_day>
        <dob_month>12</dob_month>
        <dob_year>1980</dob_year>
        <forename>Joe</forename>
        <gender>Male</gender>
        <surname>Bloggs</surname>
      </Basic>
      <CreditDebitCard>
        <card_expiry_date>0109</card_expiry_date>
        <card_issue_number>1</card_issue_number>
        <card_number>6333000000000005</card_number>
        <card_verification_code>123</card_verification_code>
        <cardtype>SWITCH</cardtype>
      </CreditDebitCard>
      <Employment>
        <current_time>Years0to2</current_time>
        <employment_status>ArmedForces</employment_status>
        <residence_type>HomeOwnerMortgage</residence_type>
      </Employment>
      <InternationalPassport>
        <number1>123456789</number1>
        <number2>1</number2>
        <number3>123</number3>
        <number4>1234567</number4>
        <number5>1</number5>
        <number6>1234567</number6>
        <number7>12345678901234</number7>
        <number8>1</number8>
        <number9>1</number9>
      </InternationalPassport>
      <UKData>
        <Address1>
          <building_no>10</building_no>
          <postcode>AB1 2CD</postcode>
          <street>Easy Street</street>
          <town>Bigton</town>
        </Address1>
        <Driver>
          <number1>12345</number1>
          <number2>123456</number2>
          <number3>123</number3>
          <number4>12</number4>
        </Driver>
        <Electric>
          <mail_sort>1234</mail_sort>
          <number1>12</number1>
          <number2>1234</number2>
          <number3>1234</number3>
          <number4>123</number4>
          <postcode>AB1 2CD</postcode>
        </Electric>
        <Passport>
          <expiry_day>1</expiry_day>
          <expiry_month>1</expiry_month>
          <expiry_year>2010</expiry_year>
          <number1>1234567890</number1>
          <number2>123</number2>
          <number3>1234567</number3>
          <number4>1</number4>
          <number5>1234567</number5>
          <number6>12</number6>
        </Passport>
        <Telephone>
          <active_month>5</active_month>
          <active_year>2002</active_year>
          <exdirectory>false</exdirectory>
        </Telephone>
      </UKData>
    </UserData>
  </URUTxn>
</Response>
```

D.3. Real Time Fraud Screening

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/fraud_prevention/screening/overview.shtml

This section of the documentation assumes the reader is familiar with the Credit and Debit Card Service, which is discussed in section B.1.

D.3.1. Schema Elements for Request

In this section the required fields for each transaction type will be presented, along with example XML for those fields. The following key will be used:

- R – Required
- MS – Market Sector merchants should consider this field as mandatory
- O – Optional for Market Sector merchants
- I – Ignored if presented

If you are using a Bespoke model – instead of the Market Sector model – please ensure that you are providing the correct data to enable each rule to be triggered.

Authorisation Requests
For authorisation requests, the *additional* information for fraud screening is passed in several different places in the schema. Most of these are located within the `Request.Transaction.TxnDetails` element:

- `TxnDetails`
  - `Order` - D.3.1.9
    - `Customer` - D.3.1.4
      - `Address` – the customer's address, section D.3.1.1
      - `Company` – for corporate orders, section D.3.1.3
        - `Address` –the companies address, section D.3.1.1
      - `CustomerHistory` - D.3.1.5
    - `BillingAddress` – the card billing address, if different from the customer's address, section D.3.1.2
  - `Shipping`, section D.3.1.12
  - `OrderDetails` - D.3.1.10
    - `LineItem` – D.3.1.8
      - `Shipping` - D.3.1.12
  - `Recipient`, section D.3.1.11
    - `Address`, section D.3.1.1
    - `LineItem` – D.3.1.8
      - `Shipping` - D.3.1.12
      -

If you are also using the AVSCV2 Service, an additional element may be submitted in the Cv2Avs element, section D.3.1.6

Over-riding Challenges
To over-ride a Fraud Challenged Response, information is passed in these places in the schema:

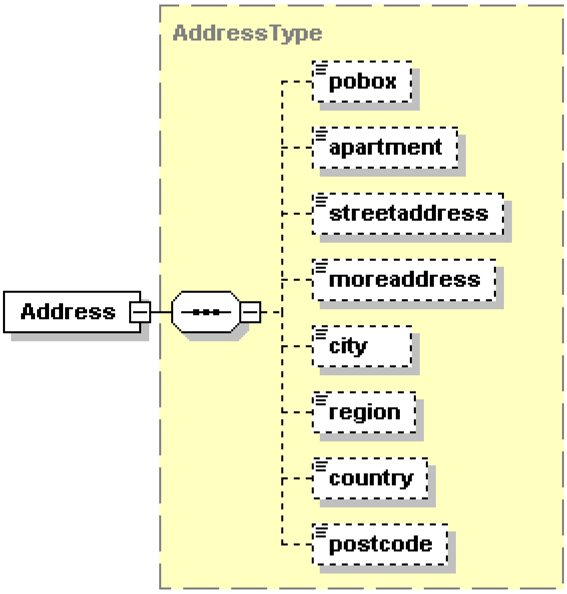
- `Request`
  - `Authentication`- section A.1.1.1
  - `Transaction`
    - `HistoricTxn`- section D.3.1.7

D.3.1.1. Address

The Address element is found in several different contexts, as the customer, customer's company and the recipients may all have different addresses. For orders being shipped to more than one address, each Address can be specified using multiple `Recipient.Address elements.` The `CustomerAddress` element will only be fraud screened if there are no `Recipient` elements present.

Element Name:	`Address`

Position(s)	`Request.Transaction.TxnDetails.Order.Customer`
	`Request.Transaction.TxnDetails.Order.Customer.Company`
	`Request.Transaction.TxnDetails.Order.Recipient`



Elements of Address			
Element Name	**description**	**values / limitations**	
`apartment`	Apartment, Suite or Flat Number	Maximum 6 characters	O
`streetaddress`	street address	Maximum 30 characters	R
`city`	City or town	Maximum 30 characters	R
`postcode`	Postcode	Maximum 30 characters	R
`region`	US State or Canadian Province code	For addresses in USA and Canada only, must be excluded for other countries	O
`country`	Country	Use the numeric country codes	R
`moreaddress`	Additional space for address if required	Maximum 30 characters	O

Example XML element for Address

```
<Address>
  <city>New York</city>
  <country>840</country>
  <postcode>12345</postcode>
  <region>NY</region>
  <streetaddress>123 Some Street </streetaddress>
</Address>
```

Example XML element for Address

```
<Address>
  <city>Dover</city>
  <country>826</country>
  <moreaddress>Kent</moreaddress>
  <postcode>XX12 3AA</postcode>
  <streetaddress>56 Kings Road</streetaddress>
</Address>
```

Example XML element for Address

```
<Address>
  <apartment>3/1</apartment>
  <city>Edinburgh</city>
  <country>826</country>
  <moreaddress>Great Michael House</moreaddress>
  <postcode>EH6 7EZ</postcode>
  <streetaddress>14 Links Place</streetaddress>
</Address>
```

D.3.1.2. BillingAddress

This element must be provided if the customer’s address is not the address to which the card is registered – for example if the customer has moved house & has not yet informed their bank.

If you are performing the AVS check, the card address needs to be entered in both the `Cv2AVS` (section D.1) and `BillingAddress` elements.

Element Name:	`BillingAddress`
Position(s)	`Request.Transaction.TxnDetails.Order`

The child elements within this parent are the same as for `Address` (section D.3.1.1)

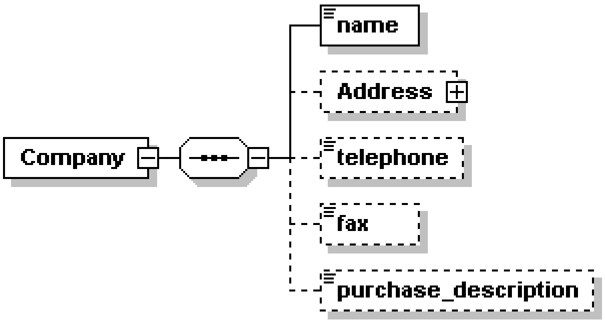
Example XML elements for BillingAddress

```
<BillingAddress>
  <apartment>6</apartment>
  <city>London</city>
  <country>826</country>
  <moreaddress>Newtown</moreaddress>
  <postcode>W1 2ZZ</postcode>
  <streetaddress>Main Street</streetaddress>
</BillingAddress>
```

D.3.1.3. Company

This element should be provided if the order is a company purchase.

Element Name: `Company`
Position(s) `Request.Transaction.TxnDetails.Order.Customer`



Elements of Company

Element Name	description	values / limitations	
`Address`	See section D.3.1.1		O
`fax`	The company fax number	1-12 numerics only	O
`name`	The company name	Maximum 30 characters	O
`purchase_description`	A description of corporate purchase	Maximum 160 characters	O
`telephone`	The company phone number	1-12 numerics only	O

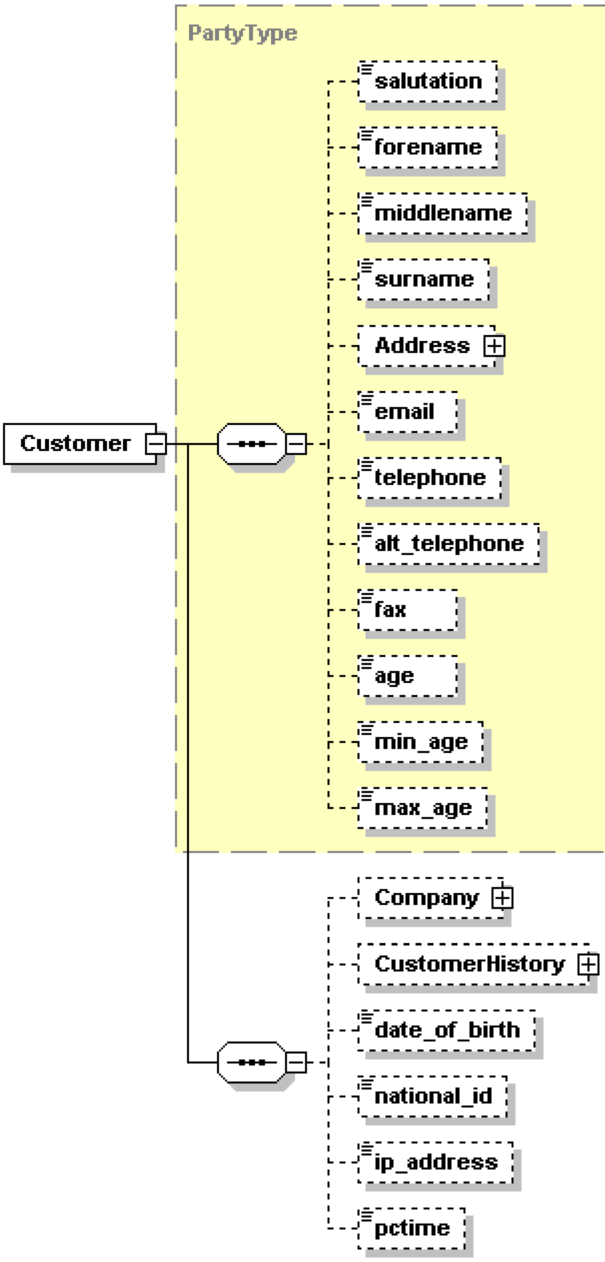
Example XML element for Company

```
<Company>
  <Address>
    …
  </Address>
  <fax>111122223001</fax>
  <purchase_description>Hardware</purchase_description>
  <name>My Company Ltd</name>
  <telephone>111122223000</telephone>
</Company>
```

D.3.1.4. Customer

This element contains various details about the customer.

Element Name:	`Customer`
Position(s)	`Request.Transaction.TxnDetails.Order`



Elements of Customer

Element Name	description	values / limitations	
`Address`	see D.3.1.1		
`age`	Age	minimum value: 0 maximum value: 150 This value will be ignored if presented for `Recipient`	O
`alt_telephone`	Home telephone number	A maximum of twelve numerics. International numbers: include Country code, exclude access code.	MS
`Company`	See section D.3.1.3		O
`CustomerHistory`	See section D.3.1.5		O
`date_of_birth`	date of birth	CCYY-MM-DD	O
`email`	email address	60 characters max	MS
`fax`	Fax number	as for `alt_telephone`	O
`forename`	First name	30 characters max	MS
`ip_address`	The IP address of the computer placing the order.	Minimum 7 characters (eg 1.2.3.4) Maximum of 17 (eg 255.255.255.255)	MS
`max_age`	Maximum age	minimum value: 0 maximum value: 150 This value will be ignored if presented for `Recipient`	O
`middlename`	Middle name or initial	30 characters max	O
`min_age`	Minimum age	minimum value: 0 maximum value: 150 This value will be ignored if presented for `Recipient`	O
`national_id`	Nationally recognised individual identification number	12 characters maximum	O
`pctime`	time on the Customer's PC	HH:MM:SS	O
`salution`	Title	5 characters max	O
`surname`	Surname / family name	30 characters max	MS
`telephone`	Work / day phone number of Customer, or main number for `Company / Recipient`	A maximum of twelve numerics. International numbers: include Country code, exclude the International dialling/access code.	O

Example XML element for Customer

```
<Customer>
  <age>43/<age>
  <email>me@hotmail.com</email>
  <Address>…</Address>
  <surname>Brown</surname>
  <forename>Mark</forename>
  <salution>Dr</salutation>
  <ip_address>1.2.3.255</ip_address>
</Customer>
```

Example XML element for Customer

```
<Customer>
  <date_of_birth>1965-01-31</<date_of_birth>
  <Address>…</Address>
  <surname>Patel</surname>
  <forename>Sanjay</forename>
  <CustomerHistory>…</CustomerHistory>
</Customer>
```

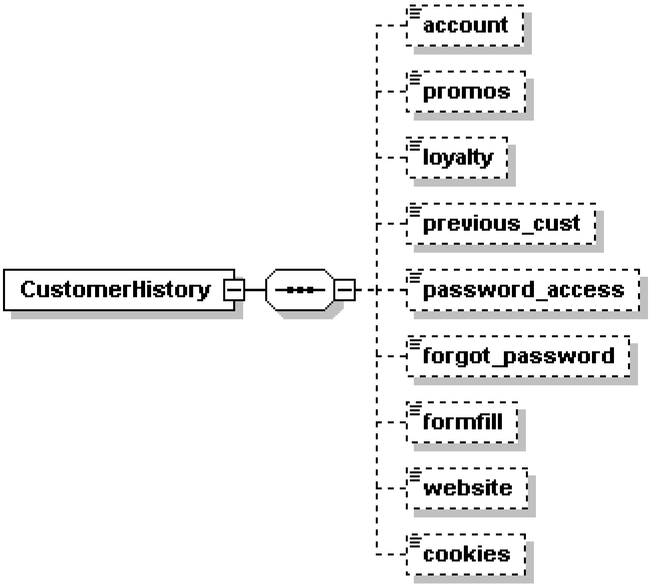
Example XML element for Customer

```
<Customer>
  <min_age>18</<min_age>
  <max_age>25</max_age>
  <forename>Mary</forename>
  <surname>O’Connor</surname>
  <telephone>…</telephone>
  <alt_telephone>…</alt_telephone>
</Customer>
```

D.3.1.5. CustomerHistory

This element enables information about the customer history to be passed. Most of the elements within this parent will only be applicable for e-Commerce transaction – for example `cookies` and `website`, though MoTo merchants may find some elements of use (eg `previous_cust`). Flagging MoTo and e-Commerce is covered in section D.3.1.13.

Element Name:	`CustomerHistory`
Position(s)	`Request.Transaction.TxnDetails.Order.Customer`



Elements of CustomerHistory			
Element Name	description	values / limitations	
`account`	username, userid etc.	max 64	O
`cookies`	cookies enabled on customers PC	XML boolean	O
`forgot_password`	did customer require the password to be reset	XML boolean	O
`formfill`	The degree to which the online purchase form filled automatically	`E` - All fields filled automatically `M` - All fields filled automatically but some fields modified manually `N` – None, all manually filled	O
`loyalty`	Has customer registered for a loyalty program	XML boolean	O
`password_access`	Did customer enter a password to access the account?	XML boolean	O
`previous_cust`	Has customer previously bought products	XML boolean	O
`promos`	Is customer registered for promotions	XML boolean	O
`website`	URL of website the purchase was made on. Particularly useful for merchants with more than one website	max 60	O

Example XML element for CustomerHistory

```
<CustomerHistory>
  <previous_cust>false</previous_cust>
</CustomerHistory>
```

Example XML element for CustomerHistory

```
<CustomerHistory>
  <account>johnsmith99</account>
  <promos>false</promos>
  <loyalty>false</loyalty>
  <previous_cust>true</previous_cust>
  <password_access>true</password_access>
  <forgot_password>false</forgot_password>
  <formfill>E</formfill>
  <website>http://www.computershop.com</website>
  <cookies>true</cookies>
</CustomerHistory>
```

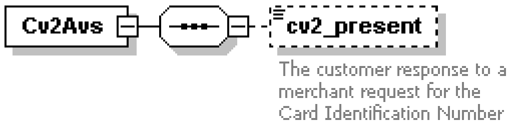
Example XML element for CustomerHistory

```
<CustomerHistory>
  <account>userXXX</account>
  <previous_cust>false</previous_cust>
  <formfill>N</formfill>
  <website>http://www.XXX.com</website>
</CustomerHistory>
```

D.3.1.6. Cv2Avs

If you are using both the ReD and AVSCV2 services, the cv2_present element must be considered to be mandatory. The other child elements of Cv2Avs are described in section D.1.1.1.

Element Name:	`Cv2Avs`
Position(s)	`Request.Transaction.CardTxn.Card`


Cv2Avs
cv2_present
The customer response to a merchant request for the Card Identification Number

Elements of Cv2Avs			
Element Name	**description**	**values / limitations**	
`cv2_present`	Gauges whether the customer had the opportunity to enter the cv2 number.	`0` - requested from the customer, but not provided `1` - entered by customer `2` - not legible on the card `8` - cv2 details were not requested on the website `9` - card had no cv2 number	R

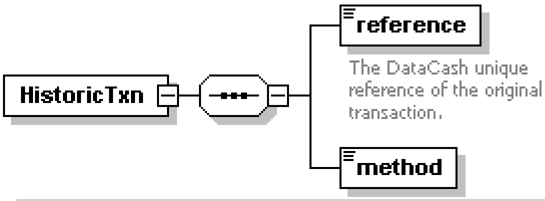
Example XML elements for Cv2Avs

```
<Cv2Avs>
  <cv2>123</cv2>
  <cv2_present>1<cv2_present>
</Cv2Avs>

<Cv2Avs>
  <street_address1>1 High Street</street_address1>
  <street_address2>This Town</street_address2>
  <street_address3>Somewhere</street_address3>
  <street_address4>United Kingdom</street_address4>
  <postcode>S01 2CD</postcode>
  <cv2_present>0<cv2_present>
</Cv2Avs>

<Cv2Avs>
  <street_address1>54 Kings Road</street_address1>
  <street_address2>This Village</street_address2>
  <street_address3>Derbyshire</street_address3>
  <postcode>DE1 1AA</postcode>
  <cv2_present>8<cv2_present>
  <ExtendedPolicy>…</ExtendedPolicy>
</Cv2Avs>
```

D.3.1.7. HistoricTxn

This element is required to over-ride Fraud Challenged Responses. It should not be presented for authorisations.

Element Name:	`HistoricTxn`
Position(s)	`Request.Transaction`



Elements of HistoricTxn			
Element Name	description	values / limitations	
`method`	The transaction method	`accept_fraud`	R
`reference`	The `datacash_reference` of the challenged transaction	Must be a successfully authorised challenge received within the previous seven days	R

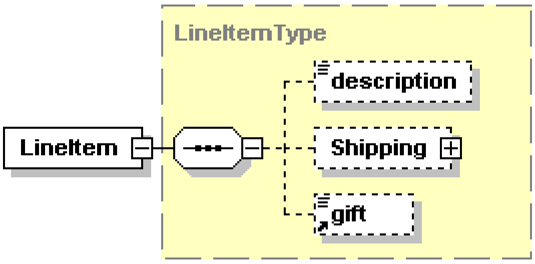
Example XML element for HistoricTxn

```
<HistoricTxn>
  <method>accept_fraud</method>
  <reference>4900200040157301<reference>
</HistoricTxn>
```

D.3.1.8. LineItem

This element enables information about each item within the order to be presented. Multiple `LineItem` elements can be presented. Each distinct product within the order should be submitted within it's own `LineItem` element.

Element Name:	`LineItem`
Position(s)	`Request.Transaction.TxnDetails.Order.OrderDetails` `Request.Transaction.TxnDetails.Order.Recipient`



Elements of LineItem			
Element Name	**description**	**values / limitations**	
`description`	a description of the product	160 max	O
`gift`	For gifts, any message attached to item. May instead be specified for the order as a whole in the `Order` element (section D.3.1.9)	160 max	O
`Shipping`	See section D.3.1.12		O

Attributes of LineItem				
Attribute Name	**Attribute of**	**description**	**values / limitations**	
`card`	`gift`	gift occasion	See Appendix I.3 for values	O
`category`	`LineItem`	category of offer	`S` - Standard `P` - Promotion	O
`manpartno`	`LineItem`	manufacturers part number	30 characters max, inc spaces	O
`manufacturer`	`LineItem`	manufacturers name	50 characters max, inc spaces	O
`product_code`	`LineItem`	your product code	18 characters max, inc spaces	O
`quantity`	`LineItem`	number of items ordered	12 digits max	O
`sku`	`LineItem`	product ID or SKU	18 characters max, inc spaces	O
`type`	`LineItem`	Product type	`P` - physical goods `D`- digital goods, eg software `C` - digital content, eg images `S` - shareware `M` - a mixture of the above.	O
`unit_price`	`LineItem`	Cost per item	If a decimal point is included, it must be followed by two digits	O
`upc`	`LineItem`	product UPC	12 characters max, inc spaces	O

wrapped	gift	Is the item/order wrapped?	XML boolean	O

Example XML for LineItem with one item

```
<LineItem category="S" quantity="1" type="P" unit_price="499.99">
  <gift card=”G” wrapped=”false”>Congratulations!</gift>
</LineItem>
```

Example XML for LineItem with eight copies of one product

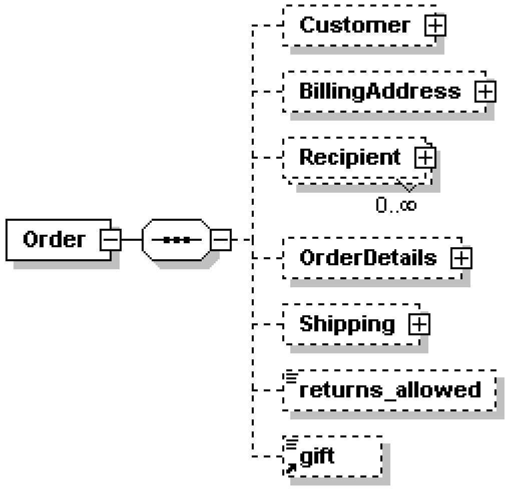
```
<LineItem category="S" quantity="8" type="P" unit_price="5.58">
</LineItem>
```

Example XML for an order with two products

```
<LineItem category="S" manpartno="part number 1J800"
    manufacturer="Dell" product_code="DELL-1J800"
    quantity="1" type="P" unit_price="900">
  <description>Desktop</description>
  <Shipping>
    …
  </Shipping>
</LineItem>
<LineItem category="P" manpartno="CD0001" manufacturer="Sony"
    product_code="SONY-CD0001" quantity="20" sku="sku 2"
    unit_price="0.50" upc="upc 2">
  <description>CD boxes</description>
  <Shipping>
    …
  </Shipping>
</LineItem>
```

D.3.1.9. Order

The majority of the extra information is contained within this element.



Elements of Order			
Element Name	**description**	**values / limitations**	
`Customer`	See section D.3.1.4		MS
`BillingAddress`	See section D.3.1.2		MS
`Recipient`	See section D.3.1.11		O
`OrderDetails`	See section D.3.1.10		O
`Shipping`	See section D.3.1.12		O
`returns_allowed`	Indicates if you enable the customer to return goods to you	XML boolean	O
`gift`	For gifts, any message attached to the entire order. May also be specified on a per item basis in the `LineItem` element (section D.3.1.8)	160 max	O

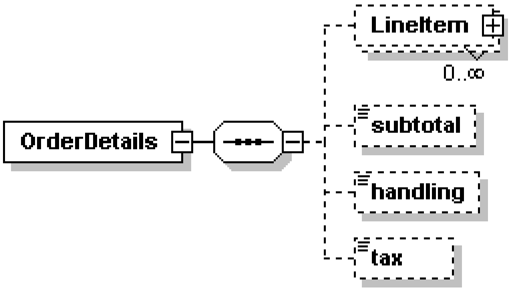
Attributes of Order				
Attribute Name	**Attribute of**	**description**	**values / limitations**	
`wrapped`	`gift`	is the item/order wrapped?	XML boolean	O
`card`	`gift`	Gift occasion	See appendix I.3	O

Example XML elements for Order

```
<Order>
  <Customer>…</Customer>
  <OrderDetails>…</OrderDetails>
  <returns_allowed>true</returns_allowed>
  <gift card=”B” wrapped=”true” />
</Order>

<Order>
  <Customer>…</Customer>
  <Recipient>…</Recipient>
  <BillingAddress></BillingAddress>
</Order>
```

D.3.1.10. OrderDetails

Element Name:	`OrderDetails`
Position(s)	`Request.Transaction.TxnDetails.Order`



Elements of OrderDetails			
Element Name	description	values / limitations	
`handling`	Shipping and handling charges	If a decimal point is included it must be followed by two digits	O
`subtotal`	Order subtotal, without tax, shipping and handling		O
`tax`	Order tax		O
`LineItem`	See section D.3.1.8		O

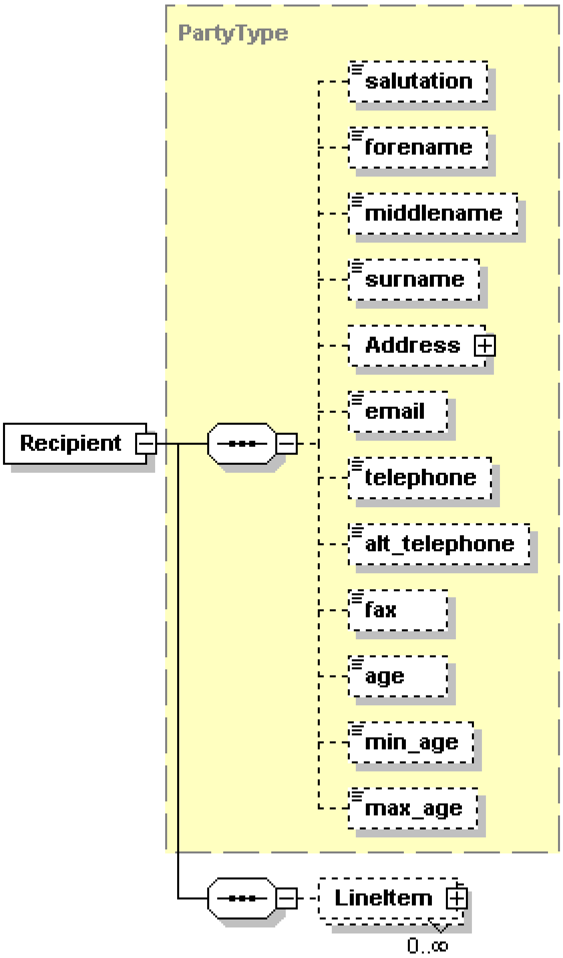
Example XML element for OrderDetails

```
<OrderDetails>
  <handling>5.00</handling>
  <subtotal>26.99</subtotal>
  <tax>5.60</tax>
  <LineItem>…</LineItem>
  <LineItem>…</LineItem>
</OrderDetails>
```

D.3.1.11. Recipient

This element enables information about the recipients of the order to be submitted. For example, if the customer is placing the order on behalf of a third party, details of that person can be passed for screening. Orders that are being shipped to several people can be presented by using one `Recipient` element for each person.

Element Name:	`Recipient`
Position(s)	`Transaction.TxnDetails.Order.Recipient`



Elements in the `PartyType` box are common to both `Customer` and `Recipient` are described in section D.3.1.4

Elements of Recipient		
Element Name	**description**	**values / limitations**
`Address`	See D.3.1.1	R
`age`	See D.3.1.4	I
`alt_telephone`		O
`email`		O
`fax`		O
`forename`		O
`max_age`		I
`middlename`		O
`min_age`		I
`salution`		O
`surname`		O
`telephone`		O
`LineItem`	See D.3.1.8	O

Example XML element for Recipient

```
<Recipient>
  <Address>…</Address>
  <forename>Julia</forename>
  <tracking_number>123</tracking_number>
  <instructions>A</instructions>
  <comments>FAO Mrs Jane White</comments>
</Recipient>
```

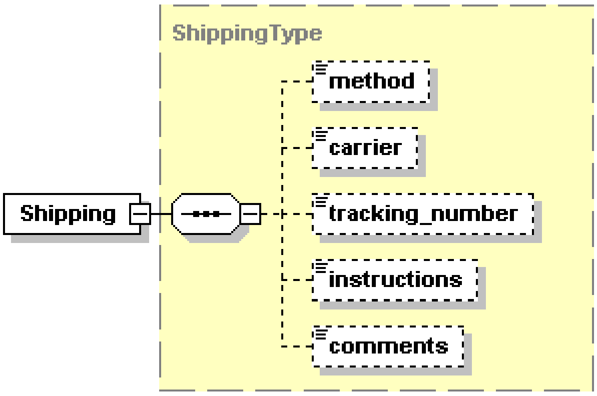
D.3.1.12. Shipping

This element enables information about the method of shipment to be entered. It may be presented in one of three places in the schema.

Element Name:	`Shipping`

Position(s)	`Request.Transaction.TxnDetails.Order.OrderDetails.LineItem`
	`Request.Transaction.TxnDetails.Order.Recipient.LineItem`
	`Request.Transaction.TxnDetails.Order`

If all items within the order are being shipped within the same parcel, the `Order.Shipping` element should be used. If the items are being shipped to the customer as they become available, the `OrderDetails.LineItem.Shipping` elements should be used for each item within the order. For orders which are being shipped to several different recipients, the `Recipient.LineItem.Shipping` element should be used for each item within the order



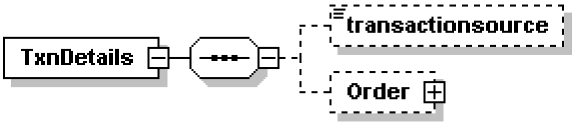
Elements of Shipping			
Element Name	description	values / limitations	
`carrier`	Shipment carrier for order/item	`F` - Fedex `P` - USPS `U` - UPS `L` – Purolator `G` - Greyhound `D` - DHL `O` - Other	O
`comments`	Additional text for shipment of order/item	160 max	O
`instructions`	When the shipment is to be made	`C` - when order is complete `A` - as items become available `D` - on a specified date	O
`method`	The delivery method	`N` - Next Day/Overnight `T` - Two Day Service `W` - Three Day Service `C` - Lowest Cost `D` - Carrier designated by Customer `I` - International `M` - Military `P` - Store delivery service `O` - Other	O
`tracking_number`	shipment tracking number for item/order	max 19	O

Example XML element for Shipping

```
<Shipping>
  <method>N</method>
  <carrier>O</carrier>
  <tracking_number>123</tracking_number>
  <instructions>A</instructions>
  <comments>FAO Mrs Jane White</comments>
</Shipping>
```

D.3.1.13. TxnDetails

Element Name:	`TxnDetails`
Position(s)	`Request.Transaction`



Elements of TxnDetails			
Element Name	**description**	**values / limitations**	
`transactionsource`	The environment in which the transaction was made. Particularly important if a single ReD account is used to screen transactions from several sources	`ecommerce` `mail-order/telephone-order`	O
`Order`	D.3.1.9		R

Example XML elements for TxnDetails

```
<TxnDetails>
  <transactionsource>ecommerce</transactionsource>
  <Order>…</Order>
</TxnDetails>

<TxnDetails>
  <transactionsource>mail-order/telephone-order</transactionsource>
  <Order>…</Order>
</TxnDetails>
```

D.3.2. Example XML Request

Example XML Request where the order is being shipped to two different recipients

```
<Request>
  <Authentication>...</Authentication>
  <Transaction>
    <TxnDetails>
      <Order>
        <Customer>
          <salutation>Mrs</salutation>
          <forename>J</forename>
          <surname>Brown</surname>
          <Address>
            <apartment>1</apartment>
            <streetaddress>10 Front Street</streetaddress>
            <city>Edinburgh</city>
            <region>Lothian</region>
            <country>826</country>
            <postcode>EH9 9ZZ</postcode>
          </Address>
          <email>j.brown@compuserve.com</email>
          <alt_telephone>44 131 999 8888</alt_telephone>
        </Customer>
        <Recipient>
          <forename>Jenny</forename>
          <surname>Black</surname>
          <Address>
            <streetaddress>26 The High Street</streetaddress>
            <city>Cardiff</city>
            <country>826</country>
          </Address>
          <LineItem manufacturer="CO Q" quantity="1"
                 unit_price="20.00">
            <description>Present</description>
            <gift card="R" wrapped="true">Happy Christmas</gift>
          </LineItem>
        </Recipient>
        <Recipient>
          <forename>Laura</forename>
          <surname>Green</surname>
          <Address>
            <streetaddress>99 Royal York Crescent</streetaddress>
            <city>London</city>
            <country>826</country>
          </Address>
          <LineItem manufacturer="ABC" quantity="2"
               unit_price="10.00">
            <description>Present</description>
            <gift card="R" wrapped="true">Happy Christmas</gift>
          </LineItem>
        </Recipient>
      </Order>
    ...
    </TxnDetails>
  <CardTxn>…</CardTxn>
  </Transaction>
</Request>
```

Example XML Request where ReD and AVSCV2 checks are being used

```
<Request>
  <Authentication>...</Authentication>
  <Transaction>
    <TxnDetails>
      <Order>
        <Customer>
          <salutation>Ms</salutation>
          <forename>Sandra</forename>
          <middlename>K</middlename>
          <surname>Littlejohn</surname>
          <Address>
            <streetaddress>37 Queen Street</streetaddress>
            <city>Oxford</city>
            <country>826</country>
            <postcode>OX20 8FP</postcode>
          </Address>
          <email>slittlejohn@ox.co.uk</email>
          <alt_telephone>0870 1234 4566</alt_telephone>
          <ip_address>48.12.3.4</ip_address>
        </Customer>
        <OrderDetails>…</OrderDetails>
      </Order>
      …
    </TxnDetails>
  <CardTxn>
    <Card>
      <Cv2Avs>
        <cv2_present>1</cv2_present>
        <cv2>453</cv2>
        <street_address1>37 Queen Street</street_address1>
        <street_address2>Oxford</streetaddress2>
        <postcode>OX20 8FP</postcode>
        <ExtendedPolicy>…</ExtendedPolicy>
      </Cv2Avs>
      <pan>4444**********11</pan>
      <expirydate>12/06</expirydate>
    </Card>
  </CardTxn>
  </Transaction>
</Request>
```

Example XML Request where the billing address is not the same as the customer’s address

```
<Request>
  <Authentication>...</Authentication>
  <Transaction>
    <TxnDetails>
      <Order>
        <Customer>
          <salutation>Mr</salutation>
          <forename>Joe</forename>
          <middlename>K</middlename>
          <surname>Brown</surname>
          <Address>
            <apartment>1</apartment>
            <streetaddress>10 Front Street</streetaddress>
            <city>Edinburgh</city>
            <region>Lothian</region>
            <country>826</country>
            <postcode>EH9 9ZZ</postcode>
          </Address>
          <email>joe.brown@compuserve.com</email>
          <alt_telephone>44 131 999 8888</alt_telephone>
          <ip_address>231.0.0.1</ip_address>
        </Customer>
        <BillingAddress>
          <apartment>6</apartment>
          <streetaddress>Main Street</streetaddress>
          <moreaddress>Newtown</moreaddress>
          <city>London</city>
          <country>826</country>
          <postcode>W1 2ZZ</postcode>
        </BillingAddress>
      </Order>
    ...
    </TxnDetails>
  <CardTxn></CardTxn>
  </Transaction>
</Request>
```

Example XML Request element to override a fraud challenged transaction

```
<Request>
  <Authentication>
    <client>21859999</client>
    <password>thepassword</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>accept_fraud</method>
      <reference>9910670336394837</reference>
    </HistoricTxn>
  </Transaction>
</Request>
```

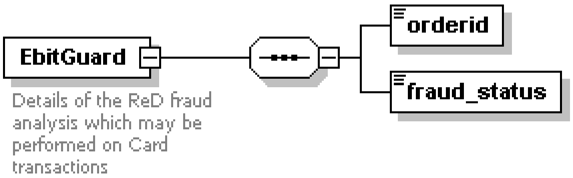
D.3.3. Schema Elements for Response

When a transaction is screened using this service, the general Response elements (section A.1.2) and CardTxn elements (section B.1.3.1) elements will be present. An additional element will also be present for authorisation Responses

D.3.3.1. EbitGuard

This element contains the results of the fraud screening

Element Name:	EbitGuard
Position(s)	Response.CardTxn


EbitGuard
Details of the ReD fraud analysis which may be performed on Card transactions
orderid
fraud_status

Elements of EbitGuard		
Element Name	**description**	**Values**
orderid	The ReD order ID. This uniquely identifies the order	
fraud_status	The overall result of the transaction	Accept Deny Challenge

Attributes for Elements of EbitGuard			
Attribute Name	**Attribute of element**	**description**	**Values**
score	fraud_status	The ReD return code	4 digit numeric.

Example XML elements for EbitGuard

```
<EbitGuard>
  <fraud_status score='0150'>Accept</fraud_status>
  <orderid>00000000000000000000001060764550</orderid>
</EbitGuard>

<EbitGuard>
  <fraud_status score="0800">Deny</fraud_status>
  <orderid>00000100000ADC00000000001118324936</orderid>
</EbitGuard>
```

D.3.4. Example Responses

Example XML Response XML element for fraud declined transaction

```
<Response>
  <CardTxn>
    <card_scheme>VISA Delta</card_scheme>
    <country>United States</country
    <EbitGuard>
      <fraud_status score='0200'>Deny</fraud_status>
      <orderid>00000000000000000000001060764550</orderid>
    </EbitGuard>
  </CardTxn>
  <datacash_reference>4700200040952107</datacash_reference>
  <merchantreference>…</merchantreference>
  <mode>…</mode>
  <reason>FRAUD DECLINED 0200</reason>
  <status>7</status>
  <time>…</time>
</Response>
```

Example XML Response XML element for fraud challenged transaction, authorised by the bank

```
<Response>
  <CardTxn>
    <authcode>278052</authcode>
    <EbitGuard>
      <fraud_status score='0600'>Challenge</fraud_status>
      <orderid>00000000000000000000001060764524</orderid>
    </EbitGuard>
    …
  </CardTxn>
  <datacash_reference>4600200040952400</datacash_reference>
  <merchantreference>00000007</merchantreference>
  <mode>…</mode>
  <reason>FRAUD CHALLENGED 0600</reason>
  <status>7</status>
  <time>…</time>
</Response>
```

Example XML Response XML element for fraud challenged transaction, declined by the bank

```
<Response>
  <CardTxn>
    <authcode>NOT AUTHORISED</authcode>
    <EbitGuard>
      <fraud_status score='0600'>Challenge</fraud_status>
      <orderid>00000000000000000000001060764524</orderid>
    </EbitGuard>
    …
  </CardTxn>
  <datacash_reference>4600200040500007</datacash_reference>
  <merchantreference>00000007</merchantreference>
  <mode>…</mode>
  <reason>DECLINED</reason>
  <status>7</status>
  <time>…</time>
</Response>
```

Example XML Response XML element for an accepted transaction

```
<Response>
  <CardTxn>
    <authcode>743065</authcode>
    <card_scheme>VISA</card_scheme>
    <country>United Kingdom</country>
    <EbitGuard>
      <fraud_status score='0100'>Accept</fraud_status>
      <orderid>0000000000000000000000001060694599</orderid>
    </EbitGuard>
  </CardTxn>
  <datacash_reference>4600200040500130</datacash_reference>
  <merchantreference>00000006</merchantreference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML. A successful accept_fraud transaction

```
<Response>
  <datacash_reference>4600200040952400</datacash_reference>
  <merchantreference>4600200040952400</merchantreference>
  <mode>…</mode>
  <reason>ACCEPT FRAUD CHALLENGE OK</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML. A failed accept_fraud

```
<Response>
  <datacash_reference>4300200040500382</datacash_reference>
  <merchantreference>4600200040500007</merchantreference>
  <mode>…</mode>
  <reason>accept_fraud attempted on CV2AVS DECLINED
       transaction</reason>
  <status>19</status>
  <time>…</time>
</Response>
```

D.4. 3-D Secure, with DataCash MPI

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/fraud_prevention/3D-Secure/DC-mpi.shtml

This service is utilised by sending a normal Credit and Debit Card Service Request with several pieces of extra information. This section of the documentation assumes the reader is familiar with the Credit and Debit Card Service, as described in section B.1.

It is also possible to use this service in conjuction with the Pre-Registered Card Service, as outlined in section C.1.

The following key will be used in this section of the document:

- R – Required
- M –Mandatory if 3D-Secure check is to be performed
- O – Optional

Cardholder Verification Check
The authand pre transaction types require the same information about the transaction to be provided. This data is passed using the following schema elements:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `CardTxn` - details about the card, section B.1.1.2
    - `TxnDetails` - contains details of the transaction, section B.1.1.3
      - `ThreeDSecure` – contains all the details required to initiate the 3-D Secure check, section D.4.1.1
        - `Browser` – details of the browser the cardholder is using, section D.4.1.2

Authorization and Referred Authorization
The `threedsecure_authorization_request` and `threedsecure_authorize_referral_request` transaction types require information to be provided in the `HistoricTxn` element:

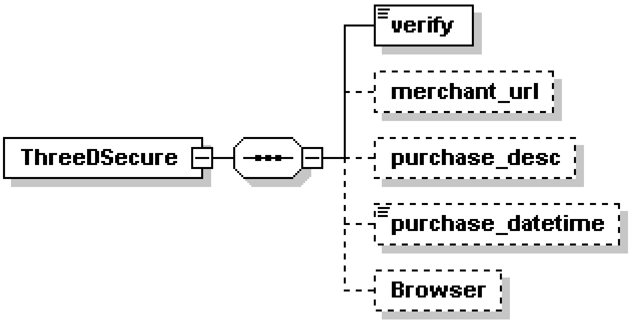
- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `HistoricTxn` – section D.4.1.3

D.4.1. Schema Elements for Request

D.4.1.1. ThreeDSecure

This element contains all of the extra details which are required for the 3-D Secure check to be initiated and should be provided for `auth` and `pre` Requests.

Element Name: `ThreeDSecure`
Position: `Request.Transaction.TxnDetails`



Elements of ThreeDSecure			
Element Name	**description**	**values / limitations**	**required**
`verify`	Indicates whether the transaction should be checked for enrolment in the 3-D Secure system	`yes` `no`	R
`merchant_url`	The URL of the website on which the payment is being made	1-2048 characters. Must be fully qualified (i.e. include `https://`)	M
`purchase_desc`	A short description of what has been purchased. This is shown to the card holder by the ACS.	1-125 characters	M
`purchase_datetime`	The date and time of the transaction	`YYYYMMDD HH:MM:SS` format	M
`Browser`	See section D.4.1.2		M

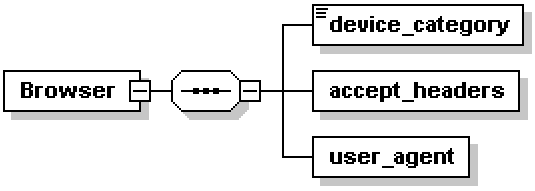
Example XML for ThreeDSecure complex elements

```
<ThreeDSecure>
  <verify>yes</verify>
  <merchant_url>https://www.widgets.com</merchant_url>
  <purchase_desc>DVD</purchase_desc>
  <purchase_datetime>20060731 21:59:42</purchase_datetime>
  <Browser>…</Browser>
</ThreeDSecure>

<ThreeDSecure>
  <verify>no</verify>
</ThreeDSecure>
```

D.4.1.2. Browser

The Browser element contains all the information about the browser the cardholder is using to access your website, and must be provided for auth and pre requests where the 3-D Secure check is to be performed.

Element Name:	Browser
Position:	Request.Transaction.TxnDetails.ThreeDSecure



Elements of Browser			
Element Name	**description**	**values / limitations**	**required**
device_category	Indicates the type of device used to the transaction	0 – for PC 1 – for mobile device	R
accept_headers	The MIME types of the headers accepted by this device		R
user_agent	The User Agent associated with the device		R

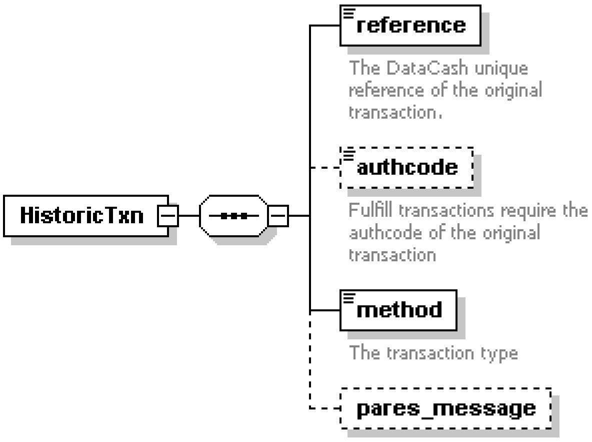
Example XML for Browser complex elements

```
<Browser>
  <device_category>0</device_category>
  <accept_headers>*/*</accept_headers>
  <user_agent>IE/6.0</user_agent>
</Browser>

<Browser>
  <device_category>0</device_category>
  <accept_headers>txt/xml, application/xml, application/xhtml+xml,
      text/html;q=0.9, text/plain;q=0.8,video/x-mng, image/png,
      image/jpeg, image/gif;q=0.2, text/css, */*;q=0.1
  </accept_headers>
  <user_agent>Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en)
      AppleWebKit/417.9 (KHTML, like Gecko) Safari/417.9.2
  </user_agent>
</Browser>
```

D.4.1.3. HistoricTxn

The `HistoricTxn` element is used for `threedsecure_authorization_request`, `threedsecure_authorize_referral_request` and `fulfill` transactions. It enables these transactions to be tied back to the original transaction.

Element Name:	`HistoricTxn`
Position:	`Request.Transaction`


reference
The DataCash unique reference of the original transaction.
HistoricTxn
authcode
Fulfill transactions require the authcode of the original transaction
method
The transaction type
pares_message

Elements of HistoricTxn					
Element Name	**description**	**values / limitations**	**3DS ref[1]**	**3DS aut[2]**	**3DS ful[3]**
`reference`	The datacash_reference number for the transaction	16 digits	R	R	R
`authcode`	The authorisation code provided by your Bank's Authorisation Centre	-	R	n/a	R
`method`	The transaction type	See bullets below	R	R	R
`pares_message`	The Payer Authentication Response (PARes) returned by the ACS	As returned by the ACS	O	O	n/a

1. `threedsecure_authorize_referral_request`
2. `threedsecure_authorization_request`
3. `fulfill`

Attributes of elements within HistoricTxn				
Attribute Name	**Attribute of Element**	**description**	**values / limitations**	**required**
`tx_status_u`	`method`	Accepts the transaction if the ACS is unable to complete the validation of the card holder	`accept`	O

Example XML for HistoricTxn complex elements

```
<HistoricTxn>
  <reference>4400200042810513</reference>
  <method>threedsecure_authorization_request</method>
  <pares_message>uyt45t89cnwu3rhc98a4hterjklth4o8c
      tsrjzth4</pares_message>
</HistoricTxn>

<HistoricTxn>
  <reference>4400200042810513</reference>
  <method tx_status_u=”accept”>
     threedsecure_authorization_request</method>
  <pares_message>4o7sbvohyv8s4075mytudriotu0359ucmi
     udtoyuu</pares_message>
</HistoricTxn>
```

Example XML for HistoricTxn complex elements for transactions which cannot be 3-D Secure checked (e.g. card scheme not supported, non-enrolled cards)

```
<HistoricTxn>
  <reference>4400200042810513</reference>
  <method>threedsecure_authorization_request</method>
</HistoricTxn>
```

Example XML for HistoricTxn complex elements for referred authorisation

```
<HistoricTxn>
  <reference>4400200042810513</reference>
  <authcode>12AB52</authcode>
  <method>threedsecure_authorize_referral_request</method>
  <pares_message>w5670w5689kg74u76490wf
       uvaero5ielktekt=vre</pares_message>
</HistoricTxn>
```

Example XML for HistoricTxn complex elements for a fulfill

```
<HistoricTxn>
  <reference>4400200042810513</reference>
  <authcode>12AB52</authcode>
  <method>fulfill</method>
</HistoricTxn>
```

D.4.2. XML Example Requests

D.4.2.1. Cardholder Verification Requests

Example XML for pre transactions which by-pass the 3-D Secure check

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>387545924AdkWdd</merchantreference>
      <amount currency="GBP">10.04</amount>
      <ThreeDSecure>
        <verify>no</verify>
      </ThreeDSecure>
    </TxnDetails>
    <CardTxn>
      <Card>
        <pan>4444************</pan>
        <expirydate>06/12</expirydate>
      </Card>
      <method>pre</method>
    </CardTxn>
  </Transaction>
</Request>

<Request>
  <Authentication>
    <client>99000001</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>4564234523</merchantreference>
      <amount currency="AUD">59.45</amount>
      <capturemethod>ecomm</capturemethod>
      <ThreeDSecure>
        <verify>no</verify>
        <merchant_url>https://www.mywebsite.com/12</merchant_url>
        <purchase_desc>CDs and DVDs</purchase_desc>
        <purchase_datetime>20060531 13:06:28</purchase_datetime>
        <Browser>
          <device_category>0</device_category>
          <accept_headers>*/*</accept_headers>
          <user_agent>IE/6.0</user_agent>
        </Browser>
      </ThreeDSecure>
    </TxnDetails>
    <CardTxn>
      <Card>
        <pan>4444************</pan>
        <expirydate>06/12</expirydate>
      </Card>
      <method>pre</method>
    </CardTxn>
  </Transaction>
</Request>
```

Example XML for an auth request, requesting verification

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>387545924537037</merchantreference>
      <amount currency="GBP">10.04</amount>
      <capturemethod>ecomm</capturemethod>
      <ThreeDSecure>
        <verify>yes</verify>
        <merchant_url>https://www.mywebsite.com</merchant_url>
        <purchase_desc>CDs and DVDs</purchase_desc>
        <purchase_datetime>20060201 23:59:59</purchase_datetime>
        <Browser>
          <device_category>0</device_category>
          <accept_headers>*/*</accept_headers>
          <user_agent>IE/6.0</user_agent>
        </Browser>
      </ThreeDSecure>
    </TxnDetails>
    <CardTxn>
      <Card>
        <pan>4444************</pan>
        <expirydate>06/12</expirydate>
      </Card>
      <method>auth</method>
    </CardTxn>
  </Transaction>
</Request>
```

Example XML for a Pre-Registered Card request, requesting verification

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>387545924537037</merchantreference>
      <amount currency="GBP">10.04</amount>
      <capturemethod>ecomm</capturemethod>
      <ThreeDSecure>
        <verify>yes</verify>
        <merchant_url>https://www.mywebsite.com</merchant_url>
        <purchase_desc>CDs and DVDs</purchase_desc>
        <purchase_datetime>20060201 23:59:59</purchase_datetime>
        <Browser>
          <device_category>0</device_category>
          <accept_headers>*/*</accept_headers>
          <user_agent>IE/6.0</user_agent>
        </Browser>
      </ThreeDSecure>
    </TxnDetails>
    <CardTxn>
      <card_details type="preregistered">
         4100200043070870</card_details>
      <method>auth</method>
    </CardTxn>
  </Transaction>
</Request>
```

D.4.2.2. Historic Transactions

Example XML for a threedsecure_authorization_request, with PARes

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <reference>4900200042810407</reference>
      <method tx_status_u=”accept”>
        threedsecure_authorization_request</method>
      <pares_message>w48gj6s5ty45ty8cviozdtrjgd</pares_message>
    </HistoricTxn>
  </Transaction>
</Request>
```

Example XML for a threedsecure_authorization_request without a PARes. E.g. for a non-enrolled card, or a card scheme which is not supported by 3-D Secure

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <reference>4100200043070078</reference>
      <method>threedsecure_authorization_request</method>
    </HistoricTxn>
  </Transaction>
</Request>
```

Example XML for a threedsecure_authorize_referral_request

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <reference>4100200043070078</reference>
      <auth_code>ZY987A</auth_code>
      <method>threedsecure_authorize_referral_request</method>
    </HistoricTxn>
  </Transaction>
</Request>
```

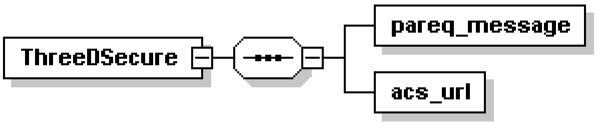
Please refer to section B.1.2.2 for an example `fulfill` request

D.4.3. Schema Elements for Response

D.4.3.1. ThreeDSecure

This element will be returned in response to a card enrolment check, if the card is enrolled. If the card is not enrolled or is not supported, this element will not be present.

Element Name:	`ThreeDSecure`
Position:	`Response.CardTxn`



Elements of ThreeDSecure	
Element Name	**description**
`pareq_message`	The PAReq
`acs_url`	The URL of the ACS, to which the cardholder needs to be re-directed. In the test environment this will be the URL of the testserver ACS. In the live environment this will the the URL of the Issuing Bank's ACS.

Example XML for ThreeDSecure complex elements

```
<ThreeDSecure>
  <acs_url>https://secure.barclaycard.co.uk/barclays/tdsecure/pa.jsp?
        partner=barclaycard.visa&amp;VAA=B</acs_url>
  <pareq_message>eJxdUltugzAQ/M8pUA+AHyEQKscSLR/NB1HU5AKWsypIBRIb
     StrT14Y4JkGAdnYWdjRjdiwVQH4A2SvgiyBgBWgtviCoTpuXswgjuiZ4tU
     xeLGnoffYJl6k26AeUrtqGkxCHlCEHHV2AkqVoOtcwLSEvb9sdj5MkSglD
     N+j5GtQ252lKcJxgzNCEPd+IGvhBKFEyNNaekm3fdOqXr2nMkAOe7tU3H4
     YhPIlOSKHLULY1Q7br5KJnvWzf24aeb7lWJ17k2TB/dvk2Ko4ZKf7khiE7
     4efNOuAU4xhTTAK6fF2l5mZo7M98qa1gTnCII2PLhDx9tkIyN2NH5p2ZB7
     1S0Ehngk7ntgHzjYnrXs/Ugpa8A90Zgba8W/PsBHv/eEhVdiYmowwv7Wu6
     xnxH4mF/ZWIh1LI3cF/i/mnWuWNmQxkPJF8w9HhY/w</pareq_message>
</ThreeDSecure>
```

D.4.4. XML Example Responses

D.4.4.1. Cardholder Verification Response

Example XML Response for a card which is not enrolled

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <CardTxn>
    <card_scheme>Visa</card_scheme>
    <country>United Kingdon</country>
    <issuer>Prudential Banking PLC</issuer>
  </CardTxn>
  <datacash_reference>4300200042810617</datacash_reference>
  <merchantreference>387546093287037</merchantreference>
  <mode>…</mode>
  <reason>3DS Card not Enrolled</reason>
  <status>162</status>
  <time>…</time>
</Response>
```

Example XML Response for a card which is enrolled

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <CardTxn>
    <card_scheme>Mastercard</card_scheme>
    <country>United Kingdon</country>
    <ThreeDSecure>
      <acs_url>https://www.clicksafe.lloydstsb.com/Lloyds/
         tdsecure/pa.jsp?partner=mc&amp;VAA=B</acs_url>
      <pareq_message>eJxdUltugzAQ/M8pUA+AHyEQKscSLR/NB1H
         U5AKWsypIBRIbStrT14Y4JkGAdnYWdjRjdiwVQH4A2Svgiy
         BgBWgtviCoTpuXswgjuiZ4tUxeLGnoffYJl6k26AeUrtqGk
         xCHlCEHHV2AkqVoOtcwLSEvb9sdj5MkSglDN+j5GtQ252lK
         cJxgzNCEPd+IGvhBKFEyNNaekm3fdOqXr2nMkAOe7tU3H4Y
         hPIlOSKHLULY1Q7br5KJnvWzf24aeb7lWJ17k2TB/dvk2Ko
         4ZKf7khiE74efNOuAU4xhTTAK6fF2l5mZo7M98qa1gTnCII
         2PLhDx9tkIyN2NH5p2ZB71S0EhngkN+AK7ntgHzjYnrXs/U
         gpa8A90Zgba8W/PsBHv/eEhVdiYmowwv7Wu6xnxH4mF/ZWI
         h1LI3cF/i/mnWuWNmQxkPJF8w9HhY/wFTsbsh
      </pareq_message>
    </ThreeDSecure>
  </CardTxn>
  <datacash_reference>4300200042810537</datacash_reference>
  <merchantreference>387546049537037</merchantreference>
  <mode>…</mode>
  <reason>3DS Payer Verification Required</reason>
  <status>150</status>
  <time>…</time>
</Response>
```

D.4.4.2. Cardholder Authorisation Responses

Example XML Response when the PARes returned is invalid

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>3000100226001681</datacash_reference>
  <information>[2:Error] Expected PARes node: got
     &apos;Error&apos;</information>
  <merchantreference>3200100226001675</merchantreference>
  <mode>…</mode>
  <reason>3DS invalid pares</reason>
  <status>176</status>
  <time>…</time>
</Response>
```

Example XML Response were a referral response is received

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <CardTxn>
    <card_scheme>Visa</card_scheme>
    <country>United Kingdom</country>
    <issuer>Prudential Banking PLC</issuer>
  </CardTxn>
  <datacash_reference>3000100226001681</datacash_reference>
  <merchantreference>13542DSDWK21</merchantreference>
  <mode>…</mode>
  <reason>3DS call auth centre</reason>
  <status>161</status>
  <time>…</time>
</Response>
```

D.4.4.3. Authorisation of Referred Cards

Example XML Response were a referral response to a three3dsecure_authorize_referral_request

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4900200043070079</datacash_reference>
  <merchantreference>13542DSDWK21</merchantreference>
  <mode>…</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

D.4.5. Using inline “Redirect Page”

Due to the increasing use of software known as “Popup Killers”, Visa and MasterCard are recommending that merchants be aware that a “Popup” window may not be the best way to direct cardholders to their issuer for authentication. Indeed, the 3-D Secure specifications have explicitly banned the use of “Popup” windows as of October 2004.

The recommended way to allow Cardholders to authenticate themselves is via an in-line Redirect Page generated either via an `<iframe />` type system or as a separate “framed” page.

Example using the <iframe /> mechanism

```
<html>
 <head>
  <title>Please Authenticate</title>
 </head>
 <body OnLoad="OnLoadEvent();" >
  <form name="downloadForm"
   action="https://mybank.com/vbyv/verify" method="POST">
   <input type="hidden" name="PaReq" value="AAABBBBCCCCHHHHHH=">
   <input type="hidden" name="TermUrl" value="https:// www.
      MyWidgits.Com/next.cgi">
   <input type="hidden" name="MD" value="200304012012a">
  </form>
  <script language="Javascript" >
   <!--
    function OnLoadEvent(){
    document.downloadForm.target = "ACSframe";
    document.downloadForm.submit();
    }
   //-->
  </script>
  <!-- MERCHANT TO FILL IN THEIR OWN BRANDING HERE -->
  <iframe src="blank.htm" name="ACSframe" width="390"
     height="450" frameborder="0"> </iframe>
  <!-- MERCHANT TO FILL IN THEIR OWN BRANDING HERE -->
 </body>
</html>
```

In this example, no Popup window is created. The ACS dialogue takes place within a floating frame on your page. After the cardholder has authenticated themselves they will be returned to your site by their issuers ACS posting the results to the script defined in the TermURL variable.

D.4.6. ACS Simulator

D.4.6.1. Introduction

The DataCash ACS simulator enabled you to simulate the behaviour of an Issuing Bank ACS. This includes sending an authenticated response, a non-authenticated response or a range of invalid and error responses.

It is intended that the ACS simulator be used during merchant integration.

D.4.6.2. How to use the ACS simulator

The URL of the ACS that the cardholder is to be directed to is returned in the Enrolment Check Response XML Document. In a test environment this will be the URL of the DataCash ACS simulator.

The simulator gives you control over what response is returned i.e. whether to simulate that the cardholder managed to authenticate themselves or whether an error occurred.

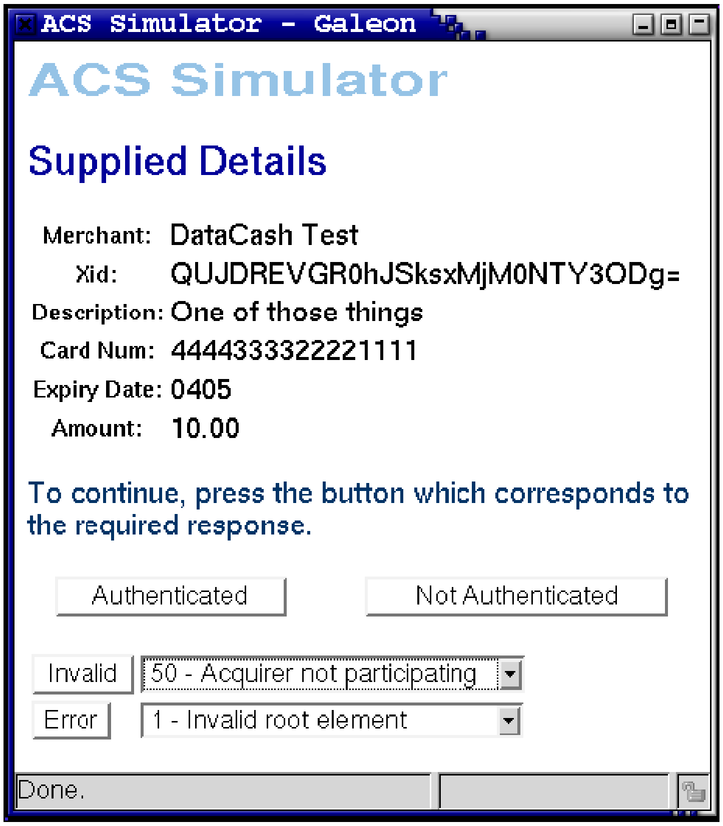
The following fields have to be supplied to the ACS URL using the POST method.

Field	Description
`PaReq`	The PaReq that was returned in the same Enrolment Check Response as the ACS URL.
`TermURL`	The URL of the script that will process the response from the ACS.
`MD`	The MD field gives you a mechanism of tracing transactions. This field should be populated with a unique value. This value will be returned to you, with the PaRes message, when the ACS has finished.

If any of these fields are missing then an error response will be displayed.

The output from the ACS Simulator should be displayed in a popup window, separate from the original checkout page.

Example output from the ACS Simulator can be seen opposite.



ACS Simulator

Supplied Details

Merchant: DataCash Test
Xid: QUJDREVGR0hJSksxMjM0NTY3ODg=
Description: One of those things
Card Num: 4444333322221111
Expiry Date: 0405
Amount: 10.00

To continue, press the button which corresponds to the required response.

At the top of the page is information that is contained in the PaReq message. This can be used to check that the correct details are being processed. The page also contains four buttons that give you the ability emulate responses from a real ACS. These buttons are:

Button	Description
Authenticated	The cardholder successfully authenticated himself or herself.
Not Authenticated	The cardholder failed to authenticate himself or herself.
Invalid	This emulates the response that will be returned from an invalid request. You can select which type of invalid response will be returned using the drop down menu. See the table below.
Error	This emulates a protocol level error, for example, badly formatted data. Again, the kind of error can be selected using the drop down menu.

The following table describes the invalid responses that can be emulated using the drop down menu:

Invalid Response	Description
50 - Acquirer not participating	Acquirer not participating in 3-D Secure.
51 - Merchant not participating	Merchant not participating in 3-D secure.
52 - Password not supplied	Password required, but no password was supplied.
53 - Password invalid	Supplied password is not valid for combination of Acquirer BIN and Merchant ID.
54 - Invalid ISO code	ISO code not valid per ISO tables. This can apply to either the currency or the country value.
55 - Transaction not valid	Transaction data not valid. For example, the amount specified does not match the purchase amount.
56 - PAReq incorrectly routed	PAReq was incorrectly routed. This would happen if the PAReq was sent to the wrong ACS or if a PAReq should never have been sent.
57 - No serial number	Serial number cannot be located.
98 - Transient system failure	Transient system failure. Such as system is currently overloaded.
99 - Permanent system failure	Permanent system failure. For example, a main disk has gone down.

Similarly, the error drop down menu can be used to select which error response is to be generated.

Error Code	Description
1 - Invalid root element	Root element is not `<ThreeDSecure>`
2 - Not a defined message	Message element is not a `PAReq`.
3 - Missing element	A required element is missing from the `PAReq`.
4 - Critical not recognized	Critical element was not recognized.
5 - Invalid format	Format of at least one element is invalid.
6 - Protocol too old	The protocol version of the supplied message is too old.
98 - Transient system failure	Transient system failure i.e. system is currently busy.
99 - Permanent system failure	Permanent system failure i.e. the database server is down.

Once one of the four buttons has been pressed the popup window will close and the output of the return script, as specified in the TermURL field, will be displayed in the original checkout window.

The following fields will be supplied to the return script using the POST method:

Field	Description
`PaRes`	A cryptographically signed XML document used to indicate whether or not the cardholder managed to authenticate. It should be returned to DataCash in the `threedsecure_authorization_request` message. If an error occurs at the ACS, the PaReS will be populated with an error message, but this should not be interpreted or examined by the merchant
`MD`	This is the value that was passed with the original request to the ACS simulator.

D.4.6.3. Magic Card Functionality

DataCash already provides a magic card facility. This allows particular responses to be received from the DataCash payment gateway when using one of the magic card numbers.

The existing list of magic cards can be found in the Developers Area.

DataCash has extended this functionality to allow certain Enrolment Verification Request responses to be obtained. This is achieved by specifying any of the magic card numbers with an appropriate value for the month in the card expiry date field.

The possible options are:

Value of month in expiry date	Type of Response returned	Return Code
`01`	Card is enrolled	150
`02`	Card is not enrolled	162
`03`	No result received from the directory server	159
`04`	An invalid response is received from the directory server	160
`05`	The directory server is running the wrong protocol (SET instead of 3-D Secure)	160
`06`	3DS Invalid VEReq	186
Any other value	'Unable to process' response is received from the directory server	187

Please see the Developers Area and Support Centre for information on the above return codes.

It should also be noted that if a non-magic card were used in a 3-D Secure transaction submitted against the testserver host, a response with return code 169 would be received.

D.4.7. MPI Only

D.4.7.1. Summary of Service

Using this service, it is possible for merchants to perform 3-D Secure authentication of a transaction using the DataCash MPI, and authorize the transaction with a 3rd party Payment Gateway.

In order to use the decoupled DataCash MPI for 3-D Secure transactions a variety of different transaction types are required. Initially an enrolment check transaction must be performed which will contain all details required to initiate the 3-D Secure authentication process as well as transaction details relating to the MPI-Only transaction. The response to this message will indicate whether the cardholder is enrolled.

If the cardholder is enrolled this can be followed by a validation authentication transaction containing the PARes message returned from the ACS and a historic reference. A successful response to this transaction will provide the merchant with enough 3-D Secure information to allow the merchant to authorize the transaction with a 3rd party Payment Gateway.

Alternatively, a merchant can optionally authorize the transaction using the DataCash Payment Gateway.

D.4.7.2. Schema elements for Request

This element contains all of the extra details which are required for the 3-D Secure check to be initiated and should be provided for `auth` and `pre` Requests.

Element Name: `MpiTxn`
Position: `Request.Transaction`

Elements of MpiTxn			
Element Name	**description**	**values / limitations**	**required**
`method`	The method of the transaction.	`mpi`	R
`Card`	The card details of the transaction to be authenticated. This element contains sub elements. See B.1.1.1 for more information	See B.1.1.1 for more information	M
`card_details`	If subscribed to the preregistered card service, this element can be used instead of the `Card` block above. See D.4.7.5 for an example. See C.1.1.1 for more information about the preregistered card service.	XML Attribute `type=”preregistered”` should be supplied in the opening tag. The value contained should be the datacash_reference of the card transaction.	M

D.4.7.3. End to end walkthrough: Cardholder enrolled

The following scenario uses the DataCash MPI to successfully authorize a 3-D Secure transaction where the cardholder is enrolled.

D.4.7.3.1. Enrolment Check

The first stage of 3-D Secure authentication is to perform an enrolment check. The enrolment check transaction is similar to the existing Cardholder Verification Check message type as used to perform normal (i.e. coupled with Card Authorization) 3-D Secure transactions with the DataCash MPI. The main difference between these two transaction types being that the enrolment check transaction holds card details within the MpiTxn block.

It should be noted that neither Cv2Avs data, nor the Verify element will be allowed in the enrolment check transaction. CV2/AVS data can be provided in a subsequent authorization transaction if required.

Example Enrolment Check Request

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>********</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>387545924537037</merchantreference>
      <amount currency="GBP">10.04</amount>
      <ThreeDSecure>
        <merchant_url>www.mywebsite.com</merchant_url>
        <purchase_desc>CDs and DVDs</purchase_desc>
        <purchase_datetime>20060201 23:59:59</purchase_datetime>
        <Browser>
          <device_category>0</device_category>
          <accept_headers>*/*</accept_headers>
          <user_agent>IE/6.0</user_agent>
        </Browser>
      </ThreeDSecure>
    </TxnDetails>
    <MpiTxn>
      <method>mpi</method>
      <Card>
        <pan>4444************</pan>
        <expirydate>06/12</expirydate>
        <startdate>06/12</startdate>
        <issuenumber>06/12</issuenumber>
      </Card>
    <MpiTxn>
  </Transaction>
</Request>
```

The following is an example of the xml message response for an enrolled card. The response contains the encoded PAReq message along with an ACS URL link.

Example Enrolment Check Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
   <status>150</status>
   <reason>3DS Payer Verification Required</reason>
   <merchantreference>387546049537037</merchantreference>
   <datacash_reference>4300200042810537</datacash_reference>
   <time>…</time>
   <mode>…</mode>
   <CardTxn>
      <ThreeDSecure>
        <pareq_message>eJxdUltugzAQ/M8pUA+AHyEQKscSLR/NB1H
           U5AKWsypIBRIbStrT14Y4JkGAdnYWdjRjdiwVQH4A2Svgiy
           BgBWgtviCoTpuXswgjuiZ4tUxeLGnoffYJl6k26AeUrtqGk
           xCHlCEHHV2AkqVoOtcwLSEvb9sdj5MkSglDN+j5GtQ252lK
           cJxgzNCEPd+IGvhBKFEyNNaekm3fdOqXr2nMkAOe7tU3H4Y
           hPIlOSKHLULY1Q7br5KJnvWzf24aeb7lWJ17k2TB/dvk2Ko
           4ZKf7khiE74efNOuAU4xhTTAK6fF2l5mZo7M98qa1gTnCII
           2PLhDx9tkIyN2NH5p2ZB71S0EhngkN+AK7ntgHzjYnrXs/U
           gpa8A90Zgba8W/PsBHv/eEhVdiYmowwv7Wu6xnxH4mF/ZWI
           h1LI3cF/i/mnWuWNmQxkPJF8w9HhY/wFTsbsh
       </pareq_message>
       <acs_url>https://www.clicksafe.lloydstsb.com/Lloyds/
          tdsecure/pa.jsp?partner=mc&amp;VAA=B</acs_url>
     </ThreeDSecure>
   </CardTxn>
</Response>
```

D.4.7.3.2. Validate Authentication

In order to authenticate an enrolled card, the merchant is required to submit a 'threedsecure_validate_authentication' transaction. This request will use the datacash_reference from the enrolment check as the historic reference along with the resulting PARes message received from the ACS.

An example threedsecure_validate_authentication transaction is shown below:

Example Validate Authentication Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>99000001 </client>
    <password>********</password>
  </Authentication>
  <Transaction>
      <HistoricTxn>
         <reference>4300200042810537</reference>
         <method>threedsecure_validate_authentication</method>
         <pares_message>
uyt45t89cnwu3rhc98a4hterjklth4o8ctsrjzth4</pares_message>
      </HistoricTxn>
  </Transaction>
</Request>
```

The response to a threedsecure_validate_authentication where the cardholder was successfully authenticated is shown below. This response contains the necessary fields should the merchant wish to authorise the transaction using a 3rd Party Payment Gateway.

Example Validate Authentication Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
<CardTxn>
    <ThreeDSecure type='visa'>
      <cardholder_registered>yes</cardholder_registered>
      <eci>05</eci>
      <security_code>Q0FWVkNBVlZDQVZWQ0FWVkNBVlY=</security_code>
      <xid>MDAwMDAwMDAwMDAwMTIzNDU2NzE=</xid>
    </ThreeDSecure>
  </CardTxn>
  <datacash_reference>3200900012345671</datacash_reference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>_UNIX_TIME_</time>
</Response>
```

D.4.7.3.3. Authorization via the DataCash Payment Gateway

The subsequent authorization of an MPI Only transaction by the DataCash Payment Gateway is entirely optional. Enough information will have been returned to the merchant by the DataCash Payment Gateway in response to the validation authentication transaction to permit authorization using 3-D Secure with an alternative Payment Service Provider.

An example authorization request with the DataCash Payment Gateway is illustrated below. This authorization is similar to that used for pre-registered card transactions. The method types permitted for a subsequent authorization attempts are `auth` or `pre` and the type of the transaction must state `from_mpi`.
CV2/AVS details can be optionally provided with the authorization request as part of CardTxn details using the new Card element.

Additionally a transaction amount can also optionally be supplied. If a new amount is supplied, then the transaction currency must also be provided and must match the currency presented in the original enrolment check. Where a new amount is specified, that amount will be used instead of the amount supplied in the original enrolment check.

If the new amount is specified in an authorisation is greater than that specified in the original authentication transaction, it is at the discretion and risk of the merchant. DataCash make no claims with regard to liability shift.

Example Authorisation Request with the Datacash Payment Gateway

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>99000001</client>
    <password>*******</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount currency="GBP">1001.02</amount>
      <merchantreference>387545924AdkWdd</merchantreference>
    </TxnDetails>
    <CardTxn>
      <method>auth</method>
      <card_details
type="from_mpi">3200900012345671</card_details>
      <Card>
      <Cv2Avs>
        <street_address1>1 High Street</street_address1>
        <street_address2>This Town Town</street_address2>
        <street_address3>Somewhere</street_address3>
        <street_address4>United Kingdom</street_address4>
        <postcode>S01 2CD</postcode>
        <cv2>123</cv2>
      </Cv2Avs>
      </Card>
    </CardTxn>
  </Transaction>
</Request>
  <status>1</status>
  <time>_UNIX_TIME_</time>
</Response>
```

The response to a subsequent authorization of an MPI Only transaction will be as per standard bankcard responses.

Example Authorisation Response from the Datacash Payment Gateway

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <CardTxn>
    <authcode>100000</authcode>
    <card_scheme>VISA</card_scheme>
    <country>United Kingdom</country>
  </CardTxn>
  <datacash_reference>3000900012345672</datacash_reference>
  <merchantreference>4421000009</merchantreference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>_UNIX_TIME_</time>
</Response>
```

D.4.7.4. End to end walkthrough: Cardholder not enrolled

As can be seen in the example xml response message below, where the cardholder is not enrolled the ECI value will be returned to the merchant. If the merchant wishes to continue with an authorization, they can do so using an alternative PSP or via the DataCash Payment Gateway.

D.4.7.4.1. Enrolment Check

An example response whereby the cardholder is not enrolled is shown below. This response will contain an ECI value indicating that the cardholder authentication could not be performed.

Example Enrolment Check Response – Cardholder not enrolled

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
   <CardTxn>
      <ThreeDSecure type=”visa”>
        <cardholder_registered>no</cardholder_registered>
        <eci>06</eci>
     </ThreeDSecure>
   </CardTxn>
   <datacash_reference>4300200042810537</datacash_reference>
   <merchantreference>387546049537037</merchantreference>
   <mode>LIVE</mode>
   <status>162</status>
   <reason>3DS Card not Enrolled</reason>
   <time>__UNIX_TIME__</time>
</Response>
```

D.4.7.4.2. Validate Authentication

For an MPI Only transaction with a cardholder that is not enrolled, a threedsecure validate authentication transaction will not be permitted. Such transactions will be rejected by the DPG.

The following example response message would be returned in response to a threedsecure_validate_authentication where the enrolment check determined that the cardholder is not enrolled.

Example Validate Authentication Response – Cardholder not enrolled

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>3200900012345671</datacash_reference>
  <mode>LIVE</mode>
  <reason>Authentication can only be performed for enrolled
cards</reason>
  <status>600</status>
  <time>_UNIX_TIME_</time>
</Response>
```

D.4.7.4.3. Authorization via the DataCash Payment Gateway

Authorization of an MPI Only transaction where the cardholder was not enrolled can optionally be performed via the DataCash Payment Gateway. The request and response messages for a successful authorization are as shown in the section for the Authorization of an enrolled card.

D.4.7.5. Preregistered Card support

Preregistered cards can be used in the ‘MpiTxn’ segment of the Enrolment check instead of the ‘Card’ block.

The following scenario uses the MPI to authenticate a 3-D Secure transaction using preregistered card details.

The enrolment check specifies which earlier (successful) transaction to use to retrieve the card details as highlighted below:

Example Enrolment Check using Preregistered Card Details

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>fred</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>387545924537037</merchantreference>
      <amount currency="GBP">10.04</amount>
      <ThreeDSecure>
        <merchant_url>http://www.mywebsite.com</merchant_url>
        <purchase_desc>CDs and DVDs</purchase_desc>
        <purchase_datetime>20080808 23:59:59</purchase_datetime>
        <Browser>
          <device_category>0</device_category>
          <accept_headers>*/*</accept_headers>
          <user_agent>IE/6.0</user_agent>
        </Browser>
      </ThreeDSecure>
    </TxnDetails>
    <MpiTxn>
      <method>mpi</method>
      <card_details
type=’preregistered’>4300200042810536</card_details>
    <MpiTxn>
  </Transaction>
</Request>
```

DataCash will use the specified transaction to retrieve the card details to be used for the new transaction.

The process of completing 3-D Secure authentication using the preregistered card service differs only in how the card details are specified in the initial enrolment check transaction. All responses will be as if the card details had been supplied explicitly with the exception of the additional return code of 250 indicating that the details for the supplied reference cannot be found.

D.5. 3-D Secure, with 3rd party MPI

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/fraud_prevention/3D-Secure/mpi.shtml

This service is utilised by sending a normal Credit and Debit Card Service Request with up to four pieces of extra information. This section of the documentation assumes the reader is familiar with the Credit and Debit Card Service, as described in section B.1.

The service may also be used in conjunction with the Pre-Registered Card service, as outlined in section C.1

D.5.1. Schema Elements for Request

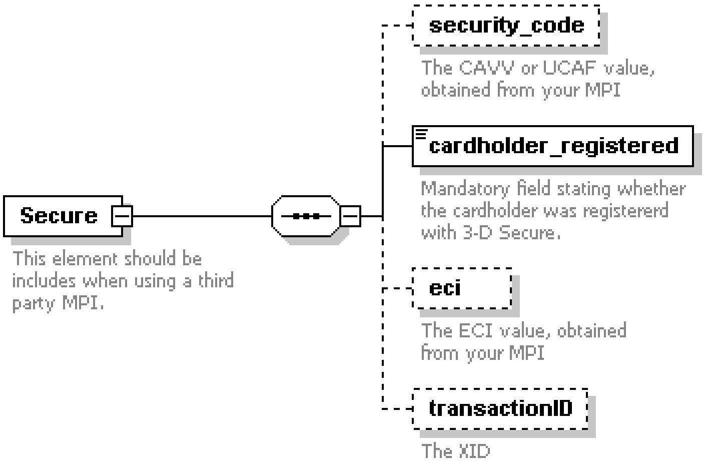
The data for the service is passed in five distinct places in the schema. The additional information for this Service is all passed in the `Secure` element:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `CardTxn` – as described in B.1.1.2
      - `Card –` refer to section B.1.1.1
      - `Secure` - up to four extra pieces of information about the card transaction, section D.5.1.1
    - `TxnDetails` – section B.1.1.3

D.5.1.1. Secure

The Secure element contains the extra information for the 3-D Secure transaction. The data within this element is obtained from your MPI

Element Name:	Secure
Position:	Request.Transaction.TxnDetails.CardTxn


security_code
The CAVV or UCAF value, obtained from your MPI
cardholder_registered
Mandatory field stating whether the cardholder was registererd with 3-D Secure.
Secure
This element should be includes when using a third party MPI.
eci
The ECI value, obtained from your MPI
transactionID
The XID

Elements of Secure			
Element Name	**description**	**values / limitations**	
security_code	The Cardholder Authentication Verification Value (CAVV) for Visa cards, or Universal Cardholder Authentication Field (UCAF) for MasterCard		M
cardholder_registered	Indicates whether the cardholder was registered for 3-D Secure and the PARes / VERes status	yes no attempted – enrolled, PARes status ‘A’ ch_enrolled_u – VERes status ‘U’ tx_status_u – enrolled, PARes status ‘U’	R
eci	The Electronic Commerce Indicator (ECI) obtained from your MPI	01 02 05 06 00	M
transactionID	The XID from PARes.Purchase.xid	As copied from the PARes	M

Attributes of Secure				
Attribute name	**Attribute of Element**	**description**	**values / limitations**	
type	Secure	Indicates the Card Scheme	visa ucaf	R

Example Secure complex element

```
<Secure type='visa'>
  <cardholder_registered>no</cardholder_registered>
</Secure>

<Secure type='ucaf'>
  <security_code>AAABAwRzYAAAAAAAAAAAAAAAAAA=</security_code>
  <cardholder_registered>yes</cardholder_registered>
  <eci>06</eci>
  <transactionID>TbwiW8VLThG3TjDuI7KS5wAJCAI=</transactionID>
</Secure>
```

D.5.2. XML Example Requests

Example Request XML for 3-D Secure with a 3rd party MPI, for a registered card

```
<?xml version="1.0" encoding="UTF-8"?>
 <Request>
  <Authentication>
   <client>99000001</client>
   <password>******</password>
  </Authentication>
  <Transaction>
   <TxnDetails>
    <merchantreference>123456</merchantreference>
    <amount currency='GBP'>10.00</amount>
    <capturemethod>ecomm</capturemethod>
   </TxnDetails>
   <CardTxn>
    <Card>
     <pan>4444********1111</pan>
     <expirydate>12/09</expirydate>
    </Card>
    <method>pre</method>
    <Secure type='visa'>
      <security_code>BwABCEYiBQAAAAAGViIFAAA
          AAAA=</security_code>
      <cardholder_registered>yes</cardholder_registered>
      <eci>06</eci>
      <transactionID>TbwiW8VLThG3TjDuI7KS5wAJ
          CAI=</transactionID>
    </Secure>
   </CardTxn>
  </Transaction>
 </Request>
```

Example Request XML for 3-D Secure with a 3rd party MPI, for a registered card using the Pre-Registered Card service

```
<?xml version="1.0" encoding="UTF-8"?>
 <Request>
  <Authentication>
   <client>99000001</client>
   <password>******</password>
  </Authentication>
  <Transaction>
   <TxnDetails>
    <merchantreference>123456</merchantreference>
    <amount currency='GBP'>10.00</amount>
    <capturemethod>ecomm</capturemethod>
   </TxnDetails>
   <CardTxn>
    <card_details type=”preregistered”>
       4900200053281123</card_details>
    <method>pre</method>
    <Secure type='visa'>
      <security_code>BwABCEYiBQAAAAAGViIFAAA
          AAAA=</security_code>
      <cardholder_registered>yes</cardholder_registered>
      <eci>06</eci>
      <transactionID>TbwiW8VLThG3TjDuI7KS5wAJ
          CAI=</transactionID>
    </Secure>
   </CardTxn>
  </Transaction>
 </Request>
```

Example Request XML for 3-D Secure with a 3rd party MPI, for an un-registered card

```
<?xml version="1.0" encoding="UTF-8"?>
 <Request>
  <Authentication>
   <client>99000001</client>
   <password>******</password>
  </Authentication>
  <Transaction>
   <TxnDetails>
    <merchantreference>123456</merchantreference>
    <amount currency='GBP'>56.26</amount>
    <capturemethod>ecomm</capturemethod>
   </TxnDetails>
   <CardTxn>
    <Card>
     <pan>5374********0001</pan>
     <expirydate>01/08</expirydate>
    </Card>
    <method>pre</method>
    <Secure type='ucaf'>
      <cardholder_registered>no</cardholder_registered>
    </Secure>
   </CardTxn>
  </Transaction>
 </Request>
```

Example Request XML for a card scheme which is not supported for the 3-D Secure check

```
<?xml version="1.0" encoding="UTF-8"?>
 <Request>
  <Authentication>
   <client>99000001</client>
   <password>******</password>
  </Authentication>
  <Transaction>
   <TxnDetails>
    <merchantreference>45786906</merchantreference>
    <amount currency='GBP'>23.45</amount>
    <capturemethod>ecomm</capturemethod>
   </TxnDetails>
   <CardTxn>
    <Card>
     <pan>6759********9999</pan>
     <expirydate>12/07</expirydate>
     <startdate>12/04</startdate>
     <issuenumber>03</issuenumber>
    </Card>
    <method>auth</method>
   </CardTxn>
  </Transaction>
 </Request>
```

D.5.3. Schema Elements for Response

This service does not have any service-specific elements in the response. Please refer to the Bank Card Service (sectionB.1) and General Response Elements (section A.1.2) for details of the generic elements returned for credit and debit card transaction

D.6. Batched Fraud Screening

This service is utilised by sending a normal Credit and Debit Card Service Request with additional information. This section of documentation assumes the Credit and Debit Card Service has already been integrated and the reader is familiar with it. The Credit and Debit Card Service is described in section B.1

D.6.1. Schema Elements for Request

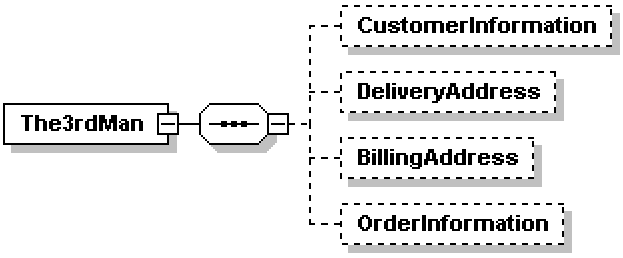
All of these fields described within this section are optional, unless otherwise stated.

Initial Transactions with Card details
The 3rd Man check is carried out for any `pre` and `auth` transactions while your account is subscribed to this service. To perform the check, additional information about the customer needs to be presented with the transaction.
The data is passed within the The3rdMan element in the schema:

- `The3rdMan`
    - `CustomerInformation` – additional details about the customer, section D.6.1.2
    - `DeliveryAddress` – details the delivery address for the order, section D.6.1.3
    - `OrderInformation` – details about the order, section D.6.1.4
        - `Products` - section D.6.1.6
            - `Product` – details about the individual products within the order, section D.6.1.7

D.6.1.1. The3rdMan

Element Name:	`The3rdMan`
Position:	`Request.Transaction.TxnDetails`



Elements of The3rdMan		
Element Name	**description**	**values / limitations**
CustomerInformation	See section D.6.1.2	
DeliveryAddress	See section D.6.1.3	
BillingAddress	See section D.6.1.4	
OrderInformation	See section D.6.1.5	

Example XML for The3rdMan complex elements

```
<The3rdMan>
  <CustomerInformation>…</CustomerInformation>
  <DeliveryAddress>…</DeliveryAddress>
  <BillingAddress>…</BillingAddress>
  <OrderInformation>…</OrderInformation>
</The3rdMan>

<The3rdMan>
  <CustomerInformation>…</CustomerInformation>
</The3rdMan>
```

D.6.1.2. CustomerInformation

This element contains details of the customer. T3M themselves are the authoritative source on what the contents of these fields should be.

Element Name:	`CustomerInformation`
Position:	`Request.Transaction.TxnDetails.The3rdMan`

Elements of `CustomerInformation`		
Element Name	**description**	**values / limitations**
`alt_telephone`	Alternative Telephone Number	Max 20 characters *International numbers*: include Country code, exclude access code.
`customer_dob`	Customer Date Of Birth	`YYYY-MM-DD` format
`customer_reference`	Customer reference	Max 50 characters
`delivery_forename`	The firstname of the person to which the order is being delivered	Max 50 characters
`delivery_phone_number`	The phone number of the person to which the order is being delivered	Max 20 characters
`delivery_surname`	The surname of the person to which the order is being delivered	Max 50 characters
`delivery_title`	The title of the person to which the order is being delivered	Max 10 characters
`driving_licence_number`	Driving License Number	Max 30 characters
`email`	Email Address	Max 50 characters
`first_purchase_date`	First Purchase Date By Customer	`YYYY-MM-DD` format
`forename`	Customer firstname	Max 50 characters
`introduced_by`	Customer Ref Of Introducing Customer	Max 50 characters
`ip_address`	IP Address	Must be in "dotted-quad" notation
`previous_purchases`	Previous Purchases	See below
`order_number`	Order Number	Max 60 characters
`sales_channel`	Type of sale	`1` – mail order `2` – telephone order `3` – internet `4` - other
`surname`	Customer Surname	Max 50 characters
`telephone`	Telephone Number	A maximum of 20 characters. *International numbers*: include Country code, exclude access code.
`time_zone`	Customer's time zone	Customer's time zone. Should be of the form "GMT", "GMT+1", "CET", etc. Websites such as http://wwp.greenwichmeantime.com/ list possible timezones.
`title`	Customer Title (Mr, Mrs etc)	Max 10 characters

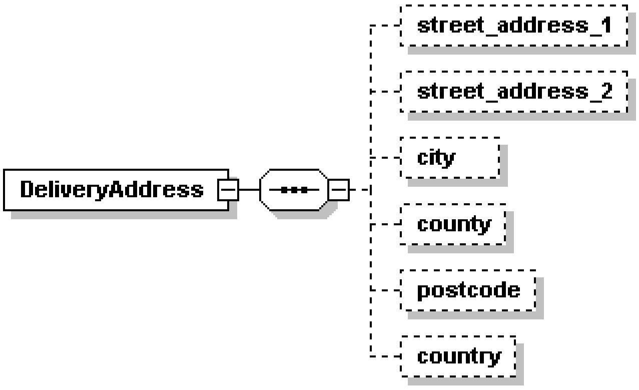
Attributes for Elements of `CustomerInformation`		
Attribute Name	**Attribute of element**	**Value/limitations**
`count`	`previous_purchases`	Number of previous purchases. Numeric
`amount`	`previous_purchases`	Total value of previous purchases. Max 15 numeric including decimal point

Example XML for CustomerInformation complex elements

```
<CustomerInformation>
  <customer_reference>CUSTREF000001</customer_reference>
  <title>Mr</title>
  <forename>John</forename>
  <surname>Smith</surname>
  <telephone>0131 123 1234</telephone>
  <alt_telephone>0131 123 1234</alt_telephone>
  <email>jsmith@devnull.co.uk</email>
  <ip_address>192.168.0.1</ip_address>
  <customer_dob>1980-03-12</customer_dob>
  <first_purchase_date>2004-02-21</first_purchase_date>
  <previous_purchases count="5" value="58.94" />
  <introduced_by>CUSTREF0004444</introduced_by>
  <driving_license_number>SMITH0987654321</driving_license_number>
  <time_zone>GMT</time_zone>
</CustomerInformation>

<CustomerInformation>
  <forename>John</forename>
  <surname>Smith<surname>
  <telephone></telephone>
  <email></email>
  <delivery_forename>Alice</forename>
  <delivery_surname>Smith<surname>
  <delivery_phone_number></telephone>
</CustomerInformation>
```

D.6.1.3. DeliveryAddress

Element Name: `DeliveryAddress`
Position: `Request.Transaction.TxnDetails.The3rdMan`



Elements of `DeliveryDetails`		
Element Name	**description**	**values / limitations**
`street_address_1`	First address line	Max 100 characters
`street_address_2`	Second address line	Max 100 characters
`city`	City	Max 20 characters
`county`	County	Max 30 characters
`postcode`	Postcode	Max 9 characters (or 10 for Amex transactions)
`country`	Country Code	Use the numeric country codes

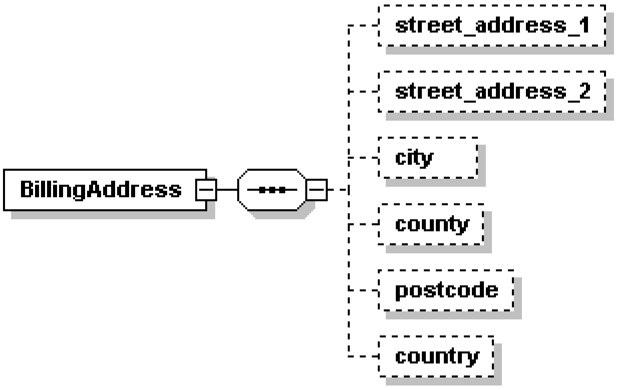
Example XML for DeliveryAddress complex elements

```
<DeliveryAddress>
  <street_address_1>10 Stratford Road</street_address_1>
  <street_address_2>Windsor</street_address_2>
  <city>London</city>
  <country>826</country>
  <postcode>AB1 2CD</postcode>
</DeliveryAddress>
```

D.6.1.4. BillingAddress

Element Name:	`BillingAddress`
Position:	`Request.Transaction.TxnDetails.The3rdMan`

The child elements of `BillingAddress` are the same as `DeliveryAddress`.



Example XML for BillingAddress complex elements

```
<BillingAddress>
  <street_address_1>Mulberry House</street_address_1>
  <street_address_2>15 Acacia Place</street_address_2>
  <city>Edinburgh</city>
  <county>Lothian</county>
  <country>826</country>
  <postcode>EH6 7EZ</postcode>
</BillingAddress>

<BillingAddress>
  <street_address_1>10 Banana Gardens</street_address_1>
  <city>Chepstowe</city>
  <country>826</country>
  <postcode>CH11 4XX</postcode>
</BillingAddress>
```

D.6.1.5. OrderInformation

Element Name: OrderInformation
Position: Request.Transaction.TxnDetails.The3rdMan

Elements of **OrderInformation**		
Element Name	**description**	**values / limitations**
Products	See section D.6.1.6	
brand	The brand, if the sales channel supports multiple brands	Max 30 characters
consumer_opt_in	Customer opt-in flag	0 – no 1 - yes
destination_location	Used for travel and ticketing. Route codes may be used and defined as required with T3M	Max 50 characters
distribution_channel	The distribution channel	Max 30 characters
event_date	The date the event takes place, eg date of travel, or date of show	YYYY-MM-DD format
event_location	The event location	Max 50 characters
gift_message	Message entered on gift card	Max 100 characters
installation_request	Whether goods are to be supplied together with physical installation	Yes No
loyalty_card_number	The loyalty card number, if held by customer	Max 50 characters
operator_id		Max 30 characters
route_via_location	Routing information	Max 50 characters

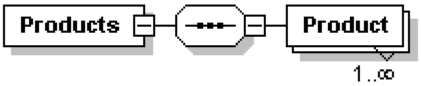
Example XML for OrderInformation complex elements

```
<OrderInformation>
  <distribution_channel>First Class Post</distribution_channel>
  <Products count="6">…</Products>
</OrderInformation>
```

D.6.1.6. Products

This element enables details of the items within the order to be passed.

Element Name:	`Products`
Position:	`Request.Transaction.TxnDetails.The3rdMan.OrderInformation`



Elements of `Products`		
Element Name	**description**	**values / limitations**
`Product`	See section D.6.1.7	One element for each distinct item ordered.

Attributes of `Products`		
Attribute name	**description**	**values / limitations**
`count`	Total number of items in the order	Must match the total of the items specified in the `Product.count` fields.

Example XML for Products complex elements

```
<Products count="1">
  <Product>…</Product>
</Products>

<Products count="2">
  <Product>…</Product>
  <Product>…</Product>
</Products>
```

D.6.1.7. Product

Enables specific information about each distinct item within the order to be presented. If the Products element is specified, at least one Product element must be present.

Element Name:	`Product`
Position:	`Request.Transaction.TxnDetails.The3rdMan.OrderInformation.Products`

Elements of `The3rdMan`		
Element Name	**description**	**values / limitations**
`code`	The product code for the item	Max 30 characters
`price`	The unit cost of a single item	Numeric with decimal point
`prod_category`	The product category	Max 50 characters
`prod_description`	The product description	Max 50 characters
`prod_risk`	The level of risk associated with the product	Max 50 characters
`prod_type`	The type of product	Max 50 characters
`quantity`	The number of items of this product ordered	Non-negative integer

Example XML for Product complex element

```
<Product>
  <code>32231</code>
  <quantity>2</quantity>
  <price>22.99</price>
</Product>
```

D.6.2. Schema Elements for Response

As the results of the service will be returned to you directly by The 3rd Man, there are no additional XML Response elements for this service.

Please refer to section A.1.2 for details of the general response elements.

D.6.3. Example XML Responses

Please refer to the Developers Guide for examples of Credit and Debit Card response

A complete list of Response Codes for this service is available on the website. The Support Centre also contains extensive examples for most error codes – including XML Responses - and also contains suggestions to prevent them occurring.

- Support Centre
- Developers Area

E. BACS Service

E.1. Direct Debit

Additional non-technical information about this Service is available on the website:
http://www.datacash.com/services/BACS/overview.shtml

E.1.1. Schema Elements for Request

In this section the required fields for each stage in the Direct Debit process will be presented, along with example XML for those fields. The XML is presented in italics for those fields that are not required for all situations. In the XML examples where a field has been highlighted fields, this indicates a situation in which other values can be presented in its place – for example `setup` in place of `presetup`.

Optional fields for each transaction type are indicated with an *O*, required fields with an *R*.

Please refer to the website for definitions of the transaction types and examples of when you may wish to implement them.

The Initial Setup –Setup and Presetup
The `setup`and `presetup` methods both require the same fields to be populated.
This information is passed in two distinct places in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `DirectDebitTxn` - contains all the information about the customer and the `method`, section E.1.1.1
    - `TxnDetails` - contains the reference number, section E.1.1.2

Updating Setups – Confirm and Revoke
If you are using the two stage model, you will need to `confirm` the `presetup` in order to activate it. Once all payments on a DDI have been completed, the DDI can be cancelled by submitting a `revoke` request.

Both of these transaction types update the existing DDI and the information is passed in two distinct schema paths:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `HistoricTxn` - the `datacash_reference` of the original `setup` or `presetup` plus the method, section E.1.1.3
    - `TxnDetails` - contains the reference number, section E.1.1.2

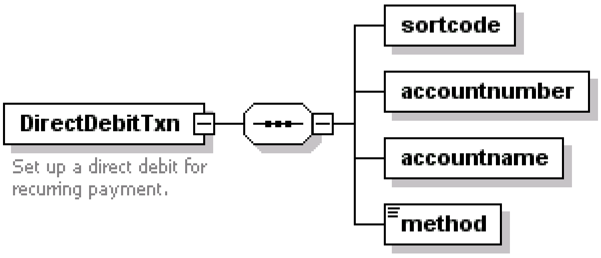
Taking Payments
Once a DDI is active, a `drawdown` can then be processed against it. A `drawdown` may also be cancelled, if required, using a `cancel`.

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `HistoricTxn` - the `datacash_reference` of the original transation, plus the `method` and for `drawdowns` (optionally) the due date and transaction code, section E.1.1.3
    - `TxnDetails` – for `drawdowns` only, contains the reference number and the amount, section E.1.1.2

E.1.1.1. DirectDebitTxn

The DirectDebitTxn element contains the details of the customer and their bank account. It also contains the method which allows you to choose the one stage setup or two stage presetup models. Each element is required for both models.

Element Name: DirectDebitTxn
Position(s) Request.Transaction



Elements of DirectDebitTxn			
Element Name	**description**	**values / limitations**	
sortcode	The sort code of the customer's bank	Six digits. Can contain additional hyphens and spaces	R
accountnumber	The customer's account number	Eight digits. Can contain additional hyphens and spaces	R
accountname	The name of the account holder	A maximum of 18 alphanumerics	R
method	determines the processing model to be used	setup - for one stage processing presetup - for two stage processing	R

In addition to these elements, there are also two attributes that may be populated - these are both optional.

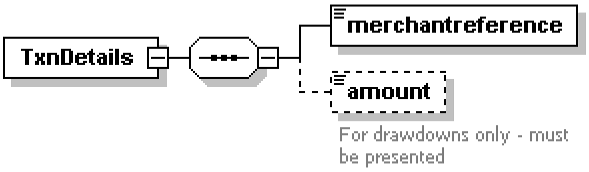
Optional Attributes of DirectDebitTxn			
Attribute Name	**description**	**values / limitations**	
active	enables an existing setup to be transfered to the DataCash system	true	O
type	enables a non_AUDDIS DDI to be converted into an AUDDIS DDI	conversion	O

Example XML for DirectDebitTxn elements

```
<DirectDebitTxn type="conversion">
  <sortcode>123456</sortcode>
  <accountnumber>12345678</accountnumber>
  <accountname>Mr A. N. Other</accountname>
  <method>setup</method>
 </DirectDebitTxn>

<DirectDebitTxn>
  <sortcode>123457</sortcode>
  <accountnumber>12345678</accountnumber>
  <accountname>Mr A. N. Other</accountname>
  <method>presetup</method>
 </DirectDebitTxn>
```

E.1.1.2. TxnDetails

Element Name:	TxnDetails
Position(s)	Request.Transaction



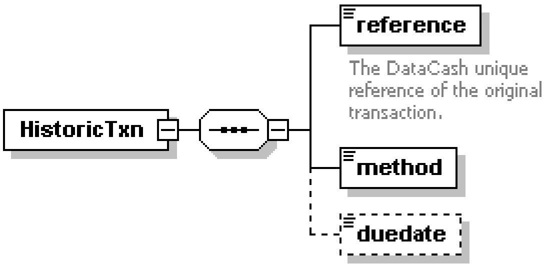
Elements of TxnDetails							
Element Name	**description**	**values / limitations**	**confirm**	**drawdown**	**revoke**	**setup & presetup**	**cancel**
merchantreference	The reference number of the DDI mandate	Must be of the format agreed with your Sponsoring Bank. Character limitations apply	R	R	R	R	-
amount	The drawdown value		-	R	-	-	-

Example XML Request for TxnDetails elements

```
<TxnDetails>
  <merchantreference>123402</merchantreference>
  <amount>30.00</amount>
</TxnDetails>
```

E.1.1.3. HistoricTxn

Element Name:	HistoricTxn
Position(s)	Request.Transaction



Elements of HistoricTxn						
Element Name	description	values / limitations	confirm	revoke	drawdown	cancel
method	Allows the type of HistoricTxn to be set	confirm revoke drawdown	R	R	R	R
reference	The datacash_reference of the original transaction		R	R	R	R
duedate	Allows a future date for the drawdown to be set if required. If this field is not presented, the next available date will be used	yyyymmdd format. Must be at least 3 working days in the future	-	-	O	-

The DataCash service automatically calculates the BACS Transaction Code for all drawdown transactions. If the drawdown is the first made against a new DDI setup, the transaction code is set to 01, otherwise the transaction code is set to 17.

You may need to set the transaction code yourself however, depending upon whether the drawdown is a representation of a previously rejected drawdown, or if this is the last drawdown to be made against a DDI setup.

Optional Attribute of duedate		
Attribute Name	description	values / limitations
tran_code	The Transaction Code for the drawdown	01 – the first drawdown 17 18 - a drawdown that is being re-presented 19 - the final drawdown. The drawdown will be processed and then the mandate will be cancelled

Example XML Request for HistoricTxn elements

```
<HistoricTxn>
  <method>confirm</method>
  <reference>12345679</reference>
</HistoricTxn>

<HistoricTxn>
  <method>drawdown</method>
  <reference>12345678</reference>
</HistoricTxn>

<HistoricTxn>
  <method>drawdown</method>
  <reference>12345678</reference>
  <duedate tran_code="18">20070101</duedate>
</HistoricTxn>

<HistoricTxn>
  <method>drawdown</method>
  <reference>12345678</reference>
  <duedate>20070131</duedate>
</HistoricTxn>

<HistoricTxn>
  <method>cancel</method>
  <reference>42912928</reference>
</HistoricTxn>
```

E.1.2. XML Example Requests

E.1.2.1. The Initial DataCash Setup

Example XML Request for presetup

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <password>******</password>
    <client>99000000</client>
  </Authentication>
  <Transaction>
    <DirectDebitTxn>
      <sortcode>123456</sortcode>
      <accountnumber>12345678</accountnumber>
      <accountname>Mr A. N. Other</accountname>
      <method>presetup</method>
    </DirectDebitTxn>
    <TxnDetails>
      <merchantreference>123402</merchantreference>
    </TxnDetails>
  </Transaction>
</Request>
```

Example XML Request for setup transfering an existing mandate to DataCash

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <password>*****</password>
    <client>99000000</client>
  </Authentication>
  <Transaction>
    <DirectDebitTxn active="true">
      <sortcode>123456</sortcode>
      <accountnumber>12345678</accountnumber>
      <accountname>Mr A. N. Other</accountname>
      <method>setup</method>
    </DirectDebitTxn>
    <TxnDetails>
      <merchantreference>123402</merchantreference>
    </TxnDetails>
  </Transaction>
</Request>
```

Example XML Request for a setupThis will convert an existing non-AUDDIS mandate to AUDDIS

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <password>*****</password>
    <client>99000000</client>
  </Authentication>
  <Transaction>
    <DirectDebitTxn type='conversion'>
      <sortcode>123456</sortcode>
      <accountnumber>12345678</accountnumber>
      <accountname>Mr A. N. Other</accountname>
      <method>setup</method>
    </DirectDebitTxn>
    <TxnDetails>
      <merchantreference>123402</merchantreference>
    </TxnDetails>
  </Transaction>
</Request>
```

E.1.2.2. Updating Setups

Example XML Request for confirm

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <password>******</password>
    <client>99000000</client>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>confirm</method>
      <reference>12345679</reference>
    </HistoricTxn>
    <TxnDetails>
      <merchantreference>123401</merchantreference>
    </TxnDetails>
  </Transaction>
</Request>
```

If the `method` was changed, the same XML could be used for a `revoke`

E.1.2.3. Taking Payments

Example XML Request for a drawdown

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <password>*****</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>drawdown</method>
      <reference>12345678</reference>>
    </HistoricTxn>
    <TxnDetails>
      <merchantreference>123401</merchantreference>
      <amount>10.00</amount>
    </TxnDetails>
  </Transaction>
</Request>
```

Example XML Request for a drawdown with a duedate

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <password>*****</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>drawdown</method>
      <reference>12345678</reference>
      <duedate>20070922</duedate>
    </HistoricTxn>
    <TxnDetails>
      <merchantreference>123401</merchantreference>
      <amount>10.00</amount>
    </TxnDetails>
  </Transaction>
</Request>
```

Example XML Request for a drawdown with a tran_code

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <password>*****</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>drawdown</method>
      <reference>12345678</reference>
      <duedate tran_code="18">20070815</duedate>
    </HistoricTxn>
    <TxnDetails>
      <merchantreference>123401</merchantreference>
      <amount>10.00</amount>
    </TxnDetails>
  </Transaction>
</Request>
```

E.1.2.4. Cancelling Payments

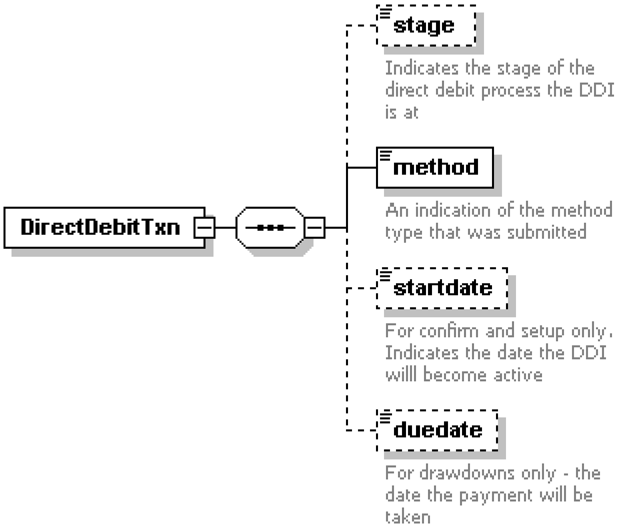
Example XML Request for a drawdown cancellation

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <password>*****</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>cancel</method>
      <reference>123412345</reference>
    </HistoricTxn>
  </Transaction>
</Request>
```

E.1.3. XML Example Responses

In addition to the standard fields, each Response will contain a DirectDebitTxn element

Element Name:	DirectDebitTxn
Position(s)	Response


stage
Indicates the stage of the direct debit process the DDI is at
method
An indication of the method type that was submitted
DirectDebitTxn
startdate
For confirm and setup only. Indicates the date the DDI willl become active
duedate
For drawdowns only - the date the payment will be taken

Not all elements will be returned for all transaction types, for successful Responses these are outlined below.
Failures will generally contain less of these elements - additional information about the reason for failure will be returned in the reason and information elements. Examples of failures are available in the Support Centre.

Elements of DirectDebitTxn Response									
Element Name	**description**	**Values / Limitations**	**setup**	**presetup**	**confirm**	**revoke**	**drawdown**	**cancel**
stage	The current stage of the mandate	See Support Centre	yes	yes	yes	yes	n/a	n/a
method	The method sent through with the Request		yes	yes	yes	yes	yes	yes
startdate	Indicates the first date that a drawdown can be submitted.	Will be five working days hence	yes	n/a	yes	n/a	n/a	n/a
duedate	The date the drawdown will be taken from the account.		n/a	n/a	n/a	n/a	yes	n/a

Once a successful Response for a setup or presetup has been received, the datacash_reference must be stored, as this must be presented with any future transactions refering to that mandate.

E.1.3.1. The Initial DataCash Setup

Example XML Response for a successful setup

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>12345672</datacash_reference>
  <DirectDebitTxn>
    <method>setup</method>
    <stage>active DDI</stage>
    <startdate>20020313</startdate>
  </DirectDebitTxn>
  <merchantreference>123402</merchantreference>
  <mode>TEST</mode>
  <reason>ACKNOWLEDGEMENT</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Response for a successful presetup

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>12345672</datacash_reference>
  <DirectDebitTxn>
    <method>presetup</method>
    <stage>confirm required</stage>
  </DirectDebitTxn>
  <merchantreference>123402</merchantreference>
  <mode>ACCREDITATION</mode>
  <reason>ACKNOWLEDGEMENT</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Response for a successful setup of an existing DDI

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>12345672</datacash_reference>
  <DirectDebitTxn>
    <method>setup</method>
    <stage>existing</stage>
    <startdate>20020306</startdate>
  </DirectDebitTxn>
  <merchantreference>123402</merchantreference>
  <mode>LIVE</mode>
  <reason>ACKNOWLEDGEMENT</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

E.1.3.2. Updating Setups

Example XML Response for a successful confirm

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <DirectDebitTxn>
    <method>confirm</method>
    <stage>active DDI</stage>
    <startdate>20020315</startdate>
  </DirectDebitTxn>
  <merchantreference>123401</merchantreference>
  <reason>ACKNOWLEDGEMENT</reason>
  <status>1</status>
</Response>
```

Example XML Response for a successful revoke

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <DirectDebitTxn>
    <method>revoke</method>
    <stage>revoked</stage>
  </DirectDebitTxn>
  <merchantreference>123402</merchantreference>
  <reason>ACKNOWLEDGEMENT</reason>
  <status>1</status>
</Response>
```

E.1.3.3. Taking Payments

Example XML Response for a successful drawdown

```
<Response>
  <datacash_reference>12345672</datacash_reference>
  <DirectDebitTxn>
    <duedate tran_code='01'>20070130</duedate>
    <method>drawdown</method>
  </DirectDebitTxn>
  <merchantreference>123402</merchantreference>
  <mode>TEST</mode>
  <reason>OK</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Response for a successful drawdown

```
<Response>
  <datacash_reference>12345672</datacash_reference>
  <DirectDebitTxn>
    <duedate tran_code='19'>20060924</duedate>
    <method>drawdown</method>
  </DirectDebitTxn>
  <merchantreference>123402</merchantreference>
  <mode>TEST</mode>
  <reason>OK</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

E.1.3.4. Cancelling Payments

Example XML Response for a successful drawdown cancellation

```
<Response>
  <datacash_reference>12345671</datacash_reference>
  <DirectDebitTxn>
    <method>cancel</method>
  </DirectDebitTxn>
  <merchantreference>12345671</merchantreference>
  <mode>LIVE</mode>
  <reason>CANCELLED OK</reason>
  <status>1</status>
  <time>...</time>
</Response>
```

E.1.4. Notifications

This section is only applicable if you have completed the relevant paperwork at your Sponsoring Bank, enabling DataCash to collect notifications of any rejections, cancellations and changes of your setups. These changes will be automatically reflected in the DataCash System.

To allow your system to also be kept up to date and prevent drawdown submission on cancelled mandates, these will be mailed to you in a standard format attachment.

The following information will be included in the attachment:

- `merchant_ref` - your setup reference
- `uniq_ref` - the `datacash_reference` of the mandate
- `reason_code` - the ADDACS / AUDDIS reason code
- `reason_code_text` - further information about the `reason_code`
- `effective_date` - the date on which the notification takes affect

Full details about the reason codes are available in the Originators Guide and Rules to the Direct Debit Scheme which is available from your bank. These are also detailed in the Support Centre.

Email notifying of nine modifications to Direct Debits

```
Direct Debit Advices: BACS ID 556622

Please find attached your CSV file
containing Direct Debit advices.

Total updated:    9
Total cancelled:  6
Total changed:    2
Total reinstated: 1
```

Example attachment with six cancellations, two changes and one reinstatement - the changes are reason_codes C and E

```
merchant_ref,uniq_ref,reason_code,reason_code_text,effective_date
"ABC001","99990001","1","Instruction Cancelled by Payer","2004-01-30"
"ABCDE1","99990000","0","Instruction Cancelled - Refer to
     Payer","2004-01-30"
"ABC053","99992222","2","Payer deceased","2004-02-01"
"ABC296","89898989","1","Instruction Cancelled by Payer","2004-02-03"
"ABC00A","99995555","1","Instruction Cancelled by Payer","2004-01-31"
"ABC009","99991111","C","Account transferred to a different
     account/branch of Bank/Building Society","2004-01-30"
"ABCDE9","99993333","G","Bank will not accept Direct Debit on
     Account","2004-02-22"
"ABC921","99966633","E","Instruction Amended","2004-01-31"
"ABC828","99779977","R","Instruction Reinstated","2004-01-30"
```

E.2. Direct Debit Continuous Authority

This Service allows drawdowns to be automatically created against a DDI.

A technical introduction to this Service is available on the website:
http://www.datacash.com/services/BACS/ddca.shtml.

E.2.1. Schema Elements for Request

In this section the required fields for Direct Debit refunds will be presented, along with example XML for those fields. The XML is presented in italics for those fields that are not required for all situations. In the XML examples where a field has been highlighted fields, this indicates a situation in which other values can be presented in its place – for example `weekly` in place of `monthly`.

Optional fields for each transaction type are indicated with an O, required fields with an R.

Setting up a Direct Debit Continuous Authority Account
The information required to set up a Direct Debit Continuous Authority Account is passed in these distinct places in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `ContAuthTxn` – details of the account to be set up and it's regular payments, section E.2.1.1
      - `FirstPayment` – optional first payment date & amount, section E.2.1.2
      - `LastPayment` – optional last payment date & amount, section E.2.1.3
    - `TxnDetails` – your reference number for the account, section E.2.1.4
    - `HistoricTxn` - details of the DDI to create an account for, section E.2.1.5

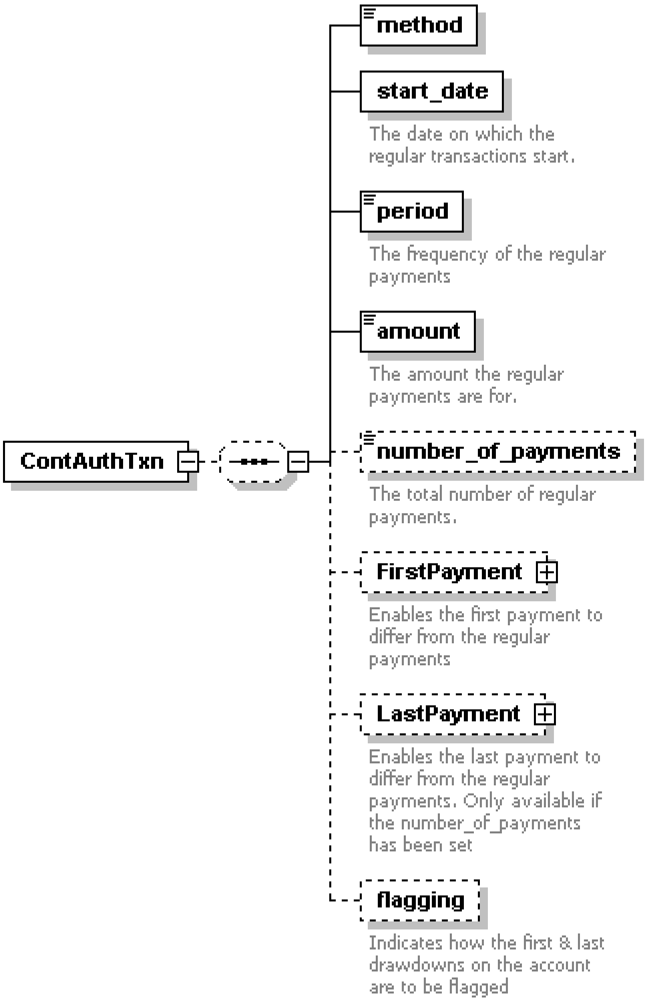
Cancelling a Continuous Authority Account
The Direct Debit Continuous Authority Account can also be cancelled by using the `cancel` transaction type. This transaction type requires information passed in one distinct place in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `HistoricTxn` - the `datacash_reference` of the account, plus the `method`, section E.2.1.5

E.2.1.1. ContAuthTxn

The elements within this parent may only be presented for `drawdown` transactions. They are excluded for `cancel` transactions.

Element Name:	`ContAuthTxn`
Position(s)	`Request.Transaction`


ContAuthTxn
method
start_date
The date on which the regular transactions start.
period
The frequency of the regular payments
amount
The amount the regular payments are for.
number_of_payments
The total number of regular payments.
FirstPayment
Enables the first payment to differ from the regular payments
LastPayment
Enables the last payment to differ from the regular payments. Only available if the number_of_payments has been set
flagging
Indicates how the first & last drawdowns on the account are to be flagged

Elements of ContAuthTxn			
Element Name	**description**	**values / limitations**	
`start_date`	The date on which the first regular payment will be taken. Subsequent payments will be based on this date.	Must be more than three days after the transaction is sent. Date in the format `dd/mm/yyyy`	R
`number_of_payments`	Number of regular payments to be collected.	Payments will be taken until the Continuous Authority account is cancelled if a value is not specified	O
`FirstPayment`	Details of any initial payment to be taken. This is in addition to the regular payments. See section E.2.1.2		O
`LastPayment`	Details of any final payment to be taken. This is in addition to the regular payments. See section E.2.1.3		O

`period`	The frequency of payments	`weekly`
		`monthly`
		`quarterly`
		`annually`
			R

`method`	The transaction method	`drawdown`	R
`amount`	The value of the drawdown	Must be specified to two decimal places. GBP only	R
`flagging`	This element has two attributes. If the element is supplied, both attributes must be supplied. See table below for more information		O

Attributes of flagging			
Element Name	**description**	**values / limitations**	
`first`	Indicates whether the first payment made using the Cont Auth account is to be flagged as the first drawdown	`yes` `no`	O
`last`	Indicates whether the last payment made using the Cont Auth account is to be flagged as the last drawdown	`yes` `no`	O

For further information about drawdown flagging, please refer to section E.1.1.3

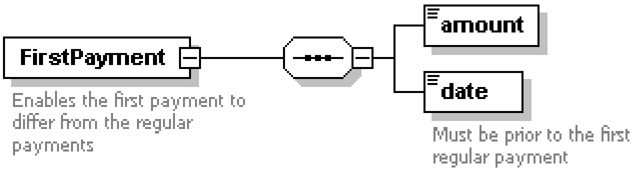
Example XML for ContAuthTxn elements

```
<ContAuthTxn>
  <method>drawdown</method>
  <start_date>26/02/2006</start_date>
  <period>weekly</period>
  <amount>35.00</amount>
</ContAuthTxn>

<ContAuthTxn>
  <method>drawdown</method>
  <start_date>01/04/2005</start_date>
  <period>annually</period>
  <amount>100.00</amount>
  <number_of_payments>3</number_of_payments>
  <flagging first="yes" last="no"/>
  <FirstPayment>…</FirstPayment>
</ContAuthTxn>
```

E.2.1.2. FirstPayment

The elements within this parent may only be presented for `drawdown` transactions, in which case both child elements must be completed.
This element is excluded for `cancel` transactions.

Element Name: `FirstPayment`
Position(s) `Request.Transaction.ContAuthTxn`


FirstPayment
Enables the first payment to differ from the regular payments
amount
date
Must be prior to the first regular payment

Elements of FirstPayment		
Element Name	**description**	**values / limitations**
`date`	The date the first payment is to be taken	Must be in `dd/mm/yyyy` format. Must be at least three days from the current date. It must also be at least one day ahead of the `start_date`
`amount`	The value of the first payment	If a currency is supplied, it must be GBP

Example XML for FirstPayment element

```
<FirstPayment>
  <amount>50.00</amount>
  <date>25/02/2006</date>
</FirstPayment>
```

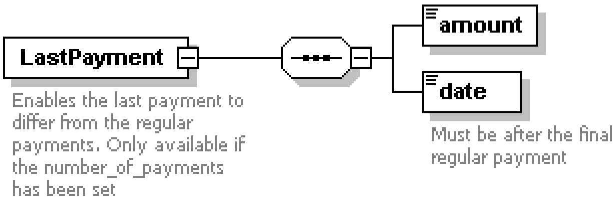
E.2.1.3. LastPayment

The elements within this parent may only be presented for `drawdown` transactions, in which case both child elements must be completed.

This element is excluded for `cancel` transactions.

The `LastPayment` element can only be used if a set number of drawdowns have been specified

Element Name:	`LastPayment`
Position(s)	`Request.Transaction.ContAuthTxn`


LastPayment
amount
date
Enables the last payment to differ from the regular payments. Only available if the number_of_payments has been set
Must be after the final regular payment

Elements of LastPayment		
Element Name	**description**	**values / limitations**
`date`	The date the last payment is to be taken	Must be in `dd/mm/yyyy` format. Must be a minimum of one date after the last regular payment
`amount`	The value of the last payment	If a currency is supplied, it must be GBP

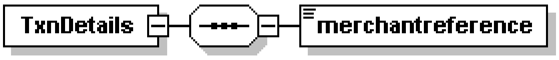
Example XML for LastPayment

```
<LastPayment>
  <amount>50.00</amount>
  <date>25/03/2008</date>
</LastPayment>
```

E.2.1.4. TxnDetails

The elements within this parent may only be presented for `drawdown` transactions. They are excluded for `cancel` transactions.

Element Name:	`TxnDetails`
Position(s)	`Request.Transaction`



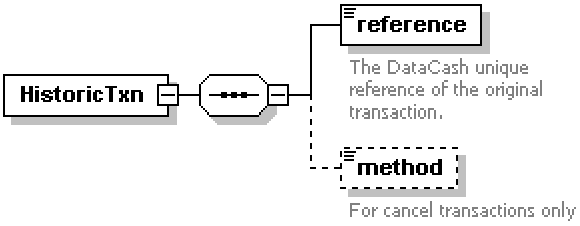
Elements of TxnDetails		
Element Name	**description**	**values / limitations**
`merchantreference`	A unique reference number to identify the individual Direct Debit Continuous Authority account	Between 6 and 26 alphanumeric characters

Example XML for TxnDetails elements

```
<TxnDetails>
  <merchantreference>123402</merchantreference>
</TxnDetails>
```

E.2.1.5. HistoricTxn

Element Name:	`HistoricTxn`
Position(s)	`Request.Transaction`


HistoricTxn
reference
The DataCash unique reference of the original transaction.
method
For cancel transactions only

Elements of HistoricTxn				
Element Name	**description**	**values / limitations**	**drawdown**	**cancel**
`method`	cancellations only	`cancel`	-	R
`reference`	The `datacash_reference` of the original transaction		R	R

Example XML element for cancel

```
<HistoricTxn>
   <method>cancel</method>
   <reference>3000123412345678</reference>
</HistoricTxn>
```

Example XML element for drawdown

```
<HistoricTxn>
   <reference>12345678</reference>
</HistoricTxn>
```

E.2.2. XML Example Requests

E.2.2.1. Payments

Example XML Request for a five weekly payments of 10.00

```
<Request>
   <Authentication>
     <client>21859999</client>
     <password>*********</password>
   </Authentication>
   <Transaction>
     <ContAuthTxn>
       <method>drawdown</method>
       <start_date>10/02/2005</start_date>
       <period>weekly</period>
       <number_of_payments>5</number_of_payments>
       <amount>10.00</amount>
     </ContAuthTxn>
     <TxnDetails>
       <merchantreference>12345678</merchantreference>
     </TxnDetails>
     <HistoricTxn>
       <reference>12345678</reference>
     </HistoricTxn>
   </Transaction>
 </Request>
```

Example XML Request for an unknown number of payments to be taken on the 31st of each month

```
<Request>
   <Authentication>
     <client>21859999</client>
     <password>********</password>
   </Authentication>
   <Transaction>
     <ContAuthTxn>
       <method>drawdown</method>
       <start_date>31/03/2005</start_date>
       <period>monthly</period>
       <amount>25.00</amount>
     </ContAuthTxn>
     <TxnDetails>
       <merchantreference>12345678</merchantreference>
     </TxnDetails>
     <HistoricTxn>
       <reference>12345678</reference>
     </HistoricTxn>
   </Transaction>
 </Request>
```

Example XML Request for a eight payments: six regular payments, and separate first and last payments
The first payment is a one off administration fee of £30, followed by 6 monthly payments of £25 and a final payment of £20. The first and last payments are to be flagged with "01" and "19" respectively, which will cancel the underlying DDI after the final payment.

```
<Request>
   <Authentication>
     <client>21859999</client>
     <password>********</password>
   </Authentication>
   <Transaction>
     <ContAuthTxn>
       <method>drawdown</method>
       <start_date>31/05/2005</start_date>
       <period>monthly</period>
       <amount>25.00</amount>
       <number_of_payments>6</number_of_payments>
       <FirstPayment>
         <amount>30.00</amount>
         <date>20/04/2005</date>
       </FirstPayment>
       <LastPayment>
         <amount>20.00</amount>
         <date>30/11/2005</date>
       </LastPayment>
       <flagging first="yes" last="yes"/>
     </ContAuthTxn>
     <TxnDetails>
       <merchantreference>12345678</merchantreference>
     </TxnDetails>
     <HistoricTxn>
       <reference>12345678</reference>
     </HistoricTxn>
   </Transaction>
 </Request>
```

E.2.2.2. Cancellations

Example XML Request for cancel

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>cancel</method>
      <reference>4900200000000001</reference>
    </HistoricTxn>
  </Transaction>
</Request>
```

Merchants using the Credit and Debit Card Services may note that this XML structure is the same as for cancelling a card payment – section B.1.4.2.

E.2.3. Schema Elements for Response

There are no Response elements specific to this service. The general Response elements are returned, these are discussed in section A.1.2.

E.2.4. XML Example Responses

Example XML Response for a successful account setup

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>9910670039375669</datacash_reference>
  <information>Direct Debit Continuous Authority account
     set up</information>
  <merchantreference>ABC00005</merchantreference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
</Response>
```

Further examples of unsuccessful transactions are available in the Support Centre.

E.2.5. Notifications

There are two types of notifications, one for batched drawdowns and one for account cancellations.

E.2.5.1. Batched Drawdowns

Each day a payment is taken from an account, you will receive an email summarising the payments taken that day, in a standard CSV format attachment.
The following information will be included in the attachment:

- `Account Reference` – the last 11 digits of the `datacash_reference` number of the account
- `Merchant Reference` – the DDI mandate number (`merchantreference`) against which the drawdown will be processed
- `DataCash Reference` – the `datacash_reference` of the drawdown

Email notifying of batched drawdowns

```
Drawdown Results for Mon 17-Oct-2005
Attachments: batched.csv

This email contains details of drawdowns which have been processed
on Live production site by DataCash on your behalf, for 2185****
Mon 17-Oct-2005.

If you have any questions please contact our support team.
```

Example attachment with two drawdowns

```
Account Reference,Merchant Reference,DataCash Reference
41740256,1823362AHC,4200200040467707
41756562,1825472RHP,4400200040373171
```

E.2.5.2. Account Cancellations

If a DDI is cancelled or revoked, the account will be automatically cancelled.
To allow your system to also be kept up to date, details of these will be mailed to you in a standard format CSV attachment.
The following information will be included in the attachment:

- `Account reference` – the last 11 digits of the `datacash_reference` number of the account
- `Merchant Reference` – the `merchantreference` of the account
- `Cancel DataCash Reference` – the `datacash_reference` number of the transaction which cancelled the account. If the account was automatically cancelled when a DDI `revoke` is submitted, this field will be the same as `DDI Setup Reference`
- `DDI Setup Reference` – the `datacash_reference` number of the DDI setup
- `DDI Merchant Reference` – your mandate number
- `DDI Stage` – the stage of the DDI

Email notifying of account cancellations

```
Account Cancellations for Mon 17-Oct-2005
Attachments: cancelled.csv

This email contains details of Direct Debit Recurring Transaction
accounts which have been cancelled on Live production site by
DataCash on your behalf, for 2185*** Mon 17-Oct-2005, where the
associated DDI setup is no longer active.
```

Example attachment with four cancellation notifications

```
Account Reference,Merchant Reference,Cancel DataCash Reference,DDI
Setup Reference,DDI Merchant Reference,DDI Stage
46453486,EFG987456,4600200040488736,4600200040488741,ABC1234562,revoked
47453828,EFG453454,4200200040493495,4000200040493496,ABC483,cancelled
at source
47455267,EFG456454,4300200040493522,4400200040493526,SJI45l,cancelled
at source
47511296,EFG15648,4100200041248347,4800200041260637,HUR45423,revoked
```

E.3. Direct Credit – Direct Debit Refunds

This Service allows funds to be returned to any DDI held within the DPG

A technical introduction to this service is available on the website:
http://www.datacash.com/services/BACS/refunds.shtml

E.3.1. Schema Elements for Request

In this section the required fields for Direct Debit refunds will be presented, along with example XML for those fields.

Refunding Direct Debits
The information required to process a direct debit refund - `ddrefund` - is passed in these distinct places in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `TxnDetails` - contains the `merchantreference` number and the `amount`, section E.3.1.1
    - `HistoricTxn` - the `datacash_reference` of the original mandate and the `method`, section E.3.1.2.

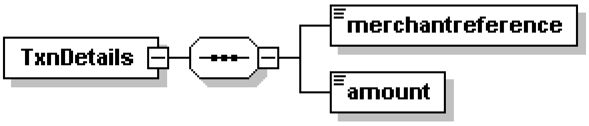
Cancelling Direct Debit Refunds
The information required to `cancel` a ddrefund is passed in these distinct places in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `HistoricTxn` - the `datacash_reference` of the original `ddrefund` and the `method`, section E.3.1.2.

E.3.1.1. TxnDetails

The elements within this parent are required for `ddrefund` transactions. They are excluded for `cancel` transactions.

Element Name:	`TxnDetails`
Position(s)	`Request.Transaction`



Elements of TxnDetails		
Element Name	**description**	**values / limitations**
`merchantreference`	The reference number of the DDI mandate	Between six and eighteen alphanumeric characters
`amount`	The value of the refund	

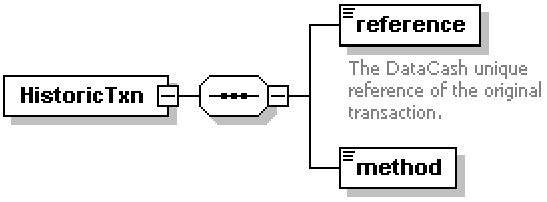
Example XML for TxnDetails elements

```
<TxnDetails>
      <merchantreference>123402</merchantreference>
      <amount>30.00</amount>
</TxnDetails>
```

E.3.1.2. HistoricTxn

The elements within this parent are required for both `cancel` and `ddrefund` transaction types.

Element Name:	`HistoricTxn`
Position(s)	`Request.Transaction`



Elements of HistoricTxn		
Element Name	**description**	**values / limitations**
`method`	Identifies the transaction type as a refund of a direct debit	`ddrefund` `cancel`
`reference`	The `datacash_reference` of the original transaction	

Example XML for HistoricTxn element for a ddrefund

```
<HistoricTxn>
   <method>ddrefund</method>
   <reference>12345678</reference>
</HistoricTxn>
```

Example XML for HistoricTxn element for a cancel

```
<HistoricTxn>
   <method>cancel</method>
   <reference>4000000012345678</reference>
</HistoricTxn>
```

E.3.2. XML Example Requests

Example XML Request for a ddrefund

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>*******</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>ddrefund</method>
      <reference>12345678</reference>
    </HistoricTxn>
    <TxnDetails>
      <merchantreference>ABCDEF456</merchantreference>
      <amount>10.00</amount>
    </TxnDetails>
  </Transaction>
</Request>
```

Example XML Request for a cancel

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>cancel</method>
      <reference>4000000059832490</reference>
    </HistoricTxn>
  </Transaction>
</Request>
```

E.3.3. Schema Elements for Response

There are no additional XML elements returned for specifically for this Service. The general XML elements in the Response are covered in section A.1.2.

E.3.4. XML Example Responses

Further examples of unsuccessful transactions are available in the Support Centre.

E.3.4.1. Refunds

Example XML Response for a successful ddrefund

```
<Response>
  <datacash_reference>4800200040644359</datacash_reference>
  <merchantreference>ABCDEF456</merchantreference>
  <mode>LIVE</mode>
  <reason>Accepted</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Response for an unsuccessful ddrefund

```
<Response>
  <datacash_reference>4000000059832491</datacash_reference>
  <information>Unable to locate matching DDI
      transaction</information>
  <merchantreference>123411</merchantreference>
  <mode>TEST</mode>
  <reason>Unable to locate matching DDI transaction</reason>
  <status>137</status>
  <time>…</time>
</Response>
```

E.3.4.2. Cancellations

Example XML Response for a successfully cancelled Direct Credit

```
<Response>
  <datacash_reference>39375274</datacash_reference>
  <merchantreference>4800200040644359</merchantreference>
  <reason>CANCELLED OK</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Response for an unsuccessful Direct Credit cancellation

```
<Response>
  <information>Direct Credit has already been
     cancelled</information>
  <merchantreference>4800200040644359</merchantreference>
  <mode>TEST</mode>
  <reason>Already Cancelled</reason>
  <status>128</status>
  <time>…</time>
</Response>
```

E.4. Direct Credit – Standard

This Service allows funds to be returned to a customer by placing them directly into their bank account

A technical introduction to this service is available on the website:
http://www.datacash.com/services/BACS/directcredit/standard.shtml

E.4.1. Schema Elements for Request

In this section the required fields for Standard Direct Credit will be presented, along with example XML for those fields. The XML is presented in italics for those fields that are not required for all situations.

Optional fields for each transaction type are indicated with an O, required fields with an R.

Performing Direct Credits
The information required to process a direct credit - `directcredit` - is passed in these distinct places in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `TxnDetails` - contains the `merchantreference` number and the `amount`, section E.4.1.2
    - `DirectCreditTxn` - the details of the customers bank account, plus the `method`, section E.4.1.1

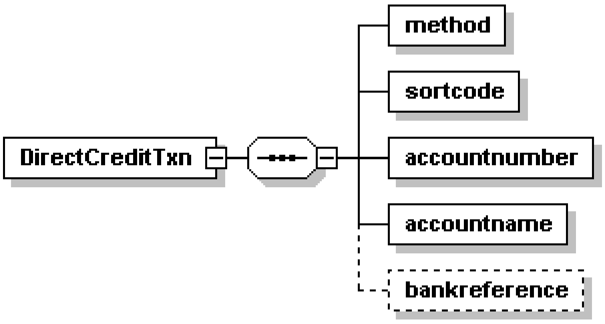
Cancelling Direct Debit Refunds
The information required to `cancel` a directcredit is passed in these distinct places in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `HistoricTxn` - the `datacash_reference` of the original `directcredit` and the `method`, section E.4.1.3

E.4.1.1. DirectCreditTxn

The elements within this parent may only be presented for `directcredit` transactions. They are excluded for `cancel` transactions.

Element Name:	`DirectCreditTxn`
Position(s)	`Request.Transaction`



Elements of DirectCreditTxn			
Element Name	**description**	**values / limitations**	**Required?**
`method`	Identifies the transaction type as a standard Direct Credit	`directcredit`	R
`sortcode`	The sort code of the account to be credited	Valid UK sortcode	R
`accountnumber`	The account number to be credited	Valid UK account number	R
`accountname`	The name of the account holder(s)	Up to 18 characters, alpha numerics and spaces only	R
`bankreference`		Up to 18 characters, alpha numerics and spaces only	O

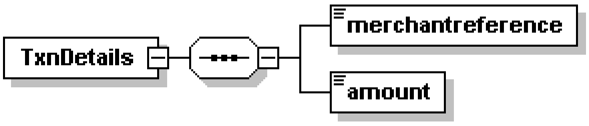
Example XML Element DirectCredit

```
<DirectCreditTxn>
  <sortcode>000001</sortcode>
  <accountnumber>12345609</accountnumber>
  <accountname>Julia McDonald</accountname>
  <method>directcredit</method>
</DirectCreditTxn>

<DirectCreditTxn>
  <sortcode>000222</sortcode>
  <accountnumber>12340000</accountnumber>
  <accountname>Michael Johnson</accountname>
  <bankreference>return of goods</bankreference>
  <method>directcredit</method>
</DirectCreditTxn>
```

E.4.1.2. TxnDetails

The elements within this parent are required for `directcredit` transactions, but excluded for `cancel` transactions.

Element Name: `TxnDetails`
Position(s) `Request.Transaction`



Elements of TxnDetails		
Element Name	**description**	**values / limitations**
`merchantreference`	A unique reference number for each transaction	Up to 30 characters
`amount`	The value of the refund	

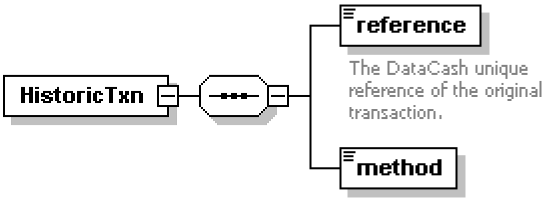
Example XML for TxnDetails elements

```
<TxnDetails>
  <merchantreference>123402</merchantreference>
  <amount>30.00</amount>
</TxnDetails>
```

E.4.1.3. HistoricTxn

The elements within this parent must be presented for `cancel` transactions. They are excluded for `directcredit` transactions.

Element Name:	`HistoricTxn`
Position(s)	`Request.Transaction.HistoricTxn`



Elements of HistoricTxn		
Element Name	**description**	**values / limitations**
`method`	Identifies the transaction type as a cancellation	`cancel`
`reference`	The `datacash_reference` of the original transaction	16 digits

Example XML Element HistoricTxn for a cancel

```
<HistoricTxn>
   <method>cancel</method>
   <reference>9999999912345678</reference>
</HistoricTxn>
```

E.4.2. XML Example Requests

Example XML Request

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>123402</merchantreference>
      <amount>1000.01</amount>
    </TxnDetails>
    <DirectCreditTxn>
      <method>directcredit</method>
      <sortcode>123456</sortcode>
      <accountnumber>12345678</accountnumber>
      <accountname>Greg Kane</accountname>
    </DirectCreditTxn>
  </Transaction>
</Request>
```

Example XML Request with optional bankreference

```
<Request>
  <Authentication>
    <password>******</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>ABDYC003</merchantreference>
      <amount>49.68</amount>
    </TxnDetails>
    <DirectCreditTxn>
      <method>directcredit</method>
      <sortcode>123456</sortcode>
      <accountnumber>12340000</accountnumber>
      <accountname>Jane Smith</accountname>
      <bankreference>Money for Petrol</bankreference>
    </DirectCreditTxn>
  </Transaction>
</Request>
```

Example XML Request for a cancel

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>cancel</method>
      <reference>4000000059832490</reference>
    </HistoricTxn>
  </Transaction>
</Request>
```

E.4.3. Schema Elements for Response

The XML Responses for this Service only contain those elements that are discussed in section A.1.2. There are no additional XML elements returned specifically for this Service

E.4.4. XML Example Responses

Example XML Response for a successful directcredit

```
<Response>
  <datacash_reference>4803000012345672</datacash_reference>
  <merchantreference>ABDYC003</merchantreference>
  <mode>TEST</mode>
  <reason>Accepted</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Response for an unsuccessful directcredit

```
<Response>
  <information>Invalid format for Account Name</information>
  <merchantreference>384086694791667</merchantreference>
  <mode>TEST</mode>
  <reason>Invalid format for Account Name</reason>
  <status>134</status>
  <time>…</time>
</Response>
```

Examples of directcredit cancellations are available in section E.3.4.2

Further examples of unsuccessful transactions are available in the Support Centre.

E.5. Direct Credit – Card Collection Accounts

This Service allows funds to be returned to a cardholder by using their card collection account.

A technical introduction to this service is available on the website:
http://www.datacash.com/services/BACS/directcredit/cardaccountpayment.shtml.

E.5.1. Schema Elements for Request

In this section the required fields for Direct Credit to Card Collection Accounts will be presented, along with example XML for those fields. The XML is presented in italics for those fields that are not required for all situations.

Optional fields for each transaction type are indicated with an *O*, required fields with an `R`.

Performing Card Account Payments
The information required to process a direct debit refund - `cardaccountpayment` - is passed in these distinct places in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `TxnDetails` - contains the `merchantreference` number and the `amount`, section E.5.1.2
    - `DirectCreditTxn` – details of the customers `card`, plus the `method`, section E.5.1.1

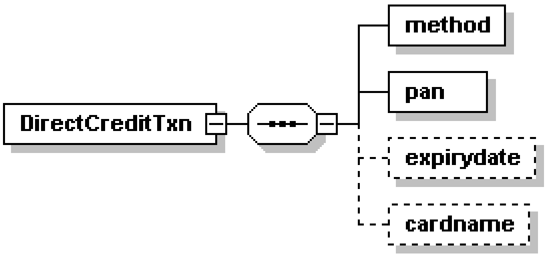
Cancelling Direct Debit Refunds
The information required to `cancel` a `cardaccountpayment` is passed in these distinct places in the schema:

- `Request`
  - `Authentication` – section A.1.1.1
  - `Transaction`
    - `HistoricTxn` - the `datacash_reference` of the `cardaccountpayment` and the `method`, section E.5.1.3

E.5.1.1. DirectCreditTxn

The elements within this parent may only be presented for `cardaccountpayment` transactions. They are excluded for `cancel` transactions.

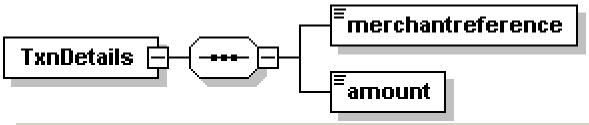
Element Name:	`DirectCreditTxn`
Position(s)	`Request.Transaction`



Elements of DirectCreditTxn			
Element Name	**description**	**values / limitations**	**Required?**
`method`	Identifies the transaction type as a refund to a card collection account	`cardaccountpayment`	R
`pan`	The card number	Card must be Visa or Mastercard branded and issued in the UK	R
`expirydate`	The card expiry date	MM/YY format	O
`cardname`	The card holder's name	Up to 18 characters. "`COLLECTION ACCOUNT`" will be automatically used if this element is not presented	O

E.5.1.2. TxnDetails

Element Name:	`TxnDetails`
Position(s)	`Request.Transaction`



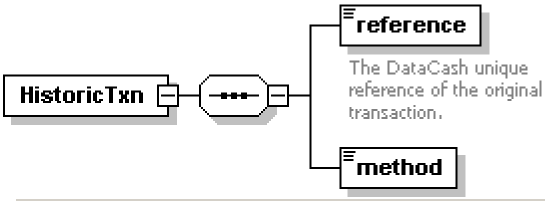
Elements of TxnDetails		
Element Name	**description**	**values / limitations**
`merchantreference`	A unique reference number for each transaction	Up to 30 characters
`amount`	The value of the refund	

Example XML for TxnDetails elements

```
<TxnDetails>
      <merchantreference>123402</merchantreference>
      <amount>30.00</amount>
 </TxnDetails>
```

E.5.1.3. HistoricTxn

Element Name:	`HistoricTxn`
Position(s)	`Request.Transaction.HistoricTxn`



Elements of HistoricTxn		
Element Name	**description**	**values / limitations**
`method`	Identifies the transaction type as a cancellation	`cancel`
`reference`	The `datacash_reference` of the original transaction	16 digits

Example XML Element HistoricTxn for a cancel

```
<HistoricTxn>
  <method>cancel</method>
  <reference>3009999912345678</reference>
</HistoricTxn>
```

E.5.2. XML Example Requests

Example XML Request for cardaccountpayment

```
<Request>
  <Authentication>
    <password>****</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>10397897589</merchantreference>
      <amount>59.00</amount>
    </TxnDetails>
    <DirectCreditTxn>
      <method>cardaccountpayment</method>
      <pan>5473********0007</pan>
    </DirectCreditTxn>
  </Transaction>
</Request>
```

Example XML Request for cardaccountpayment

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>****</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>B00123895</merchantreference>
      <amount>23.50</amount>
    </TxnDetails>
    <DirectCreditTxn>
      <method>cardaccountpayment</method>
      <pan>4444********0001</pan>
      <cardname>Wes Scantlin</cardname>
      <expirydate>03/09</expirydate>
    </DirectCreditTxn>
  </Transaction>
</Request>
```

Example XML Request for a cancel

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>******</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>cancel</method>
      <reference>4000000059832490</reference>
    </HistoricTxn>
  </Transaction>
</Request>
```

E.5.3. Schema Elements for Response

The XML Responses for this Service only contain those elements that are discussed in section A.1.2. There are no additional XML elements returned specifically for this Service

E.5.4. XML Example Responses

Example XML Response for a successful cardaccountpayment

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000200045345250</datacash_reference>
  <merchantreference>390726429629630</merchantreference>
  <mode>…</mode>
  <reason>Accepted</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Requests for unsuccessful cardaccountpayment

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4900200043479577</datacash_reference>
  <information>The card number does not pass the standard
               Luhn checksum test</information>
  <merchantreference>384084003240741</merchantreference>
  <mode>…</mode>
  <reason>Bad checksum</reason>
  <status>25</status>
  <time>…</time>
</Response>

<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>3600200044366081</datacash_reference>
  <information>No payment details for this card</information>
  <merchantreference>389377230555556</merchantreference>
  <mode>…</mode>
  <reason>Unknown Payment Details</reason>
  <status>126</status>
  <time>…</time>
</Response>
```

Examples of cancellations are available in section E.3.4.2

Further examples of unsuccessful transactions are available in the Support Centre.

F. Batch Input

The Batch Input Service allows many of the DataCash Services to be utilised via batch process, instead of submitting the transactions in real time. Further non-technical information about this service is available on the website: http://www.datacash.com/services/batch/overview.shtml

The transaction data may be presented in either a XML or a CSV file format. This file will be referred to in this document as the *Batch Request* XML (or CSV). This file must then be Zlib-compressed and Base64 encoded.

The compressed and encoded *Batch Request* file is then placed within the *Batch Submission* Request document, which is itself an XML document. The *Batch Submission* Request is then sent to DataCash and a *Batch Submission* Response is returned.

To collect the results of the batch, a *Batch Query* Request is sent to DataCash. A *Batch Query* Response will be returned. This will contain the results of the batch, which have been Zlib –compressed and Base64 encoded. These results will be referred to as the *Batch Response* XML (or CSV). The format of the *Batch Response* document will match that of the *Batch Request*.

Naturally, the data you need to supply will depend upon the particular service you are using. However there are certain aspects of Batch Input that are common to each service that it can be used with. The features common to all services will be covered first

- Available Services – section F.1
- Common Elements - each of these elements are independent of the service used, section F.2
  - File Formats Attributes, section F.2.1
  - *Batch Submission* elements - also independent of format, section F.2.2
  - *Batch Query* elements - also independent of format, section F.2.3
  - *Batch Request* elements - for both XML and CSV, section F.2.4
  - *Batch Response* elements - for both XML and CSV, section F.2.5
- Service Specific Elements – these are specific to both the service and the file format used, section F.3

The following key will be used for this section:

- *O* - Optional
- *R* - Required, field must be presented
- *M* - Mandatory if Available, if the information is available, it should be presented

F.1. Available Services

Various Services can be accessed via the Batch Input Service using both the XML and CSV file formats. These are outlined in the table below.

Service Type	Service Name	Restrictions	Additional Information
Credit and Debit Cards	**Bank Card**	-	Section B.1
	Pre-Registered Cards	Account must be configured for this service	Section C.1
	Recurring Transactions – Capture Method	Account must be configured for this service	Section C.2
	Chip and PIN	Account must be configured for this service. Transaction method must be `auth` or `refund`.	Section G.1
BACS Services	**Standard Direct Debit**	Account must be configured for this service	Section E.1
	Direct Credit: DDrefunds	Account must be configured for this service	Section E.3
	Standard Direct Credit	Account must be configured for this service	Section E.4
	Direct Credit: Card Collection Accounts	Account must be configured for this service	Section E.5

In addition, the following Fraud Screening Services may be used in conjunction with the Credit and Debit Card Services:

Service Type	Service Name	Restrictions	Additional Information
Fraud Prevention for Cards	**ReD**	Account must be configured for this service. The extended dataset cannot be provided.	Section D.3
	AVSCV2	AVS check only. Not available for Chip and PIN	Section D.1

F.2. Overview of Common Elements and Attributes

F.2.1. File Formats Attributes

The individual transactions within each file may be presented in either XML or CSV format. Each format has a name, which is passed within both the Batch Submission document and the Batch Request. The file names are presented below.

XML File Formats		
Format	**Service**	**Details in**
`xml_cardtxn`	Credit and Debit Cards	Section F.3.1
`xml_chp_cardtxn`	Cardholder Present Cards	Section F.3.2
`xml_directcredit`	BACS – Direct Credit	Section F.3.3
`xml_directdebit`	BACS - Direct Debit	Section F.3.4

CSV file Formats			
Format	**Service**	**Used for**	**Details in**
`csv_avs_card_txn`	Credit and Debit Cards	Authorisations with standard AVS checking	Section F.3.6
`csv_card_txn`	Credit and Debit Cards	Authorisations	Section F.3.5
`csv_chp_card_txn`	Cardholder Present	`auth` and `refund` transactions	
`csv_historic`	Credit and Debit Cards	Historic transactions: `fulfill`, `txn_refund`, `cancel`, `accept_fraud`	Section F.3.7
`csv_prereg`	Credit and Debit Cards	Pre-Registered Cards	Section F.3.9
`csv_dc_cc_account`	BACS Services	Direct Credit `cardaccountpayment`	Section F.3.10
`csv_dc_ddrefund`	BACS Services	Direct Credit `ddrefunds`	Section F.3.11
`csv_dc_std`	BACS Services	Standard Direct Credit	Section F.3.12
`csv_dd_drawdown`	BACS Services	Direct Debit `drawdown`	Section F.3.13
`csv_dd_setup`	BACS Services	Direct Debit `setup` and `presetup`	Section F.3.14
`csv_dd_setup_edit`	BACS Services	Direct Debit `confirm` and `revoke`	Section F.3.15

F.2.2. Batch Submission Elements

These elements are common to all of the Batch Input Services, regardless of the service or format used

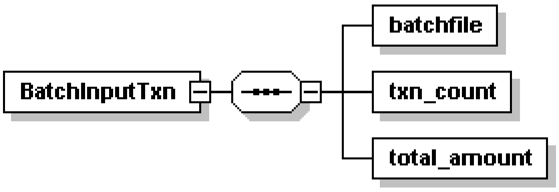
F.2.2.1. Schema Elements for Batch Submission

Each batch contains header information:

- `Request`
    - `Authentication` – each Batch Submission can contain transactions for only one DataCash account (vTID). The Authentication element is covered in section A.1.1.1
    - `Transaction`
        - `BatchInputTxn` - section F.2.2.1.1

F.2.2.1.1. BatchInputTxn

Element Name: `BatchInputTxn`
Position: `Request.Transaction`



Elements of BatchInputTxn			
Element Name	**description**	**values / limitations**	
`batchfile`	The Batch Request, compressed using Zlib compression, then Base64 encoded	Uncompressed Batch Request must be less than 1MB	*R*
`txn_count`	The total number of transactions in the batchfile	Must be an integer > 0.	*R*
`total_amount`	The total gross value of transactions in the batchfile, regardless of currency and transaction type	Tolerant to 0.001. If the format of the transactions within the Batch Request does not contain an amount, the `total_amount` must be set to zero	*R*

Attributes of batchfile			
Element Name	**description**	**values / limitations**	
`format`	The format of the file you are submitting	Please refer to section F.2.1	*R*

XML Examples of Batch Submission for BatchInputTxn complex elements

```
<BatchInputTxn>
  <batchfile format=”xml_cardtxn”>…</batchfile>
  <txn_count>3</txn_count>
  <total_amount>395.99</total_amount>
</BatchInputTxn>

<BatchInputTxn>
  <batchfile format=”csv_historic”>…</batchfile>
  <txn_count>16</txn_count>
  <total_amount>0</total_amount>
</BatchInputTxn>

<BatchInputTxn>
  <batchfile format=”xml_directdebit”>…</batchfile>
  <txn_count>4</txn_count>
  <total_amount>75.60</total_amount>
</BatchInputTxn>

<BatchInputTxn>
  <batchfile format=”csv_dd_setup”>…</batchfile>
  <txn_count>29</txn_count>
  <total_amount>0</total_amount>
</BatchInputTxn>

<BatchInputTxn>
  <batchfile format=”csv_dd_drawdown”>…</batchfile>
  <txn_count>63</txn_count>
  <total_amount>1565.09</total_amount>
</BatchInputTxn>
```

F.2.2.2. XML Example Batch Submission

Example XML Batch Submission for a Batch Request in XML format containing three transactions

```
<Request>
  <Authentication>
    <password>******</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <BatchInputTxn>
      <batchfile format=”xml_cardtxn”>…</batchfile>
      <txn_count>3</txn_count>
      <total_amount>395.99</total_amount>
    </BatchInputTxn>
  </Transaction>
</Request>
```

Example XML Batch Submission for a Batch Request in CSV format containing five transactions

```
<Request>
  <Authentication>
    <password>******</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <BatchInputTxn>
    <batchfile format=”csv_historic”>…</batchfile>
    <txn_count>5</txn_count>
    <total_amount>2501.00</total_amount>
  </BatchInputTxn>
  </Transaction>
</Request>
```

Example XML Batch Submission for a Batch Request in XML format containing five transactions

```
<Request>
  <Authentication>
    <password>******</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <BatchInputTxn>
      <batchfile format=”xml_directdebit”>…</batchfile>
      <txn_count>5</txn_count>
      <total_amount>395.99</total_amount>
    </BatchInputTxn>
  </Transaction>
</Request>
```

Example XML Batch Submission for a Batch Request in CSV format containing eighteen transactions

```
<Request>
  <Authentication>
    <password>******</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <BatchInputTxn>
    <batchfile format=”csv_dd_drawdown”>…</batchfile>
    <txn_count>18</txn_count>
    <total_amount>156.00</total_amount>
  </BatchInputTxn>
  </Transaction>
</Request>
```

F.2.2.3. Schema Elements for Response

The Response for Batch Submission contains the normal Response fields. These are covered in section A.1.2

The `datacash_reference` returned is the reference number for the Batch Submission. This should be stored to allow a Batch Query (section F.2.3) to be performed. The `merchantreference` returned is extracted from within the Batch Request

F.2.2.4. XML Example Batch Submission Response

Two Batch Submission Responses are shown below.

A full list of return codes for this service is available on the website here.

Example XML Batch Submission Responses for a successful and a rejected batch submission

```
<Response>
  <datacash_reference>3900900100010001</datacash_reference>
  <merchantreference>batch_ABCDEF</merchantreference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>…</time>
</Response>

<Response>
  <datacash_reference>3800900100010006</datacash_reference>
  <information>Supplied and actual transaction counts do
      not match</information>
  <merchantreference>batch_XYZ</merchantreference>
  <mode>LIVE</mode>
  <reason>BatchInput: Error in batch data</reason>
  <status>262</status>
  <time>…</time>
</Response>
```

F.2.3. Batch Query

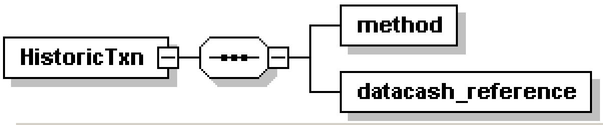
The Batch Query allows you to receive the results of each transaction within the Batch Submission.

F.2.3.1. Schema Elements for Batch Submission Query

- `Request`
  - `Authentication` - section A.1.1.1
  - `Transaction`
    - `HistoricTxn` - section F.2.3.1.1

F.2.3.1.1. HistoricTxn

Element Name:	`HistoricTxn`
Position:	`Request.Transaction`



Elements of HistoricTxn

Element Name	description	values / limitations	
`method`	The transaction type	`query`	R
`reference`	The `datacash_reference` of the Batch Submission		R

Example XML Request for HistoricTxn complex elements

```
<HistoricTxn>
  <datacash_reference>3900900100010001</datacash_reference>
  <method>querymethod>
</HistoricTxn>
```

F.2.3.2. XML Example Batch Submission Query

Example XML Batch Query

```
<Request>
  <Authentication>
    <password>*****</password>
    <client>99000001</client>
  </Authentication>
  <Transaction>
    <HistoricTxn>
      <method>query</method>
      <datacash_reference>3900900100010001</datacash_reference>
    </HistoricTxn>
  </Transaction>
</Request>
```

F.2.3.3. Schema Elements for Query Response

F.2.3.3.1. BatchInputTxn

Element Name:	`BatchInputTxn`
Position:	`Response`

The element `BatchInputTxn` contains the Zlib compressed and Base64 encoded results of the Batch Request.

F.2.3.4. XML Example Query Response

Two Example XML Batch Query Responses

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <BatchInputTxn>…<BatchInputTxn>
  <datacash_reference>3900900100010001</datacash_reference>
  <merchantreference>batch_ABCDEF</merchantreference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>…</time>
</Response>

<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>3400900100010008</datacash_reference>
  <information>You have queried a Batch Input Transaction,
      which is currently being processed</information>
  <merchantreference>batch_ABCDEG</merchantreference>
  <mode>LIVE</mode>
  <reason>BatchInput: Processing</reason>
  <status>273</status>
  <time>…</time>
</Response>
```

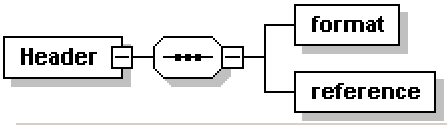
F.2.4. Batch Input Request and Response - Common XML Elements

F.2.4.1. Schema Elements for Batch Request

- `BatchInputRequest`
  - `Header` – see section F.2.4.1.1
  - `Transactions` – see section F.2.4.1.2

F.2.4.1.1. Header

Element Name:	`Header`
Position:	`BatchInputRequest`



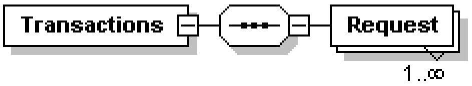
Elements of Header			
Element Name	**description**	**values / limitations**	
`format`	The name of the file format used	See section F.2.1 for list. Must match that listed in the Batch Submission	*R*
`reference`	A reference number for the entire Batch	Must be between six and thirty alphanumeric in length	*R*

Example XML Request for Header complex elements

```
<Header>
  <format>xml_cardtxn</format>
  <reference>mybatch0002</reference>
</Header>

<Header>
  <format>xml_directdebit</format>
  <reference>mybatch_0003</reference>
</Header>
```

F.2.4.1.2. Transactions

Element Name: `Transactions`
Position: `BatchInputRequest`



Elements of Transactions

Element Name	description	values / limitations	
Request	One Request element per transaction	Each Request must have identical Authentication elements, which must also be the same as those supplied in the Authentication element of the Batch Submission	*R*

Example XML for Transactions complex elements. There are three transactions present

```
<Transactions>
  <Request>
    ...
  </Request>
  <Request>
    ...
  </Request>
  <Request>
    ...
  </Request>
</Transactions>
```

F.2.4.2. Example XML Batch Request

Example XML for a Batch Request

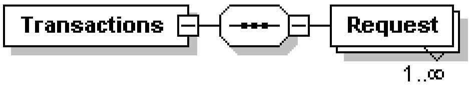
```
<BatchInputRequest>
  <Header>
    <format>xml_cardtxn</format>
    <reference>batchref1234</reference>
  </Header>
  <Transactions>
    <Request>
      ...
    </Request>
    <Request>
      ...
    </Request>
    <Request>
      ...
    </Request>
  </Transactions>
</BatchInputRequest>
```

F.2.4.3. Schema Elements for Batch Response

- `BatchInputResponse`
  - `Header` – this element is currently unpopulated
  - `Transactions` – see section F.2.4.3.1

F.2.4.3.1. Transactions

Element Name: `Transactions`
Position: `BatchInputResponse`



Elements of Transactions	
Element Name	**description**
`Response`	One Response element per original transaction

Example XML for Transactions complex elements. There are two transactions present

```
<Transactions>
  <Response>
    ...
  </Response>
  <Response>
    ...
  </Response>
</Transactions>
```

F.2.4.4. Example XML Batch Response

Example XML for a Batch Response, without the transaction details shown

```
<BatchInputResponse>
  <Header/>
  <Transactions>
    <Response>...</Response>
    <Response>...</Response>
  </Transactions>
</BatchInputResponse>
```

An Example XML Response, with three transactions

```
<BatchInputResponse>
  <Header></Header>
  <Transactions>
    <Response>
      <CardTxn>
        <authcode>183425</authcode>
        <card_scheme>Mastercard</card_scheme>
        <country>United Kingdom</country>
        <Cv2Avs>
          <cv2avs_status>ADDRESS MATCH ONLY</cv2avs_status>
          <policy>1</policy>
        </Cv2Avs>
        <issuer>BARCLAYS BANK PLC</issuer>
      </CardTxn>
      <datacash_reference>3300900100010037</datacash_reference>
      <merchantreference>1000001</merchantreference>
      <mode>LIVE</mode>
      <reason>ACCEPTED</reason>
      <status>1</status>
      <time>1067274153</time>
    </Response>
    <Response>
      <datacash_reference>3300900100010037</datacash_reference>
      <merchantreference>123402</merchantreference>
      <reason>Prereg: Merchant Not Subscribed</reason>
      <status>251</status>
      <time>1067274153</time>
    </Response>
    <Response>
      <CardTxn>
        <authcode>100000</authcode>
      </CardTxn>
      <datacash_reference>3300900100010037</datacash_reference>
      <merchantreference>123402</merchantreference>
      <reason>ACCEPTED</reason>
      <status>1</status>
      <time>1067274153</time>
    </Response>
  </Transactions>
</BatchInputResponse>
```

F.2.5. Batch Request and Response - Common CSV Elements

F.2.5.1. CSV File Format Definition

The definition of CSV format used is as follows

- Allowable characters within a CSV field include 0x09 (tab) and the inclusive range of 0x20 (space) through 0x7E (tilde).
- A field within CSV may be surrounded by double-quotes. Double quoting is used by default to surround fields in CSV format responses.
- A field within CSV must be surrounded by double-quotes to contain a comma.
- A field within CSV must be surrounded by double-quotes to contain an embedded double-quote, represented by a pair of consecutive double-quotes.
- A CSV field may be terminated by 0x0A (line feed) or by 0x0D, 0x0A (carriage return, line feed).
- Blank lines and lines starting with 0x23 (#) are ignored.

F.2.5.2. CSV Request Headers

Each CSV file contains should contain a header line containing these fields:

Headers for CSV format, in order			
Element Name	**description**	**values / limitations**	
`format`	The name of the file format used	See section F.2.1 for list. Must match that listed in the Batch Submission	*R*
`reference`	A reference number for the entire Batch	Must be between six and thirty alphanumeric in length	*R*
`client`	The vTID of the account the batch is to be processed against	Must be the same as those supplied in the Authentication element of the Batch Submission	*R*
`password`	The password of the account the batch is to be processed against		*R*

Two CSV Examples for a Batch Request Headers

```
csv_card_txn,batch_csv_card_txn_6,99000001,mypasswd

csv_prereg,batch_csv_prereg_3,99000001,mypasswd
```

F.2.5.3. CSV Response

The Response elements returned will be the same for each file format. These are:

Fields for CSV responses			
	Element Name	**description**	**XML equivalent**
1	`merchant_ref`	The `merchant_ref` of the original transaction	`merchant_reference`
2	`datacash_ref`	The `datacash_reference` of the transaction	`datacash_reference`
3	`response_code`	The DataCash Response Code	`status`
4	`status_message`	Additional information about the `response_code`	`reason`
5	`authcode`	The result of the authorisation request, for transactions submitted for authorisation only –otherwise blank	`authcode`
6	`time`	The Unix Timestamp of the transaction	`time`

Example CSV responses

```
“ABCD9876”,“3000900100010005”,“7”,“REFERRED”,“CALL AUTH”,“1080920429”

“3400900100010008”,“3400900100010008”,“1”,“CANCELLED
OK”,“”,“1080920429”

“payment_1223”,“3600900100010889”,“1”,“ACCEPTED”,“896786”,“1080920430”

“ACBD9877”,“3600900100010890”,“1”,“ACCEPTED”,“548620”,“1080920431”

"valid_ddrefund","3000900100010048","1","ACCEPTED","","1080920435"

"ddrefund_no_setup","3800900100010049","137","Unable to locate
matching DDI transaction","","1080920436"
```

F.3. Example Batch Requests

This section covers the Batch Request file, for all services.

Please note that the maximum size of a single Batch Request – before compression – is 1MB (1024KB). Any files larger than this will be rejected

F.3.1. xml_cardtxn

An example file with three transactions is shown below. The file contains three transactions, one Pre-Registered and two Credit and Debit Card (a `pre` with AVS data and a `cancel`). The transactions could be screened using either the ReD or T3M services if the account is configured for this at DataCash. A Batch Submission example for this file is shown in section F.2.2.1.

Example XML Batch Request containing three transactions with a total value of 395.99

```
<BatchInputRequest>
  <Header>
    <format>xml_cardtxn</format>
    <reference>batchref1234</reference>
  </Header>
  <Transactions>
    <Request>
      <Authentication>
        <client>99000001</client>
        <password>******</password>
      </Authentication>
      <Transaction>
        <CardTxn>
          <Card>
            <expirydate>04/06</expirydate>
            <startdate>01/04</startdate>
            <pan>633300*********1</pan>
            <Cv2Avs>
              <street_address1>1 High Street, UK </street_address1>
              <postcode>S01 2CD</postcode>
              <policy>1</policy>
            </Cv2Avs>
          </Card>
          <method>pre</method>
        </CardTxn>
        <TxnDetails>
          <amount currency="GBP">146.00</amount>
          <merchantreference>1000001</merchantreference>
        </TxnDetails>
      </Transaction>
    </Request>
    <Request>
      <Authentication>
        <password>******</password>
        <client>99000001</client>
      </Authentication>
      <Transaction>
        <TxnDetails>
          <merchantreference>123402</merchantreference>
          <amount currency="USD">249.99</amount>
        </TxnDetails>
        <CardTxn>
          <method>auth</method>
          <card_details type="preregistered">
              3100000088888881212</card_details>
        </CardTxn>
      </Transaction>
    </Request>
    <Request>
      <Authentication>
        <client>99000001</client>
        <password>******</password>
      </Authentication>
      <Transaction>
        <HistoricTxn>
          <reference>3100000088888888</reference>
          <method>txn_refund</method>
       </HistoricTxn>
      </Transaction>
    </Request>
  </Transactions>
</BatchInputRequest>
```

F.3.2. xml_chp_cardtxn

Example XML Batch Request containing three transactions with a total value of 120.72 in two currencies

```
<BatchInputRequest>
  <Headder>
    <format>xml_chp_cardtxn</format>
    <reference>batchref1234</reference>
  </Headder>
  <Transactions>
    <Request>
      <Authentication>
        <password>*******</password>
        <client>99000001</client>
      </Authentication>
      <Transaction>
        <TxnDetails>
          <capturemethod>parked</capturemethod>
          <merchantreference>1234567890blahh</merchantreference>
          <amount currency='GBP'>54.00</amount>
        </TxnDetails>
        <CardTxn>
          <card_details type="track2_data">
               ****************</card_details>
          <Terminal id='82000062'>
            <terminal_capabilities ic_reader='true'
               magnetic_stripe_reader='true'
               manual_card_entry='true' />
            <features_capabilities pin_pad_available='true' />
          </Terminal>
          <method>auth</method>
          <reason_online_code>10</reason_online_code>
          <receipt_no>000123</receipt_no>
          <ICC>
            <seq_or_issue_number>*</seq_or_issue_number>
            <auth_response_code>00</auth_response_code>
            <crypto_txn_amount>54.00</crypto_txn_amount>
            <crypto_tran_type>00</crypto_tran_type>
            <term_txn_date>031107</term_txn_date>
            <txn_currency_code>826</txn_currency_code>
            <txn_country_code>826</txn_country_code>
            <arqc>****************</arqc>
            <aip>****</aip>
            <atc>****</atc>
            <unpredictable>********</unpredictable>
            <tvr>**********</tvr>
            <issuer_app_data>**************</issuer_app_data>
            <app_usage_control>****</app_usage_control>
            <crypto_info_data>**</crypto_info_data>
            <cvm>******</cvm>
            <aid>**********</aid>
            <term_app_ver_no>****</term_app_ver_no>
            <txn_status_info>****</txn_status_info>
            <term_type>**</term_type>
            <term_capabilities>******</term_capabilities>
            <pos_entry_mode>32</pos_entry_mode>
            <other_card_data>**************</other_card_data>
          </ICC>
        </CardTxn>
      </Transaction>
    </Request>
    <Request>
      <Authentication>
        <password>*******</password>
        <client>99000001</client>
      </Authentication>
      <Transaction>
        <TxnDetails>
          <capturemethod>keyed</capturemethod>
          <merchantreference>1234560002</merchantreference>
          <amount currency='EUR'>30.00</amount>
        </TxnDetails>
        <CardTxn>
          <Terminal id='82000062'>
            <terminal_capabilities ic_reader='true'
                magnetic_stripe_reader='true'
                manual_card_entry='true' />
            <features_capabilities pin_pad_available='true' />
          </Terminal>
          <method>auth</method>
          <receipt_no>000124</receipt_no>
          <ICC>
            <term_type>**</term_type>
          </ICC>
          <Card>
            <pan>444433*********1</pan>
            <expirydate>12/09</expirydate>
          </Card>
        </CardTxn>
      </Transaction>
    </Request>
    <Request>
      <Authentication>
        <password>*******</password>
        <client>99000001</client>
      </Authentication>
      <Transaction>
        <TxnDetails>
          <capturemethod>swiped</capturemethod>
          <merchantreference>1234560002</merchantreference>
          <amount currency='GBP' cashback='20.00'>26.23</amount>
        </TxnDetails>
        <CardTxn>
          <card_details type="track2_data">
               ****************</card_details>
          <Terminal id='82000062'>
            <terminal_capabilities ic_reader='true'
                magnetic_stripe_reader='true'
                manual_card_entry='true' />
            <features_capabilities pin_pad_available='true' />
          </Terminal>
          <method>auth</method>
          <receipt_no>000125</receipt_no>
          <ICC>
            <term_type>**</term_type>
          </ICC>
        </CardTxn>
      </Transaction>
    </Request>
   <Request>
      <Authentication>
        <password>*******</password>
        <client>99000001</client>
      </Authentication>
      <Transaction>
        <TxnDetails>
          <capturemethod>keyed</capturemethod>
          <merchantreference>1234560002</merchantreference>
          <amount currency='GBP'>10.49</amount>
        </TxnDetails>
        <CardTxn>
          <Terminal id='82000062'>
            <terminal_capabilities ic_reader='true'
                magnetic_stripe_reader='true'
                manual_card_entry='true' />
            <features_capabilities pin_pad_available='true' />
          </Terminal>
          <method>refund</method>
          <receipt_no>000126</receipt_no>
          <ICC>
            <term_type>**</term_type>
          </ICC>
          <Card>
            <pan>444433*********1</pan>
            <expirydate>12/09</expirydate>
          </Card>
        </CardTxn>
      </Transaction>
    </Request>
  </Transactions>
</BatchInputRequest>
```

F.3.3. xml_direct_credit

Example XML Batch Request xml_directcredit containing two transactions with a total value of 80.00

```
<BatchInputRequest>
  <Header>
    <format>xml_directcredit</format>
    <reference>45356132</reference>
  </Header>
  <Transactions>
    <Request>
      <Authentication>
        <password>fred</password>
        <client>21850000</client>
      </Authentication>
      <Transaction>
        <TxnDetails>
          <merchantreference>standardDC0000054</merchantreference>
          <amount>45.00</amount>
        </TxnDetails>
        <DirectCreditTxn>
          <method>directcredit</method>
          <sortcode>826300</sortcode>
          <accountnumber>80000990</accountnumber>
          <accountname>Jo Bloggs</accountname>
        </DirectCreditTxn>
      </Transaction>
    </Request>
    <Request>
      <Authentication>
        <password>fred</password>
        <client>21850000</client>
      </Authentication>
      <Transaction>
        <TxnDetails>
          <merchantreference>ddrefund00434582</merchantreference>
          <amount>35.00</amount>
        </TxnDetails>
        <HistoricTxn>
          <method>ddrefund</method>
          <reference>3600900100010007</reference>
        </HistoricTxn>
      </Transaction>
    </Request>
  </Transactions>
</BatchInputRequest>
```

F.3.4. xml_directdebit

An example `xml_directdebit` file with four transactions is shown below. The file contains one `setup`, one `revoke`, two `drawdown`s and a `confirm`. A Batch Submission example for this file is shown in section F.2.2.1.

Example XML Batch Request containing five transactions with a total value of 395.99

```
<BatchInputRequest>
  <Header>
    <format>xml_directdebit</format>
    <reference>batchref1236</reference>
  </Header>
  <Transactions>
    <Request>
      <Authentication>
        <client>99000001</client>
        <password>******</password>
      </Authentication>
      <Transaction>
        <DirectDebitTxn>
          <sortcode>938611</sortcode>
          <accountnumber>02149187</accountnumber>
          <accountname>Mr A. N. Other</accountname>
          <method>setup</method>
        </DirectDebitTxn>
        <TxnDetails>
          <merchantreference>1000001</merchantreference>
        </TxnDetails>
      </Transaction>
    </Request>
    <Request>
      <Authentication>
        <password>******</password>
        <client>99000001</client>
      </Authentication>
      <Transaction>
        <TxnDetails>
          <merchantreference>123402</merchantreference>
          <amount>100.99</amount>
        </TxnDetails>
        <HistoricTxn>
          <method>drawdown</method>
          <reference>90000005</reference>
        </HistoricTxn>
      </Transaction>
    </Request>
    <Request>
      <Authentication>
        <password>******</password>
        <client>99000001</client>
      </Authentication>
      <Transaction>
        <TxnDetails>
          <merchantreference>123402</merchantreference>
          <amount>295.00</amount>
        </TxnDetails>
        <HistoricTxn>
          <method>drawdown</method>
          <reference>90000005</reference>
          <duedate>20050929</duedate>
        </HistoricTxn>
      </Transaction>
    </Request>
    <Request>
      <Authentication>
        <client>99000001</client>
        <password>******</password>
      </Authentication>
      <Transaction>
        <HistoricTxn>
          <method>revoke</method>
          <reference>40488707</reference>
        </HistoricTxn>
        <TxnDetails>
          <merchantreference>383285870486111</merchantreference>
        </TxnDetails>
      </Transaction>
    </Request>
    <Request>
      <Authentication>
        <client>99000001</client>
        <password>********</password>
      </Authentication>
      <Transaction>
        <HistoricTxn>
          <method>confirm</method>
          <reference>40502431</reference>
        </HistoricTxn>
        <TxnDetails>
          <merchantreference>383366196990741</merchantreference>
        </TxnDetails>
      </Transaction>
    </Request>
  </Transactions>
</BatchInputRequest>
```

F.3.5. csv_card_txn

This CSV file format should be used to obtain authorisation for Credit and Debit Card transactions using the Bank Card Service. It can be used for both the one stage processing model and the first stage of the two stage processing model.

If you are using the Capture Method model of Recurring Transactions, this file format can be used. For this Service, the `capturemethod` should be specified

Fields for format csv_card_txn				
	Element Name	**description**	**values / limitations**	
1	`pan`	As described in section B.1.1.1		*R*
2	`expiry_date`			*R*
3	`start_date`			*M*
4	`issue`			*M*
5	`amount`	As described in section B.1.1.3		*R*
6	`currency`			*R*
7	`merchant_ref`			*R*

8	`method`	The transaction type	`auth`
			`pre`
			`refund`
			`erp`
				R

9	`capturemethod`	As described in section C.2.1.1		*M*[1]
10	`authcode`	The authorisation code received from the bank	If presented, must be value received from Banks Authorisation centre	*O*

1. Mandatory if the account is configured with multiple merchant ID environments

Example csv_card_txn file – six transactions with a value of 2697.31

```
csv_card_txn,batch_csv_card_txn_6, 99000001,mypasswd
4444333322221111,01/09,,,99.99,GBP,myref0000023,pre,ecomm,
5473000000000007,01/09,,,530.01,GBP, myref0000024,auth,ecomm,
343434343434343,01/09,,,13.00,USD, myref0000025,pre,,
4936000000000000001,01/09,,2,9.95,GBP,
myref0000026,erp,ecomm,
6759000000000026,01/09,01/01,,44.36,GBP,
myref0000027,refund,ecomm,
5473000000000007,01/09,,,2000.00,GBP,
myref123,auth,ecomm,123456
```

F.3.6. csv_avs_card_txn

This CSV file format is similar to `csv_card_txn` format (section F.3.4), but it also allows AVS information to be provided with the Credit and Debit Card transactions if required. CV2 information cannot be provided as the CV2 number should not be stored

Fields for format csv_avs_card_txn

	Element Name	description	values / limitations	
`1-10`	As described in section F.3.5			
`11`	`address_line_1`	As described in section D.1.1.1		*O*
`12`	`address_line_2`			*O*
`13`	`address_line_3`			*O*
`14`	`address_line_4`			*O*
`15`	`postcode`			*O*
`16`	`policy_number`	The Standard Policy against which the transaction is to be checked against	See website[1]	*O*[2]

1. While all of these values are valid, as only the AVS check can be carried out, only policies 1 or 5 should be chosen.
2. The default policy registered against the account will be used if not value is set

Example csv_avs_card_txn file – two transactions for a total value of 3.96

```
csv_avs_card_txn,batch_csv_avs_card_txn_2,99000001,mypasswd
4444333322221111,01/09,,,1.98,GBP,ABC123,auth,,,Flat 12,345 Main
     Street,Putney,,A9 87XY,5
5473000000000007,01/09,,,1.98,GBP,ABC124,auth,ecomm,,Flat 12,345
     Main Street,Putney,,X1 23AB,5

# Note: the line wrapping is this example should not
# be used in a real file
```

F.3.7. csv_chp_card_txn

This CSV file format is used for Cardholder Present transactions

Fields for format csv_chp_card_txn				
	Element Name	description	values / limitations	
1-7	As described in section F.3.5			
8	method	The transaction type	auth refund	*R*
9	capturemethod	Indicates how the transaction was accepted at the POS	keyed swiped	*R*
10	authcode	The authorisation code generated for the transaction	-	*R*
11	authcode_method	Indicates how the authcode was obtained	terminal online telephone	*O*
12	cashback	The amount of cashback required by the customer	Amount in Major.Minor currency units (i.e. 10.98)	*O*
13	tid	The Terminal ID (TID) used for the transaction	DataCash will advise	*R*
14	hot_card_file	Indicates which Hot Card File was used during the transaction	none reserved Switch 400 Switch 8000	*O*
15	ic_reader	Whether the Terminal has Chip reading capability	true false	*R*
16	magnetic_stripe_reader	Whether the Terminal has magnetic stripe reading capability	true false	*R*
17	manual_card_entry	Whether the Terminal has the facility to enable card details to be manually entered	true false	*R*
18	pin_pad_available	Whether the Terminal has a keypad to enable the PIN to be entered	true false	*R*

Example csv_chp_card_txn file containing three transactions with a value of 643

```
csv_chp_card_txn,batch_csv_chp_card_txn_0008,99000001,mypasswd
4444333322221111,01/09,,,99.99,GBP,myref0000023,refund,keyed,482135,terminal,
      85621087,,true,true,true,true
5473000000000007,01/09,,,530.01,GBP,myref0000024,auth,swiped,955AU3,telephone,
      85621087,,true,true,true,true
343434343434343,01/09,,,13.00,USD,myref0000025,auth,swiped,FLDURN,terminal,
      85621086,,true,true,true,true

# Note: the line wrapping is this example should not
# be used in a real file
```

F.3.8. csv_historic

This CSV file format is used for the Credit and Debit Card Services to complete two stage processing using the `fulfill`transaction and to perform `cancel`, `txn_refund` and `accept_fraud` transactions.

Fields for format csv_historic				
	Element Name	**description**	**values / limitations**	
1	`reference`	DataCash unique reference of the original transaction	-	*R*

2	`method`	The transaction type	`fulfill`
			`txn_refund`
			`cancel`
			`accept_fraud`
				R

3	`authcode`	The authorisation code of the original `pre` or `erp` transaction.	For `fulfill` only	*R*
4	`amount`	The value of the transaction	For `fulfill` or `txn_refund` only	*O*[1]

1. Transactions will be fulfilled / refunded for the full value of the original transaction if this field is empty

Example csv_historic file containing five transactions with a value of 2501.00

```
csv_historic,batch_csv_historic_0008,99000001,mypasswd
# fulfill supplying amount
3000900100010005,fulfill,100000,1001.00
# cancel
3400900100010008,cancel,,
# fulfill, for the full value of the original txn
3800900100010025, ,100001,

# txn_refund, two refunds to the same original txn
3600900100010026,txn_refund,,1000.00
3600900100010026,txn_refund,,500.00
```

F.3.9. csv_prereg

This CSV file format allows the Pre-Registered Card Service to be utilised

Fields for format csv_prereg			
	Element Name	**description**	**values / limitations**
1	`merchant_ref`	As described in section F.3.4	*R*
2	`reference`	DataCash unique reference of the original transaction	*R*
3	`amount`	As described in section F.3.4	*R*
4	`currency`		*R*
5	`method`		*R*
6	`capturemethod`		*O*
7	`authcode`		*O*

Example csv_prereg file containing three transactions for a total value of 2023.34

```
csv_prereg,batch_csv_prereg_3,99000001,mypasswd

# card txns using card details from previous historic txns
prereg_230006,2185000198760000,1000.00,GBP,auth,ecomm,
prereg_230007,3200900100010014,1000.00,USD,refund,,

# using prereg to carry out manual auth of previous
# referred txn
prereg_230008,3700900100010021,23.34,GBP,pre,ecomm,123123
```

F.3.10. csv_dc_cc_account

This CSV file format enables the Direct Credit Card Collection Account Service to be accessed

Fields for format csv_dc_cc_account				
	Element Name	**description**	**values / limitations**	
1	`merchant_ref`	The reference number of the drawdown	See limitations	*R*
2	`pan`	The card number		*R*
3	`amount`	The value to credit	`Major.minor` currency units (ie (10.99)	*R*
4	`card_name`	The card holders name		*O*
5	`expiry_date`	The card expiry date	–	*O*

Example csv_dc_cc_account file containing two transactions with a value of 20.00

```
csv_dc_cc_account,batch000134,99000001,mypasswd
valid_cardaccountpayment,4560720000000007,10.00,,
optional_fields,4560720000000007,10.00,MR FOO BAR,12/09
```

F.3.11. csv_dc_ddrefund

This CSV format supports Direct Credit ddrefund transactions

Fields for format csv_dc_ddrefund				
	Element Name	**description**	**values / limitations**	
1	merchant_ref	The reference number of the DDI	See limitations	*R*
2	reference	The datacash_reference of the original transaction	-	*R*
3	amount	The amount to credit	Major.minor currency units (ie (10.99)	*R*

Example csv_dc_ddrefund file containing two transactions with a value of 25.00

```
csv_dc_ddrefund,batch000131,99000001,mypasswd
ddrefundABCD000345,10000003,10.00
45347ACBD,3400900100010008,15.00
```

F.3.12. csv_dc_std

This CSV format enables the Standard Direct Credit service to be used

Fields for format csv_dc_standard				
	Element Name	**description**	**values / limitations**	
1	merchant_ref	The reference number of the drawdown	See limitations	*R*
2	amount	The value to credit	Major.minor currency units (ie (10.99)	*R*
3	sort_code	The customer's sort code	6 digits	*R*
4	acc_num	The customer's account number	–	*R*
5	acc_name	The customer's account name	Maximum of 18 characters	*R*
6	bank_ref		See limitations	*O*

Example csv_dc_std file containing two transactions with a value of 60.00

```
csv_dc_standard,batch000130,99000001,mypasswd
directcredit00006,30.00,826300,80000990,Bert Weiss,
credit00009,30.00,826300,80000990,Bert Weiss,Money for petrol
```

F.3.13. csv_dd_drawdown

This CSV file format supports `drawdown` transactions for the Standard Direct Debit Service

Fields for format csv_dd_drawdown				
	Element Name	**description**	**values / limitations**	
1	`merchant_ref`	The reference number of the drawdown	See limitations	*R*
2	`reference`	The datacash_reference number of the DDI	-	*R*
3	`amount`	The value to debit	`Major.minor` currency units (ie (10.99)	*R*
4	`duedate`	The date to perform the drawdown	`yyyymmdd` format	*O*

5	`bacs_tran_code`	Enables the BACS transaction code to be explicitly set	`01`
			`17`
			`18`
			`19`
				O

Example csv_dd_drawdown file containing three transactions with a value of 30.00

```
csv_dd_drawdown,batch000130,99000001,mypasswd
drawdown_BE30006,10000003,10.00,,
drawdown_HI90135,10000004,10.00,20041212,
drawdown_JG00042,10000005,10.00,,18
```

F.3.14. csv_dd_setup

This CSV format is used for the Standard Direct Debit Service to perform `setup` and `presetup` transactions.

Fields for format csv_dd_setup				
	Element Name	**description**	**values / limitations**	
1	`merchant_ref`	The reference number of the DDI	Must be of the format agreed with your Sponsoring bank. See limitations	*R*
2	`sort_code`	The sort code of the customer's bank	6 digits	*R*
3	`acc_num`	The customer's account number	-	*R*
4	`acc_name`	The name of the account holder	Maximum of 18 characters	*R*
5	`method`	Determines the processing model to be used	`setup` `presetup`	*R*
6	`active`	Enables an existing setup to be transferred to the DataCash system	`true`	*O*
7	`type`	Enables a non_AUDDIS DDI to be converted to an AUDDIS DDI	`conversion`	*O*

Example csv_dd_setup file containing four transactions with a value of 0.00 (no amount values present)

```
csv_dd_setup,batch000128,99000001,mypasswd
valid_setup,938611,02149187,MR F BAR,setup,,
setup_active,938612,02149188,MR A N OTHER,setup,true,
setup_conversion,938613,02149189,MR A N OTHER,setup,,conversion
# DD presetup transaction
valid_presetup,938611,02149187,MR F BAR,presetup,,
valid_presetup456, 938611,02149236,”J, M & P Smith‟,presetup,,
```

“

F.3.15. csv_dd_setup_edit

This CSV file format is used with the Direct Debit Service to perform for `confirm` and `revoke` transactions

Fields for format csv_dd_setup_edit				
	Element Name	**description**	**values / limitations**	
1	`merchant_ref`	The reference number of the DDI	As supplied for the original	*R*
2	`reference`	The `datacash_reference` of the original transaction	-	*R*
3	`method`	The operation required	`confirm` `revoke`	*R*

Example csv_dd_setup_edit file containing four transactions with a value of 0.00 (no amount values present)

```
csv_dd_setup_edit,batch000129,99000001,mypasswd
# confirm presetup with 8 digit Unique Reference
confirm_presetup,12345678,confirm
# revoke setup with 8 digit Unique Reference
revoke_setup,12345679,revoke
# confirm presetup with 16 digit DataCash Reference returned by
previous query response
confirm_presetup2,3400900100010008,confirm
# revoke previous setup in the same batch
revoke_previous_setup,3400900100010008,revoke
```

G. Card Holder Present

G.1. Chip and Pin

A technical introduction to this service is available on the website:
http://www.datacash.com/services/chip_and_pin/overview.shtml

G.1.1. Schema Elements for Request

As different fields may be presented in different situations, each field will be labelled with the following key:

- *O* - Optional
- *R* - Required, field must be presented
- *M* – Mandatory, field must be presented if it is available
- *X* – Excluded, field is excluded in specific situations

Transactions with Card Details
The `creditcheck` and `auth` transaction types require certain information to be presented. This is passed in several places in the schema:

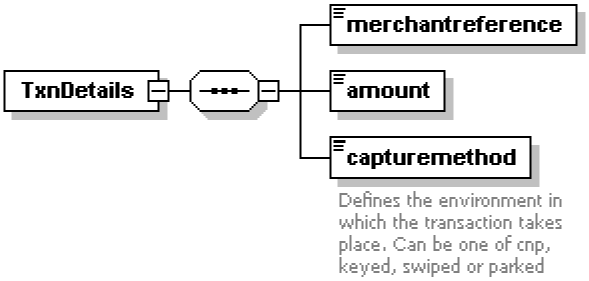
- `Request`
    - `Authentication` – section A.1.1.1
    - `Transaction`
        - `TxnDetails` – section G.1.1.1
        - `CardTxn` – section G.1.1.2
            - `Card` – section G.1.1.4
            - `ICC` - section 0
            - `Terminal` – section G.1.1.3

Cancellations and Reversals
The cancel transaction type requires information to be passed in:

- `Request`
    - `Authentication` – section A.1.1.1
    - `Transaction`
        - `HistoricTxn` – section G.1.1.5

G.1.1.1. TxnDetails

Element Name: `TxnDetails`
Position: `Request.Transaction`



Defines the environment in which the transaction takes place. Can be one of cnp, keyed, swiped or parked

Elements of TxnDetails			
Element Name	**description**	**Values / limitations**	**Required?**
`merchantreference`	A unique reference number to distinguish each transaction	See Answer	R
`amount`	The value of the transaction		R

`capturemethod`	Indicates how the transaction was accepted at the POS	`parked`
		`keyed`
		`swiped`
			R

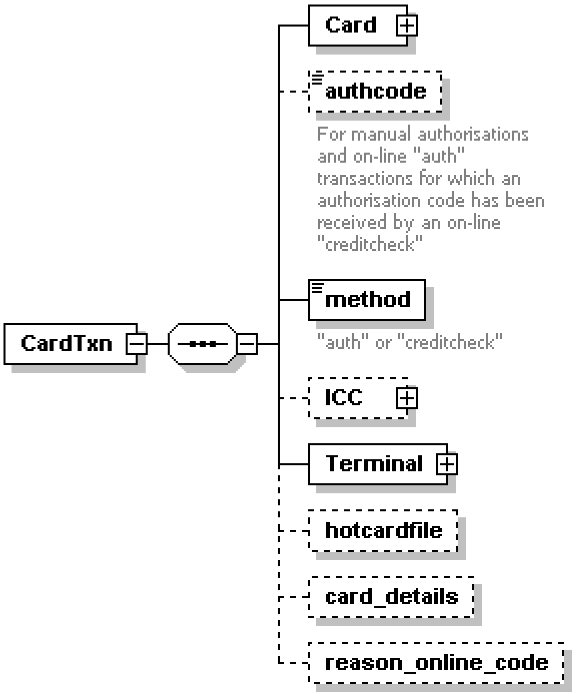
Attributes of TxnDetails			
Attribute Name	**Attribute of element**	**description**	**Required?**
`cashback`	`amount`	The amount of cashback requested by the customer (Cannot be supplied with `cashadvance` attribute)	O
`cashadvance`	`amount`	Indicates whether the transaction was a Cash Advance (Cannot be supplied with `cashback` attribute)	O

Example XML for TxnDetails element

```
<TxnDetails>
      <merchantreference>123402</merchantreference>
      <amount currency=’GBP’ cashback=’30.00’>59.99</amount>
      <capturemethod>parked</capturemethod>
</TxnDetails>

<TxnDetails>
      <merchantreference>AB78324293452</merchantreference>
      <amount currency=’EUR’>26.50</amount>
      <capturemethod>swiped</capturemethod>
</TxnDetails>

<TxnDetails>
      <merchantreference>AB78324293452</merchantreference>
      <amount currency=’GBP’ cashadvance=’true’>26.50</amount>
      <capturemethod>swiped</capturemethod>
</TxnDetails>
```

G.1.1.2. CardTxn

Element Name:	CardTxn
Position:	Request.Transaction


Card
authcode
For manual authorisations and on-line "auth" transactions for which an authorisation code has been received by an on-line "creditcheck"
method
"auth" or "creditcheck"
CardTxn
ICC
Terminal
hotcardfile
card_details
reason_online_code

Elements of CardTxn					
Element Name	**description**	**Values / Limitations**	**keyed**	**swiped**	**parked**
authcode	The authorisation code, if on-line authorisation is not required	See note below	O	X	X
Card	See section G.1.1.4		R	X	X
card_details	Data read from the magnetic stripe of the card, or equivalent from the ICC	Must be Base64 encoded	-	X	X
hotcardfile	Indicates which Hot Card File was used during the transaction	-	O	O	O
ICC	See section 0		-	-	R
method	The transaction method	auth creditcheck	R	R	R
reason_online_code	For on-line transactions: the reason for being on-line. As defined by APACs		O	O	O
Terminal	See section G.1.1.3		R	R	R

Note. For swiped and parked transactions, card details may be presented in either the Card element or the card_details element:

1. The Card element is required if an authorisation code is submitted
2. The card_details element is required if the authorisation code is not submitted.

Attributes of CardTxn

Attribute Name	Attribute of element	description	Values / limitations	Required?

`method`	`authcode`	Indicates how the authcode was obtained.	Must be present if authcode element is presented:
			`online`
			`terminal`
			`telephone`
				R

`auth_datetime`	`authcode`	The actual date & time of the transaction	`As a unixtimestamp`	R[1]
`type`	`card_details`	Indicates the type of data	`track2_data`	R

[1] Field is required for merchants using NatWest Streamline. For other Acquiring Banks this field is optional

Example XML for CardTxn element, for a parked transaction

```
<CardTxn>
  <Terminal id="…">…</Terminal>
  <card_details type="track2_data">
       NDkyOTQ5ODMxMTQwMDAwMj0wODAzMjAxMDA5ODgwMDAwMDAwMR8wMA==
  </card_details>
  <hotcardfile>…</hotcardfile>
  <ICC>…</ICC>
  <method>creditcheck</method>
  <reason_online_code>10</reason_online_code>
</CardTxn>
```

Example XML for CardTxn element, for a swiped transaction without an authcode

```
<CardTxn>
  <Terminal id="…">…</Terminal>
  <card_details type="track2_data">
       NDkyOTQ5ODMxMTQwMDAwMj0wODAzMjAxMDA5ODgwMDAwMDAwMR8wMA==
  </card_details>
  <method>creditcheck</method>
</CardTxn>
```

Example XML for CardTxn element, for a swiped or parked transaction with authcode

```
<CardTxn>
  <Terminal id="…">…</Terminal>
  <Card>…</Card>
  <method>creditcheck</method>
  <authcode method="terminal"
       auth_datetime="1164821283">451576</authcode>
</CardTxn>
```

Example XML for CardTxn element, for a keyed transaction

```
<CardTxn>
  <Terminal id="…">…</Terminal>
  <Card>…</Card>
  <method>auth</method>
</CardTxn>
```

G.1.1.3. Terminal

Element Name: `Terminal`
Position: `Request.Transaction.CardTxn`

Elements of Terminal
Element Name
terminal_capabilities
features_capabilities
message_capabilities

Attributes of Terminal and its children				
Attribute Name	**Attribute of element**	**description**	**values**	**Req.?**
id	Terminal	The terminal or TID number	DataCash will advise	R
ic_reader	terminal_capabilities	Chip reading functionality	true false	R
magnetic_stripe_reader	terminal_capabilities	Magnetic stripe reading functionality	true false	R
manual_card_entry	terminal_capabilities	Facility to manually enter card details	true false	R
cardholders_device	features_capabilities	Cardholder’s device (eg. personal computer, mobile phone, digital TV or similar device)	true false	O
card_capture_device	features_capabilities	Terminal or operator able to capture cards	true false	O
pin_pad_available	features_capabilities	Keypad to enable PIN to be entered	true false	R
unattended_device	features_capabilities	Indicates whether	true false	O

		transaction is processed from an unattended terminal		
downline_load_floor_limit	message_capabilities	As per APACS standard 70 book 2 appendix A.9	true false	O
downline_load_referral	message_capabilities		true false	O
hold_capability	message_capabilities		true false	O
response_additional_data_support	message_capabilities		true false	O

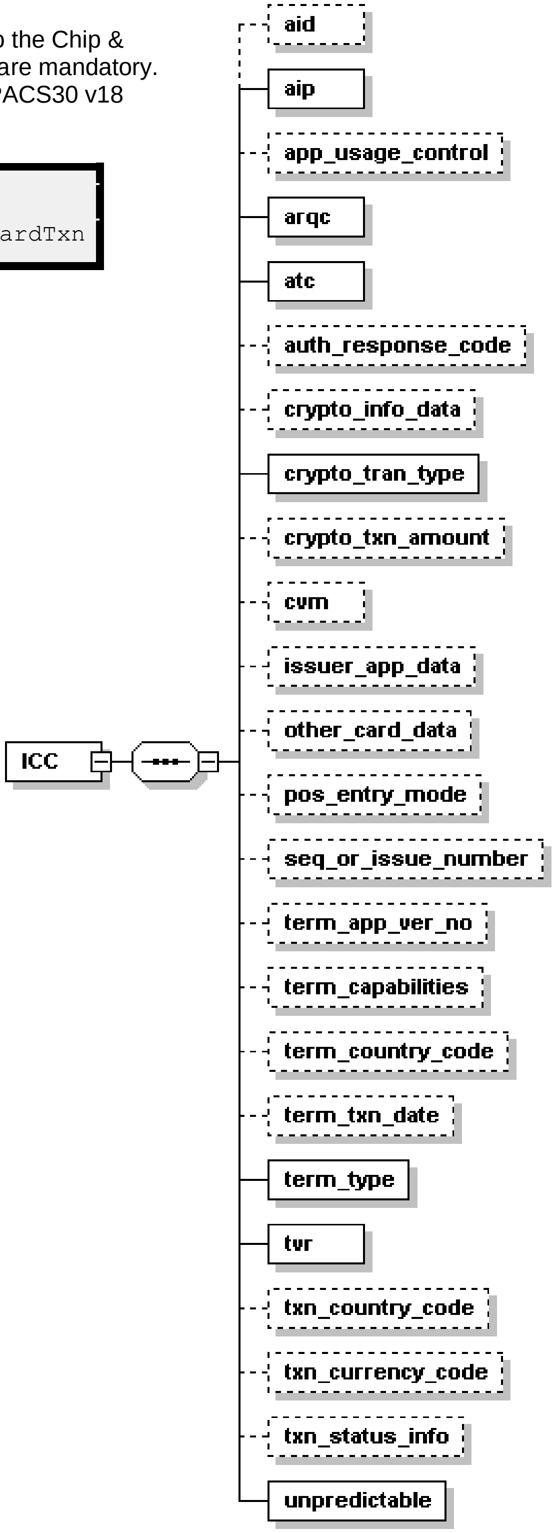
Example XML for Terminal element

```
<Terminal id="82000062">
  <terminal_capabilities ic_reader="true"
                         magnetic_stripe_reader="true"
                         manual_card_entry="true" />
  <features_capabilities pin_pad_available="true" />
  <message_capabilities downline_load_floor_limit=”false”
                        downline_load_referral=”false”
                        hold_capability=”false”
                        response_additional_data_support = “true” />
</Terminal>
```

ICC

This parent element contains the data relating to the Chip & Pin functionality. All elements within this parent are mandatory. Please refer to APACS29 v18 (page 77) and APACS30 v18 (page 73) for definitions of these elements

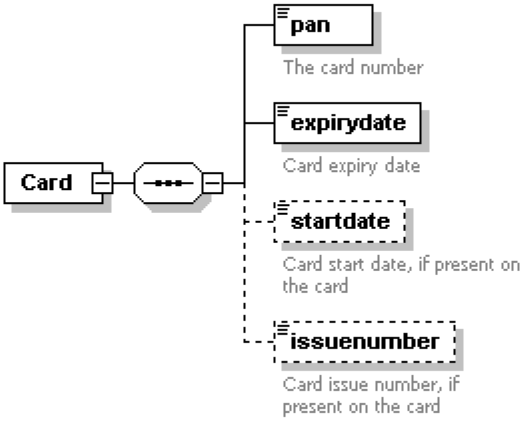
Element Name:	`ICC`
Position:	`Request.Transaction.CardTxn`



G.1.1.4. Card

This element is only required for `keyed` transactions.

Element Name: `Card`
Position: `Request.Transaction.CardTxn`


Card
pan
The card number
expirydate
Card expiry date
startdate
Card start date, if present on the card
issuenumber
Card issue number, if present on the card

Elements of Card			
Element Name	**description**	**values / limitations**	
`pan`	The card number	Must be a valid card number	R
`expiry_date`	expiry date for the card	`MM/YY` format	R
`start_date`	start date for the card	`MM/YY` format	M
`issue_number`	issue number of the card	One or two digits	M

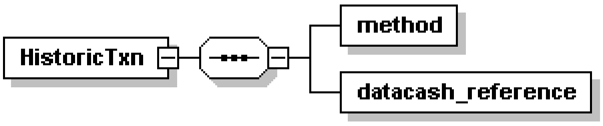
Example XML for Card element

```
<Card>
  <expirydate>01/12</expirydate>
  <pan>444433*********1</pan>
</Card>

<Card>
  <expirydate>01/06</expirydate>
  <issuenumber>1</issuenumber>
  <startdate>01/99</startdate>
  <pan>446281*********2</pan>
</Card>
```

G.1.1.5. HistoricTxn

This element should be presented for `cancel` transactions only

Element Name: `HistoricTxn`
Position: `Request.Transaction`



Elements of HistoricTxn		
Element Name	**description**	**values / limitations**
`method`	The transaction type	`cancel`
`datacash_reference`	The `datacash_reference` of the transaction to be cancelled	Must be valid transaction

Attributes of Elements in HistoricTxn			
Attribute Name	**Attribute of element**	**description**	**values / limitations**
`reversal`	`method`	Indicates whether a reversal should be attempted	`true` `false`

Example XML for HistoricTxn element

```
<HistoricTxn>
  <method reversal="true">cancel</method>
  <reference>4600200040913258</reference>
</HistoricTxn>

<HistoricTxn>
  <method>cancel</method>
  <reference>4500200040913862</reference>
</HistoricTxn>
```

G.1.2. Example XML Requests

Example XML Request for a parked transaction

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>mypasswd</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>57483573457gegrreg</merchantreference>
      <amount currency="GBP">54.00</amount>
      <capturemethod>parked</capturemethod>
    </TxnDetails>
    <CardTxn>
      <Terminal id="82000062">
        <terminal_capabilities ic_reader="true"
                  magnetic_stripe_reader="true"
                  manual_card_entry="true"/>
        <features_capabilities pin_pad_available="true"/>
      </Terminal>
      <reason_online_code>10</reason_online_code>
      <card_details type="track2_data">NDkyOTQ5ODMxMTQwMDAwMj
               0wODAzMjAxMDA5ODgwMDAwMDAwMR8wMA==</card_details>
      <ICC>
        <seq_or_issue_number>00</seq_or_issue_number>
        <crypto_tran_type>00</crypto_tran_type>
        <txn_currency_code>826</txn_currency_code>
        <term_country_code>826</term_country_code>
        <arqc>ED128CAB559EC54C</arqc>
        <aip>5C00</aip><atc>0CF3</atc>
        <unpredictable>81B72731</unpredictable>
        <tvr>0000008000</tvr>
        <issuer_app_data>06FE0A03A4A800</issuer_app_data>
        <crypto_info_data>80</crypto_info_data>
        <cvm>410302</cvm>
        <aid>A0000000031010</aid>
        <term_app_ver_no>0084</term_app_ver_no>
        <term_capabilities>7546</term_capabilities>
      </ICC>
      <method>creditcheck</method>
    </CardTxn>
  </Transaction>
</Request>
```

Example XML Request for a swiped transaction

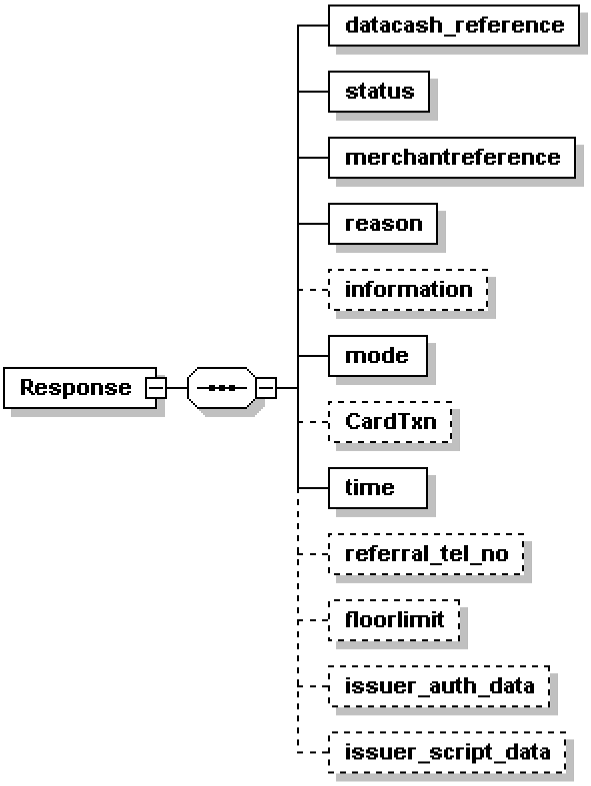
```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>mypasswd</password>
  </Authentication><Transaction>
  <TxnDetails>
    <merchantreference>924100120050503165050</merchantreference>
    <amount currency="GBP">40.01</amount>
    <capturemethod>swiped</capturemethod>
  </TxnDetails>
  <CardTxn>
    <Terminal id="82000062">
      <terminal_capabilities magnetic_stripe_reader="true"
                 manual_card_entry="true" ic_reader="true"/>
      <features_capabilities pin_pad_available="true"/>
    </Terminal>
    <card_details type="track2_data">OzQ5Mjk0OTgzMTE0MDAwMDI9MDg
                   wMzIwMTAwOTg4MDAwMDAwMDE/Pw==</card_details>
    <method>creditcheck</method>
    </CardTxn>
  </Transaction>
</Request>
```

Example XML Request for a cancellation

```
<Request>
  <Authentication>
    <client>99000001</client>
    <password>mypasswd</password>
  </Authentication>
  <Transaction>
    <HistoricTxn>
    <method reversal="true">cancel</method>
    <merchantreference>575600720050517212621</merchantreference>
    </HistoricTxn>
  </Transaction>
</Request>
```

G.1.3. Schema Elements for Response

G.1.3.1. Response



The following additional elements returned in the Response element. Please refer to section A.1.2 for descriptions of the general elements.

Additional Elements of Response	
Element Name	**description**
`referral_tel_no`	As described in the APACS30 Response
`floorlimit`	
`issuer_auth_data`	
`issuer_script_data`	
`authorising_entity`	Indicates (where available) the party who provided the authorisation. Only returned for `creditcheck` transactions. The returned value will be one of: • `card acceptor` • `acquirer` • `card scheme` • `card issuer`

Attributes of Elements in Response		
Attribute Name	**Attribute of element**	**description**

`reversal`	`reason`	Indicates the status of reversal requests

G.1.4. Example XML Responses

Example XML Response to an auth or creditcheck

```
<Response>
  <acquirer_message>AUTH CODE:100100</acquirer_message>
  <acquirer_response_code>00</acquirer_response_code>
  <CardTxn>
    <authcode>100100</authcode>
    <card_scheme>VISA</card_scheme>
    <country>United Kingdom</country>
  </CardTxn>
  <datacash_reference>…</datacash_reference>
  <floor_limit>020</floor_limit>
  <mode>…</mode>
  <reason>ACCEPTED</reason>
  <referral_telephone_number>MDIyMjQ0=</referral_telephone_number>
  <status>1</status>
  <time>…</time>
</Response>
```

Example XML Responses to cancellation Requests

```
<Response>
  <datacash_reference>…</datacash_reference>
  <information>Too much time has elapsed between the original
               txn and the reversal request.</information>
  <merchantreference>…</merchantreference>
  <mode>…</mode>
  <reason reversal='not attempted'>CANCELLED OK</reason>
  <status>1</status>
  <time>…</time>
</Response>

<Response>
  <datacash_reference>…</datacash_reference>
  <information>Reversal ACCEPTED status=1
               authcode=100100</information>
  <merchantreference>…</merchantreference>
  <mode>…</mode>
  <reason reversal="success">CANCELLED OK</reason>
  <status>1</status>
  <time>…</time>
</Response>
```

G.2. Preallocated References

Additional non-technical information about this service is available on the website:
http://www.datacash.com/services/chip_and_pin/preallocated_refs.shtml

G.2.1. Schema Elements for Request

In this section the fields required to preallocate references will be given

Preallocations
The information required to preallocate a DataCash reference is passed in two places in the schema:

- `Request`
    - `Authentication` – section A.1.1.1
    - `Transaction`
        - `TxnDetails` - contains the merchantreference, section D.2.1.1
        - `CardTxn` – contains the method allocate_reference.

Performing Transactions
The information required to use a preallocated transaction is very similar to a normal transaction:

- `Request`
    - `Authentication` – section A.1.1.1
    - `Transaction`
        - `TxnDetails` – section G.2.1.1
        - `CardTxn` – section G.1.1.2
            - `Card –` section G.1.1.4
            - `ICC` - section 0
            - `Terminal` – section G.1.1.3

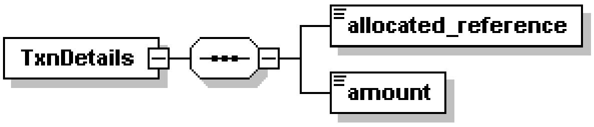
Cancellations and Reversals
The `cancel` transaction type is the same as for the normal CHP transactions:

- `Request`
    - `Authentication` – section A.1.1.1
    - `Transaction`
        - `HistoricTxn` – section G.1.1.5

G.2.1.1. TxnDetails

These elements are only required for payments made with the allocated reference – they are not required for the `allocate_reference` transactions.

Element Name:	`TxnDetails`
Position(s)	`Request.Transaction`



Elements of TxnDetails		
Element Name	**description**	**values / limitations**
`allocated_reference`	The `datacash_reference` returned from an `allocate_reference` transaction	16 digit number
`amount`	The value of the transaction	

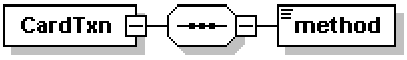
Example XML Element TxnDetails for a preallocated reference transaction

```
<TxnDetails>
  <allocated_reference>3000100012345671</allocated_reference>
  <amount>100.00</amount>
</TxnDetails>
```

G.2.1.2. CardTxn

This section shows the fields required for the `allocate_reference` transactions. Please refer to section G.1.1.2 for those required for `auth` and `creditcheck` transactions.

Element Name:	`CardTxn`
Position(s)	`Request.Transaction`



Elements of CardTxn		
Element Name	**description**	**values / limitations**
`method`	The transaction type	`allocate_reference`

Example XML CardTxn element

```
<CardTxn>
  <method>allocate_reference</method>
</CardTxn>
```

G.2.2. Example Request

Example XML Request for allocate_reference

```
<Request>
  <Transaction>
    <TxnDetails>
      <merchantreference>123402</merchantreference>
    </TxnDetails>
    <CardTxn>
      <method>allocate_reference</method>
    </CardTxn>
  </Transaction>
</Request>
```

Example XML Request

```
<Request>
  <Authentication>…</Authentication>
  <Transaction>
    <TxnDetails>
      <allocated_reference>4800200040913907</allocated_reference>
      <amount currency="GBP">54.00</amount>
      <capturemethod>parked</capturemethod>
    </TxnDetails>
    <CardTxn>
      <Terminal id="82000062">
        <terminal_capabilities ic_reader="true"
                  magnetic_stripe_reader="true"
                  manual_card_entry="true"/>
        <features_capabilities pin_pad_available="true"/>
      </Terminal>
      <card_details type="track2_data">NDkyOTQ5ODMxMTQwMDAwMj
               0wODAzMjAxMDA5ODgwMDAwMDAwMR8wMA==</card_details>
      <ICC>
        <reason_online_code>10</reason_online_code>
        <seq_or_issue_number>00</seq_or_issue_number>
        <crypto_tran_type>00</crypto_tran_type>
        <txn_currency_code>826</txn_currency_code>
        <term_country_code>826</term_country_code>
        <arqc>ED128CAB559EC54C</arqc>
        <aip>5C00</aip><atc>0CF3</atc>
        <unpredictable>81B72731</unpredictable>
        <tvr>0000008000</tvr>
        <issuer_app_data>06FE0A03A4A800</issuer_app_data>
        <crypto_info_data>80</crypto_info_data>
        <cvm>410302</cvm>
        <aid>A0000000031010</aid>
        <term_app_ver_no>0084</term_app_ver_no>
        <term_capabilities>7546</term_capabilities>
      </ICC>
      <method>creditcheck</method>
    </CardTxn>
  </Transaction>
</Request>
```

G.2.3. Schema Elements for Response

The response to transactions performed using this service contains no fields beyond the CHP response elements, covered in section G.1.3

G.2.4. Example Response

Example XML Request for an allocate_reference request

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>3000100012345671</datacash_reference>
  <merchantreference>123402</merchantreference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>1107190133</time>
</Response>
```

H. Other Services

H.1. PayPal

This service allows you to process Express Checkout and Auth and Capture PayPal transactions via your DataCash account. Successful payments can be refunded back to the customer's PayPal account. The service can be seamlessly integrated into your systems, enabling your customers and Customer Service teams to experience fast and efficient processing and management of transactions.

An introduction to this service, including the transaction types, is available on the website: http://www.datacash.com/services/paypal/index.php

H.1.1. Schema Elements for Request

In this section the required fields for each transaction type will be presented, along with example XML for those fields.

The data for the PayPal Service is passed in these distinct places in the schema:

- `Request`
  - `Authentication` - section A.1.1 of the Developers Guide
  - `Transaction`
    - `TxnDetails` - section A.1.1 of the Developers Guide, and section H.1.1.1 (PayPal documentation)
    - `PayPalTxn` – this section contains all of the details specific to the PayPal transaction, section H.1.1.1
      - `Items` – details of all items in the order – section H.1.1.3
        - `Item` – details of each item within the order – section H.1.1.4
      - `ShippingAddress` – details of the shipping address - section H.1.1.5
      - `AirlineItineraryData` – details of flights for Airline transactions – section H.1.1.6

H.1.1.1. TxnDetails

Note that most fields are only available for specific transaction types

- S – set_express_checkout
- G – get_express_checkout_details
- D – do_express_checkout_payment
- X – txn_refund, used to perform refunds of existing payments
- A – do_authorization
- C – do_capture

Element Name:	TxnDetails
Position(s)	Request.Transaction

Elements of TxnDetails								
Element Name	description	values / limitations	Required					
			S	G	D	X	A	C
amount	The value of the transaction		R	-	R	O	R	R
merchantreference	A unique reference number for each transaction	Minimum 6, maximum 30 alphanumeric characters. Must be unique	R	R	R	O	-	-

For a txn_refund, an amount may be specified if the refund is for a smaller amount than the original transaction. The value must be less than or equal to the value of the original transactions, minus any previous partial refunds.

Attributes of elements of TxnDetails						
Attribute name	Attribute of Element	description	values / limitations			
currency	amount	The currency for the payment	AUD CAD CHF CZK	DKK EUR GBP HKD	HUF JPY NOK NZD	PLN SEK SGD USD
			May not be specified for txn_refund, do_authorization or do_capture			

Example XML for TxnDetails element

```
<TxnDetails>
  <amount currency=”GBP”>19.26</amount>
  <merchantreference>123ABC</merchantreference>
</TxnDetails>

<TxnDetails>
  <merchantreference>46548tretr</merchantreference>
</TxnDetails>
```

H.1.1.2. PayPalTxn

All data specifically relating to PayPal transactions are submitted in this element and associated sub-elements. Note that most fields are only available for specific transaction types.

- S – set_express_checkout
- G – get_express_checkout_details
- D – do_express_checkout_payment
- R – txn_refund
- A – do_authorization
- C – do_capture
- V – do_void

Element Name:	PayPalTxn
Position(s)	Request.Transaction

Elements of PayPalTxn

Element Name	description	values / limitations	Required						
			S	**G**	**D**	**R**	**A**	**C**	**V**
method	The transaction type	set_express_checkout get_express_checkout_details do_express_checkout_payment txn_refund do_authorization do_capture do_void	R	R	R	R	R	R	R
return_url	URL to which the customer's browser is returned. Recommended to be the final review page on which the customer confirms the order and payment/billing agreement	URL should be URL-encoded	R	-	-	-	-	-	-
cancel_url	URL to which the customer is returned if (s)he does not approve the use of PayPal	URL should be URL-encoded	R	-	-	-	-	-	-
reference	The 16 digit datacash_reference, referring to a previous successfully processed transaction	**D** uses ref from **S** **R** uses ref from either **D** or **C** **A** uses ref from **D** **C** uses ref from **A** **V** uses ref from **A**	-	-	R	R	R	R	R
email	Email address of the buyer. Used to pre-fill the PayPal membership sign-up portion of the PayPal login page	127 single-byte characters	O	-	-	-	-	-	-
description	Description of the items the customer is purchasing		O	-	O	-	-	-	-
invnum	Your own unique invoice or tracking number. Is returned in the do_express_checkout_payment response		O	-	O	-	-	O	-
max_amount	The expected maximum total amount of the complete order, including shipping and tax	Numeric with two decimal places. Must not exceed $10,000 in any currency. Max value 9,999,999.99	O	-	-	-	-	-	-
billing_type	Type of billing agreement.		O	-	-	-	-	-	-
billing_agreement_description	Description of goods or services associated with the billing agreement		O	-	-	-	-	-	-

billing_agreement_custom	Custom annotation field for your own use		O	-	-	-	-	-	-
channel_type	Type of channel	Merchant – non-auction seller eBayItem - ebay auction	O	-	-	-	-	-	-
custom	A free-form field for your own use, such as a tracking number of other value you want PayPal to return on get_express_checkout_details and do_express_checkout_payment	256 single-byte alphanumeric characters	O	-	O	-	-	-	-
req_billing_address	Specifies whether you require the customers billing address from PayPal. This is returned the get_express_checkout_details response	0 – Billing address not required (default) 1 – Billing address required	O	-	-	-	-	-	-
req_confirmed_shipping	Indicates whether the customer's shipping address on file with PayPal must be a confirmed address	0 – need not be confirmed (default) 1 – must be confirmed	O	-	-	-	-	-	-
no_shipping	Indicates whether the shipping address fields should be displayed on the PayPal pages	0 - fields displayed (default) 1 - fields not displayed	O	-	-	-	-	-	-
override_address	Indicates whether the shipping address fields (as set by you) should be displayed on the PayPal pages, rather than the shipping address details held on file at PayPal	0 – displays details held on file (default) 1 – displays details set by you	O	-	-	-	-	-	-
payment_type	Specifies type of PayPal payment you require for the billing agreement	Any InstantOnly	O	-	-	-	-	-	-
payment_action	Specifies the type of PayPal service to use (Express Checkout or Auth and Capture)	sale (default) order order **must** be supplied during **S** for Auth and Capture txns. An order may be converted to a sale during the **D** txn.	O	-	O	-	-	-	-
completed	Indicates the final do_capture being processed against a successful do_authorization, thus completing the authorisation.	Yes	-	-	-	-	-	O	-
localecode	Locale of pages displayed by PayPal during checkout	AU - Australia DE - Germany FR - France IT - Italy GB – Great Britain ES - Spain US – United States of America (default)	O	-	-	-	-	-	-
header_style	Defines the image, border colour and background colours for the image in the header of the payment page. Header space is 750px wide by 90 px high.	See attributes table below	O	-	-	-	-	-	-
page_style	Sets the Custom Payment Page Style of the payment pages. This corresponds to the HTML variable page_style for customising payment pages	Value must be equal to the Page Style Name as chosen by you when adding/editing the page style from the My Profile -> Profile	O	-	-	-	-	-	-

		subtab of your PayPal Account							
`solution_type`	Type of checkout flow	`Sole` – for Express Checkout auctions `Mark` – normal Express Checkout	O	-	-	-	-	-	-
`token`	Time stamped token – token expires after three hours Normally this is generated by PayPal and does not need to be supplied.	20 single-byte characters max	O	-	-	-	-	-	-
`buttonsource`	An identification code for use by third party applications to identify transactions	32 single-byte alphanumeric characters	-	-	O	-	-	-	-
`notify_url`	Your URL for receiving Instant Payment Notification (IPN) about this transaction	2048 single-byte alphanumeric characters	-	-	O	-	-	-	-
`item_total`	Sum of cost of all items in this order	Numeric with two decimal places. Must not exceed $10,000 in any currency. Max value 9,999,999.99	-	-	O	-	-	-	-
`shipping_total`	Total shipping cost for this order		-	-	O	-	-	-	-
`handling_total`	Total handling costs for this order		-	-	O	-	-	-	-
`tax_total`	Sum of tax for all items in this order		-	-	O	-	-	-	-
`note`	Custom memo about the refund	255 single-byte alphanumeric characters	-	-	-	O	-	O	O
`soft_descriptor`	Per transaction description of the payment that is passed to the consumer's credit card	22 single-byte characters comprised of: Alphanumerics - (dash) * (asterisk) . (period) {space}	-	-	-	-	-	O	-
`Items`	Please refer to section H.1.1.3								
`ShippingAddress`	Please refer to section H.1.1.5								
`AirlineItineraryData`	Please refer to section H.1.1.6		-	-	O	-	-	O	-

Attributes for Elements of PayPalTxn			
Attribute Name	**Attribute of element**	**Value / limitations**	**Required**
img	header_style	URL of the image to appear on the top left of the payment page. Max image size 750px wide by 90px high. Recommended to be HTTPS link. 127 single-byte characters max, URL-encoded.	O
bordercolor	header_style	Sets the colour of the 2px border around the header of the payment page. Six character HTML hexadecimal colour code in ASCII	O
bgcolor	header_style	Background colour for the payment page header. Six character HTML hexadecimal colour code in ASCII	O
stylename	page_style	Used to select the style of checkout pages. Styles are configurable in PayPal account settings.	O
bgcolor	page_style	Background colour for payment page. Six character HTML hexadecimal colour code in ASCII	O

Example XML for PayPal elements for set_express_checkout

```
<PayPalTxn>
  <method>set_express_checkout</method>
  <return_url>https://www.example.com/myshoppingcard?action=
    complete&amp;sale_id=xyz123</return_url>
  <cancel_url>https://www.example.com/myshoppingcard?action=
    cancel&amp;sale_id=xyz123</cancel_url>
</PayPalTxn>

<PayPalTxn>
  <method>set_express_checkout</method>
  <return_url>https://www.example.com/myshoppingcard?action=
    complete&amp;sale_id=xyz123</return_url>
  <cancel_url>https://www.example.com/myshoppingcard?action=
    cancel&amp;sale_id=xyz123</cancel_url>
  <email>homer@example.com<email>
  <max_amount>1000.00</max_amount>
  <description>Twenty NeverFail(tm) Widgets</description>
  <custom>UPS_ID=12345678</custom>
  <invnum>abc123</invnum>
  <req_confirmed_shipping>0</req_confirmed_shipping>
  <no_shipping>0</no_shipping>
  <override_address>0</override_address>
  <localecode>GB</localecode>
  <header_style img="https://www.example.com/header.png"
    bordercolor="black" bgcolor="red" />
  <page_style stylename="My Style" bgcolor="white" />
  <ShippingAddress>…</ShippingAddress>
</PayPalTxn>

<PayPalTxn>
  <method>set_express_checkout</method>
  <return_url>https://www.example.com/myshoppingcard?action=
    complete&amp;sale_id=xyz123</return_url>
  <cancel_url>https://www.example.com/myshoppingcard?action=
    cancel&amp;sale_id=xyz123</cancel_url>
  <token>EC-4VL78907RS990801R</token>
  <payment_type>InstantOnly</payment_type>
</PayPalTxn>
```

Example XML for PayPal elements for get_express_checkout_details

```
<PayPalTxn>
  <method>get_express_checkout_details</method>
  <reference>4000900012345671</reference>
</PayPalTxn>
```

Example XML for PayPal elements for do_express_checkout_payment

```
<PayPalTxn>
  <method>do_express_checkout_payment</method>
  <reference>4000900012345671</reference>
</PayPalTxn>

<PayPalTxn>
  <method>do_express_checkout_payment</method>
  <reference>4000900012345671</reference>
  <description>Twenty NeverFail(tm) Widgets</description>
  <custom>UPS_ID=12345678</custom>
  <invnum>abc123</invnum>
  <buttonsource>foobar</buttonsource>
  <notify_url>http://www.example.com/mynotifyurl?id=abc123
    </notify_url>
  <item_total>85.00</item_total>
  <shipping_total>10.00</shipping_total>
  <handling_total>5.00</handling_total>
  <tax_total>11.97</tax_total>
  <Items>…</Items>
  <ShippingAddress>…</ShippingAddress>
</PayPalTxn>

<PayPalTxn>
  <method>do_express_checkout_payment</method>
  <reference>4000900012345671</reference>
  <Items>…</Items>
</PayPalTxn>
```

Example XML for PayPal elements for txn_refund

```
<PayPalTxn>
  <method>txn_refund</method>
  <reference>4000900012345671</reference>
</PayPalTxn>

<PayPalTxn>
  <method>txn_refund</method>
  <reference>4000900012345671</reference>
  <note>Refund of half the order, as promised</note>
</PayPalTxn>
```

Example XML for PayPal elements for do_authorization

```
<PayPalTxn>
  <method>do_authorization</method>
  <reference>4000900012345671</reference>
</PayPalTxn>
```

Example XML for PayPal elements for do_capture

```
<PayPalTxn>
  <method>do_capture</method>
  <reference>4000900012345671</reference>
</PayPalTxn>

<PayPalTxn>
  <method>do_capture</method>
  <reference>4000900012345671</reference>
  <invnum>inv1234</inv1234>
  <note>Finally, I have captured some funds</note>
  <soft_descriptor>FooCorp LLC</soft_descriptor>
</PayPalTxn>


<PayPalTxn>
  <method>do_capture</method>
  <reference>4000900012345671</reference>
  <completed>yes</completed>
  <invnum>inv1234</inv1234>
  <note>This is the last capture for this
     authorization</note>
  <soft_descriptor>FooCorp LLC</soft_descriptor>
</PayPalTxn>
```

Example XML for PayPal elements for do_void

```
<PayPalTxn>
  <method>do_void</method>
  <reference>4000900012345671</reference>
</PayPalTxn>

<PayPalTxn>
  <method>do_void</method>
  <reference>4000900012345671</reference>
  <note>Excess funds released<note>
</PayPalTxn>
```

H.1.1.3. Items

The Items element is optional. It contains an Item child element for each individual product or service purchased.

Element Name:	`Items`
Position(s)	`Request.Transaction.PayPalTxn`

Elements of Items

Element Name	description	values / limitations	Required			
			S	G	D	R
`Item`	See section H.1.1.4		-	-	O	-

Example XML for Items element

```
<Items>
  <Item id=”0”>…</Item>
  <Item id=”1”>…</Item>
  <Item id=”2”>…</Item>
  <Item id=”3”>…</Item>
</Items>
```

H.1.1.4. Item

The Item elements contain details of each product within the order

Element Name:	`Item`
Position(s)	`Request.Transaction.PayPalTxn.Items`

Elements of Item						
Element Name	description	values / limitations	Required			
			S	G	D	R
`ebay_item_number`	Auction item number	765 single-byte characters	-	-	O	-
`ebay_item_auction_txn_id`	Auction transaction identification number	225 single-byte characters	-	-	O	-
`ebay_item_order_id`	Auction identification number	64 single-byte characters	-	-	O	-
`name`	The item name	127 single-byte characters	-	-	O	-
`number`	The item number		-	-	O	-
`quantity`	The item quantity	Positive integer	-	-	O	-
`amount`	Cost of item	Must not exceed $10,000 in any currency. Max value 9,999,999.99	-	-	O	-
`taxamt`	Item sales tax		-	-	O	-

Attributes for Elements of Item			
Attribute Name	Attribute of element	Value / limitations	Required
`id`	`Item`	Must start from 0 and be contiguous	R

Example XML for Item elements for do_express_checkout_payment

```
<Item id="0">
  <name>Widgets</name>
  <number>1230212-A</number>
  <quantity>20</quantity>
  <amount>73.03</amount>
  <taxamt>11.97</taxamt>
</Item>
```

Example XML for Item elements for do_express_checkout_payment

```
<Item id="0">
  <ebay_item_number>9988775544</ebay_item_number>
  <ebay_item_auction_txn_id>123456123</ebay_item_auction_txn_id>
  <ebay_item_order_id>abc123def890</ebay_item_order_id>
</Item>
```

H.1.1.5. ShippingAddress

This parent element is used to present information about the shipping address. While the parent element itself is optional, if it is sent certain elements must be provided

Element Name:	`ShippingAddress`
Position(s)	`Request.Transaction.PayPalTxn`

Elements of ShippingAddress

Element Name	description	values / limitations	Required
`name`	The person's name associated with the shipping address	32 single-byte characters max	R
`street_address1`	The first line of the address	100 single-byte characters max	R
`street_address2`	The second line of the address	100 single-byte characters max	O
`city`	The name of the city	40 single-byte characters max	R
`region`	The state, province or region	40 single-byte characters max	R for US addresses only
`country_code`	The country code, as defined in ISO 3166-1	2 character	R
`postcode`	UK postcode, US ZIP code or other country-specific postal code	20 single-byte characters max	R
`telephone_number`	Phone number	20 single-byte characters max	0

Example XML for ShippingAddress elements

```
<ShippingAddress>
  <name>Joe Bloggs</name>
  <street_address1>3 Fish Street</ street_address1>
  <street_address2>Castle Street</ street_address2>
  <city>Hull</city>
  <country_code>GB</country_code>
  <postcode>HU1 1AA</postcode>
  <telephone_number>01234 345 6789</telephone_number>
</ShippingAddress>

<ShippingAddress>
  <name>John Smith</name>
  <street_address1>144 Main Street</ street_address1>
  <city>San Jose</city>
  <region>CA</region>
  <country_code>US</country_code>
  <postcode>99221</postcode>
</ShippingAddress>
```

H.1.1.6. AirlineItineraryData

This parent element is used to present flight data for Airline transactions. While this element itself is optional, if it is sent then certain elements must be provided.

This element is supported for the following transaction types, depending on the value of the `payment_action` element.

- `D – do_express_checkout_payment`
- `C – do_capture`

Where the `payment_action` of a transaction is `sale`, Airline data should be provided in the `do_express_checkout_payment` request.

Where the `payment_action` is `order`, Airline Data should be provided in the `do_capture` request.

Element Name:	`AirlineItineraryData`
Position(s)	`Request.Transaction.PayPalTxn`

Elements of AirlineItineraryDetails

Element Name	Description	Values / Limitations	Required	
			D	C
`passenger_name`	Name of the passenger	25 single-byte characters max	R	R
`issue_date`	Date of issue recorded in the airline system. In case of multiple issuances of the same ticket, use the last ticket date	`YYYYMMDD` format	O	O
`travel_agency_name`	Name of the travel agency issuing the ticket. In an integration by an airline, this is the airline name	25 single-byte characters max	O	O
`travel_agency_code`	The travel agency code. In an integration by the airline, this is the airline code from the official *Airline Guide* or its equivalent	8 single-byte characters max	O	O
`ticket_number`	The ticket number. If multiple tickets are purchased with one transaction, you should provide the primary ticket number	16 single-byte characters max	R	R
`issuing_carrier_code`	Airline code for the airline issuing the ticket. Obtain the airline code from the official *Airline Guide* or its equivalent	4 single-byte characters max	R	R
`customer_code`	A code that the cardholder supplied to you. Can be used for passing in the frequent flyer number of the customer	17 single-byte characters max	O	O
`total_fare`	Total fare for all legs on this ticket, excluding taxes and fees	Numeric with two decimal places. Must not exceed $10,000 in any currency. Max value 9,999,999.99	O	O
`total_taxes`	Total taxes for all legs on the ticket		O	O
`total_fee`	Total fee for all legs on the ticket		O	O
`restricted_ticket`	Indicated whether the ticket is restricted (refundable)	`0` – Ticket is not restricted `1` – Ticket is restricted	R	R

`clearing_sequence`	Numeric code to identify each clearing record message in cases where multiple clearing messages are allowed per authorised transaction. Applicable to multiple captures against an authorisation. In the case of single capture against an authorisation, the value should be 1	Positive integer	R	R
`clearing_count`	Numeric code to identify each clearing record message in those cases where multiple clearing messages are allowed per authorized transaction. Applicable to multiple captures against an authorization. In the case of single capture against an authorization, the value should be 1	Positive integer	R	R
`FlightDetails`	Please refer to section H.1.1.7		R	R

Example XML for AirlineItineraryData elements

```
<AirlineItineraryData>
  <passenger_name>Joseph Bloggs</passenger_name>
  <issue_date>20090701</issue_date>
  <travel_agency_name>Super Agents</travel_agency_name>
  <travel_agency_code>SUPER1</travel_agency_code>
  <ticket_number>BLAH1234567890</ticket_number>
  <issuing_carrier_code>ABCD</issuing_carrier_code>
  <customer_code>JBLOGGS09713</customer_code>
  <total_fare>85.00</total_fare>
  <total_taxes>10.00</total_taxes>
  <total_fee>5.00</total_fee>
  <restricted_ticket>0</restricted_ticket>
  <clearing_sequence>1</clearing_sequence>
  <clearing_count>1</clearing_count>
  <FlightDetails leg_id=”0”>
    ...
  </FlightDetails>
  <FlightDetails leg_id=”1”>
    ...
  </FlightDetails>
</AirlineItineraryData>

<AirlineItineraryData>
  <passenger_name>Joseph Bloggs</passenger_name>
  <ticket_number>BLAH1234567890</ticket_number>
  <issuing_carrier_code>ABCD</issuing_carrier_code>
  <restricted_ticket>0</restricted_ticket>
  <clearing_sequence>1</clearing_sequence>
  <clearing_count>1</clearing_count>
  <FlightDetails leg_id=”0”>
    ...
  </FlightDetails>
</AirlineItineraryData>
```

H.1.1.7. FlightDetails

The FlightDetails elements contain details of each leg of the trip.

Element Name:	`FlightDetails`
Position(s)	`Request.Transaction.PayPalTxn.AirlineItineraryData`

Elements of FlightDetails

Element Name	Description	Values / Limitations	Required	
			D	C
`conjunction_ticket`	Ticket issued to a passenger in conjunction with another ticket that constitutes a single contract of carriage	14 single-byte characters max	O	O
`exchange_ticket`	New ticket number that is issued when a ticket is exchanged	15 single-byte characters max	O	O
`coupon_number`	The coupon number associated with this leg of the trip. A ticket can contain several legs of travel, and each leg of travel requires a separate coupon	1 single-byte character	O	O
`service_class`	The type of service; for example, first class or coach. Obtain the service class from the official *Airline Guide* or equivalent	2 single-byte characters	R	R
`travel_date`	The date of travel in local time at the departure airport	`YYYYMMDD` format	R	R
`carrier_code`	Standard abbreviation for airline carrier. Obtain the code from the official *Airline Guide* or its equivalent	2 single-byte characters	R	R
`stopover_code`	A code indicating a non-direct flight or route on the same ticket number	`0` – Stopover not permitted `1` – Stopover permitted	R	R
`departure_airport_code`	The departure airport code. Obtain the code from the official *Airline Guide* or its equivalent	5 single-byte characters max	R	R
`arrival_airport_code`	The arrival airport code. Obtain the code from the official *Airline Guide* or its equivalent	5 single-byte characters max	R	R
`flight_number`	The flight number assigned by the airline carrier	5 single-byte characters max	R	R
`departure_time`	The departure time in local time at the departure airport	`HH:MM` format, between `00:00` and `23:59`	R	R
`arrival_time`	The arrival time in local time at the arrival airport	`HH:MM` format, between `00:00` and `23:59`	O	O
`fare_basis_code`	The alphanumeric code that carriers assign to a particular ticket type, such as business class, discounted, or non-refundable	15 single-byte characters max	R	R

`fare`	Amount of the ticket for this leg of the trip excluding taxes and fees	Numeric with two decimal places. Must not exceed $10,000 in any currency. Max value 9,999,999.99	O	O
`taxes`	Amount of the taxes for this leg of the trip		O	O
`fee`	Fee amount for this leg of the trip		O	O
`endorsement_or_restrictions`	An endorsement can be an agency-added notation or a mandatory government required notation such as value added tax. A restriction is a limitation based on the type of fare such as a ticket with a non-refundable or 3-day minimum stay	20 single-byte characters max	O	O

Attributes for Elements of FlightDetails			
Attribute Name	**Attribute of element**	**Value / limitations**	**Required**
`leg_id`	`FlightDetails`	Must start from 0 and be contiguous	R

Example XML for FlightDetails elements

```
<FlightDetails leg_id="0">
  <conjunction_ticket>CONTKT12345678</conjunction_ticket>
  <exchange_ticket>EXTKT123456789</exchange_ticket>
  <coupon_number>X</coupon_number>
  <service_class>A1</service_class>
  <travel_date>20090801</travel_date>
  <carrier_code>AB</carrier_code>
  <stopover_code>0</stopover_code>
  <departure_airport_code>EDI</departure_airport_code>
  <arrival_airport_code>LHR</arrival_airport_code>
  <flight_number>FOO12</flight_number>
  <departure_time>12:34</departure_time>
  <arrival_time>13:37</arrival_time>
  <fare_basis_code>CHEAP1</fare_basis_code>
  <fare>15.00</fare>
  <taxes>5.00</taxes>
  <fee>2.00</fee>
  <endorsement_or_restrictions>non-refundable</endorsement_or_restrictions>
</FlightDetails>

<FlightDetails leg_id="1">
  <service_class>A1</service_class>
  <travel_date>20090801</travel_date>
  <carrier_code>AB</carrier_code>
  <stopover_code>0</stopover_code>
  <departure_airport_code>EDI</departure_airport_code>
  <arrival_airport_code>LHR</arrival_airport_code>
  <flight_number>FOO12</flight_number>
  <departure_time>12:34</departure_time>
  <fare_basis_code>CHEAP1</fare_basis_code>
</FlightDetails>
```

H.1.2. XML Example Requests

This section provides full XML examples for each transaction type

H.1.2.1. set_express_checkout

Example XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount currency="GBP">100.00</amount>
      <merchantreference>12345601</merchantreference>
    </TxnDetails>
    <PayPalTxn>
      <method>set_express_checkout</method>
      <return_url>https://www.example.com/myshoppingcard?
        action=complete&amp;sale_id=xyz123</return_url>
      <cancel_url>https://www.example.com/myshoppingcard?
        action=cancel&amp;sale_id=xyz123</cancel_url>
      <email>homer@example.com<email>
      <max_amount>1000.00</max_amount>
      <description>Twenty NeverFail(tm) Widgets</description>
      <custom>UPS_ID=12345678</custom>
      <invnum>abc123</invnum>
      <req_confirmed_shipping>0</req_confirmed_shipping>
      <no_shipping>0</no_shipping>
      <override_address>0</override_address>
      <localecode>GB</localecode>
      <header_style img="https://www.example.com/header.png"
        bordercolor="black" bgcolor="red" />
      <page_style stylename="My Style" bgcolor="white" />
      <ShippingAddress>
        <name>Test User</name>
        <street_address1>1 Main Terrace</street_address1>
        <street_address2>Highgate</street_address2>
        <city>Wolverhampton</city>
        <region>West Midlands</region>
        <country_code>GB</country_code>
        <postcode>E12 4LQ</postcode>
        <telephone_number>0121 231 3122</telephone_number>
      </ShippingAddress>
    </PayPalTxn>
  </Transaction>
</Request>
```

Example XML for Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount currency="GBP">100.00</amount>
      <merchantreference>12345601</merchantreference>
    </TxnDetails>
    <PayPalTxn>
      <method>set_express_checkout</method>
      <return_url>https://www.example.com/myshoppingcard?
        action=complete&amp;sale_id=xyz123</return_url>
      <cancel_url>https://www.example.com/myshoppingcard?
        action=cancel&amp;sale_id=xyz123</cancel_url>
      <header_style img="https://www.example.com/header.png"
        bordercolor="black" bgcolor="red" />
      <page_style stylename="My Style" bgcolor="white" />
    </PayPalTxn>
  </Transaction>
</Request>
```

Example XML for Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount currency="GBP">100.00</amount>
      <merchantreference>12345601</merchantreference>
    </TxnDetails>
    <PayPalTxn>
      <method>set_express_checkout</method>
      <return_url>https://www.example.com/myshoppingcard?
        action=complete&amp;sale_id=xyz123</return_url>
      <cancel_url>https://www.example.com/myshoppingcard?
        action=cancel&amp;sale_id=xyz123</cancel_url>
      <channel_type>Merchant</channel_type>
      <solution_type>Mark</solution_type>
      <billing_type>None</billing_type>
      <billing_agreement_description>Delivery by 10pm
         Tuesday</billing_agreement_description>
      <billing_agreement_custom>Remember to tell
         Bob</billing_agreement_custom>
      <payment_type>InstantOnly</payment_type>
    </PayPalTxn>
  </Transaction>
</Request>
```

H.1.2.2. get_express_checkout_details

Example XML for Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>46548tretr</merchantreference>
    </TxnDetails>
    <PayPalTxn>
      <method>get_express_checkout_details</method>
      <reference>4000900012345671</reference>
    </PayPalTxn>
  </Transaction>
</Request>
```

H.1.2.3. do_express_checkout_payment

Example XML for Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount currency="GBP">100.00</amount>
      <merchantreference>12345601</merchantreference>
    </TxnDetails>
    <PayPalTxn>
      <method>do_express_checkout</method>
      <reference>4000900012345671</reference>
      <Items>
        <Item id="0">
          <ebay_item_number>9988775544</ebay_item_number>
          <ebay_item_auction_txn_id>123456123
             </ebay_item_auction_txn_id>
          <ebay_item_order_id>abc123def890</ebay_item_order_id>
        </Item>
      </Items>
      <ShippingAddress>
        <name>Test User</name>
        <street_address1>1 Main Terrace</street_address1>
        <city>Wolverhampton</city>
        <country_code>GB</country_code>
        <postcode>E12 4LQ</postcode>
      </ShippingAddress>
    </PayPalTxn>
  </Transaction>
</Request>
```

Example XML for Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount currency="GBP">100.00</amount>
      <merchantreference>12345601</merchantreference>
    </TxnDetails>
    <PayPalTxn>
      <method>do_express_checkout_payment</method>
      <reference>4000900012345671</reference>
    </PayPalTxn>
  </Transaction>
</Request>
```

Example XML for Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount currency="GBP">100.00</amount>
      <merchantreference>12345601</merchantreference>
    </TxnDetails>
    <PayPalTxn>
      <method>do_express_checkout_payment</method>
      <reference>4000900012345671</reference>
      <description>Twenty NeverFail(tm) Widgets</description>
      <custom>UPS_ID=12345678</custom>
      <invnum>abc123</invnum>
      <buttonsource>foobar</buttonsource>
      <notify_url>http://www.example.com/mynotifyurl?id=
        abc123</notify_url>
      <item_total>85.00</item_total>
      <shipping_total>10.00</shipping_total>
      <handling_total>5.00</handling_total>
      <tax_total>11.97</tax_total>
    </PayPalTxn>
  </Transaction>
</Request>
```

Example XML for Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount currency="GBP">100.00</amount>
      <merchantreference>12345601</merchantreference>
    </TxnDetails>
    <PayPalTxn>
      <method>do_express_checkout_payment</method>
      <reference>4000900012345671</reference>
      <AirlineItineraryData>
        <passenger_name>Joseph Bloggs</passenger_name>
        <ticket_number>BLAH1234567890</ticket_number>
        <issuing_carrier_code>ABCD</issuing_carrier_code>
        <restricted_ticket>0</restricted_ticket>
        <clearing_sequence>1</clearing_sequence>
        <clearing_count>1</clearing_count>
        <FlightDetails leg_id=”0”>
          <service_class>A1</service_class>
          <travel_date>20090801</travel_date>
          <carrier_code>AB</carrier_code>
          <stopover_code>0</stopover_code>
          <departure_airport_code>EDI</departure_airport_code>
          <arrival_airport_code>LHR</arrival_airport_code>
          <flight_number>FOO12</flight_number>
          <departure_time>12:34</departure_time>
          <fare_basis_code>CHEAP1</fare_basis_code>
        </FlightDetails>
      </AirlineItineraryData>
    </PayPalTxn>
  </Transaction>
</Request>
```

H.1.2.4. txn_refund

Example XML for a full refund

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <PayPalTxn>
      <method>txn_refund</method>
      <reference>4000900012345671</reference>
    </PayPalTxn>
    <TxnDetails>
      <merchantreference>4655eubtr</merchantreference>
    </TxnDetails>
  </Transaction>
</Request>
```

Example XML for a partial refund

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount>19.00</amount>
      <merchantreference>45gy7byrty</merchantreference>
    </TxnDetails>
    <PayPalTxn>
      <method>txn_refund</method>
      <reference>4000900012345671</reference>
      <note>Here's the refund I promised</note>
    </PayPalTxn>
  </Transaction>
</Request>
```

H.1.2.5. do_authorization

Example XML for do_authorization

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <amount>100.00</amount>
    </TxnDetails>
    <PayPalTxn>
      <method>do_authorization</method>
      <reference>4000900012345671</reference>
    </PayPalTxn>
  </Transaction>
</Request>
```

H.1.2.6. do_capture

Example XML for do_capture

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>  <Transaction>
    <TxnDetails>
      <amount>100.00</amount>
    </TxnDetails>
    <PayPalTxn>
      <method>do_capture</method>
      <reference>4000900012345671</reference>
    </PayPalTxn>
  </Transaction>
</Request>
```

Example XML for do_capture

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
 <Transaction>
    <TxnDetails>
      <amount>100.00</amount>
    </TxnDetails>
    <PayPalTxn>
      <method>do_capture</method>
      <reference>4000900012345671</reference>
      <completed>yes</completed>
      <invnum>inv1234</inv1234>
      <note>Finally, I have captured some funds<note>
      <soft_descriptor>FooCorp LLC<soft_descriptor>
    </PayPalTxn>
  </Transaction>
</Request>
```

H.1.2.7. do_void

Example XML for do_void

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
 <Transaction>
   <PayPalTxn>
     <method>do_void</method>
     <reference>4000900012345671</reference>
   </PayPalTxn>
 </Transaction>
</Request>
```

Example XML for do_void

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>
 <Transaction>
   <PayPalTxn>
     <method>do_void</method>
     <reference>4000900012345671</reference>
     <note>Excess funds released<note>
   </PayPalTxn>
 </Transaction>
</Request>
```

H.1.3. Schema Elements for Response

The following section will highlight elements which are specific to PayPal transactions. In addition to these, some general elements will be returned. Please refer to section A1.2 of the Developers Guide for descriptions of these elements.

H.1.3.1. PayPalTxn

The PayPalTxn element returns all of the details which are specific to the PayPal Service. It should be noted that this document shows all possible fields which may be returned by PayPal. The fields that are actually returned will depend upon the data provided by your customer to PayPal, plus how your own merchant account at PayPal is set up. The following key will be used:

- R – always returned
- A – returned if available

Many fields are only available for specific transaction types

- S – set_express_checkout
- G – get_express_checkout_details
- D – do_express_checkout_payment
- R – txn_refund
- A – do_authorization
- C – do_capture
- V – do_void
- E – error messages

Element Name:	PayPalTxn
Position(s)	Response.Transaction

Elements of PayPalTxn										
Element Name	**description**	**values / limitations**	**Provided (if available)?**							
			S	**G**	**D**	**R**	**A**	**C**	**V**	**E**
ack		Success SuccessWithWarning Failure FailureWithWarning Warning	A	A	A	A	A	A	A	A
addressid			-	A	-	-	-	-	-	-
addressstatus	Status of street address on file with PayPal	Confirmed None Unconfirmed	-	A	-	-	-	-	-	-
amt	The final amount charged, including any shipping and taxes from your PayPal Merchant Profile	Must not exceed $10,000 in any currency. Max value 9,999,999.99	-	-	A	-	A	A	A	-
Billingagreementacceptedstatus	Whether or not the customer accepted the billing agreement	Yes	-	A	-	-	-	-	-	-
billingagreementid	The value of the Billing Agreement ID returned from get_express_checkout_details		-	-	A	-	-	-	-	-
build	A string representing the specific		A	A	A	A	A	A	A	A

	software build that processes the request and generates the response									
business	Payer’s business name	127 single-byte characters	-	A	-	-	-	-	-	-
correlationid	A string that identifies the exact API request that produced this response	Max 41 characters	A	A	A	A	-	-	-	A
countrycode	Payer’s country of residence	As defined in ISO 3166-1	-	A	-	-	-	-	-	-
currencycode	The currency of the transaction		-	-	A	-	-	-	-	-
custom	As section H.1.1.1		-	A	-	-	-	-	-	-
email	As section H.1.1.1		-	A	-	-	-	-	-	-
Errors	Returned for error and warning transactions. Please refer to section H.1.3.2									
exchangerate	Exchange rate if a currency conversion occurred. Relevant only if you are billing in the customer’s non-primary currency. If the customer chooses to pay with a currency other than the primary currency, the conversion occurs in the customer’s account	Decimal. Max size 17 characters including decimal point	-	-	A	-	-	A	-	-
feeamt	PayPal fee amount charged for the transaction	Must not exceed $10,000 in any currency. Max value 9,999,999.99	-	-	A	-	-	A	-	-
feerefundamount	Transaction fee refunded to original recipient of payment		-	-	-	A	-	-	-	-
firstname	Payer’s first name	25 single-byte characters	-	A	-	-	-	-	-	-
grossrefundamt	Amount of money refunded to original payer	Must not exceed $10,000 in any currency. Max value 9,999,999.99	-	-	-	A	-	-	-	-
invnum	As section H.1.1.1		-	A	-	-	-	-	-	-
lastname	Payer’s last name	25 single-byte characters	-	A	-	-	-	-	-	-
middlename	Payer’s middle name	25 single-byte characters	-	A	-	-	-	-	-	-
netrefundamt	Amount subtracted from PayPal balance of original recipient payment to make this refund	Must not exceed $10,000 in any currency. Max value 9,999,999.99	-	-	-	A	-	-	-	-
ordertime	Time/date stamp of the payment, in ISO 8601 (UTC/Zulu time)	YYYY-MM-DDTHH:MM:SSZ	-	-	A	-	-	A	-	-
payerid	Unique PayPal customer account identification number	13 single-byte characters	-	A	-	-	-	-	-	-
payerstatus	The status of the payer	verified unverified	-	A	-	-	-	-	-	-
paymentstatus	Status of the payment	See table below for values	-	-	A	-	-	A	-	-
paymenttype	Indicates whether the payment is instant or delayed	none echeck instant	-	-	A	-	-	A	-	-
pendingreason	The reason the payment is pending	See table below for possible values	-	-	A	-	-	-	-	-
phonenum	Payer’s contact telephone number		-	A	-	-	-	-	-	-
reasoncode	The reason for the reversal, if transactiontype is reversal	See table below	-	-	A	-	-	-	-	-

refundtransactionid	Unique transaction ID for the refund	17 single-byte characters	-	-	-	A	-	-	-	-
salutation	Payer’s salutation	20 single-byte characters	-	A	-	-	-	-	-	-
settleamt	Amount deposited in your PayPal account after currency conversion	Must not exceed $10,000 in any currency. Max value 9,999,999.99	-	-	A	-	-	A	-	-
ShippingAddress	If returned, this data will be in the same format as described in section H.1.1.5									
suffix	Payer’s suffix	12 single-byte characters	-	A	-	-	-	-	-	-
taxamt	Tax charged on the transaction	Must not exceed $10,000 in any currency. Max value 9,999,999.99	-	-	A	-	-	A	-	-
timestamp	Date/time of response in ISO 8601 (UTC/Zulu time)	YYYY-MM-DDTHH:MM:SSZ	A	A	A	A	A	A	A	A
token	A timestamped token by which you identify to PayPal that you are processing this payment with Express Checkout. Token expires after three hours	20 single-byte characters	R	A	A	-	-	-	-	-
transactionid	Unique transaction ID for the payment	19 single-byte characters	-	-	A	-	A	A	A	-
transactiontype	The type of transaction	express-checkout reversal	-	-	A	-	-	A	-	-
version	PayPal API service version number		A	A	A	A	A	A	A	A
authorizationid	Value of transactionid returned in response of do_authorization	19 single-byte characters	-	-	-	-	-	A	-	-
parenttransactionid	Value of transactionid returned in response of do_authorization	19 single-byte characters	-	-	-	-	-	A	-	-
receipt	Receipt identification number	19 single-byte characters	-	-	-	-	-	A	-	-

Values for paymentstatus	
value	**meaning**
Completed	The payment has been completed, and the funds have been added successfully to your account balance.
Pending	The payment is pending. See the pendingreason element for more information.

Values for pendingreason

value	meaning
`none`	No pending reason
`address`	The payment is pending because your customer did not include a confirmed shipping address and your Payment Receiving Preferences is set such that you want to manually accept or deny each of these payments. To change your preference, go to the `Preferences` section of your `Profile`.
`echeck`	The payment is pending because it was made by an eCheck that has not yet cleared.
`intl`	The payment is pending because you hold a non-U.S. account and do not have a withdrawal mechanism. You must manually accept or deny this payment from your `Account Overview`.
`multi-currency`	You do not have a balance in the currency sent, and you do not have your `Payment Receiving Preferences` set to automatically convert and accept this payment. You must manually accept or deny this payment.
`verify`	The payment is pending because you are not yet verified. You must verify your account before you can accept this payment.
`other`	The payment is pending for a reason other than those listed above. For more information, contact PayPal customer service.

Values for reasoncode

value	meaning
`none`	No reason code
`chargeback`	A reversal has occurred on this transaction due to a chargeback by your customer.
`guarentee`	A reversal has occurred on this transaction due to your customer triggering a money-back guarantee.
`buyer-complaint`	A reversal has occurred on this transaction due to a complaint about the transaction from your customer.
`refund`	A reversal has occurred on this transaction because you have given the customer a refund.
`other`	A reversal has occurred on this transaction due to a reason not listed above.

Example XML for PayPal elements, for set_express_checkout

```
<PayPalTxn>
 <ack>Success</ack>
 <build>1.0006</build>
 <timestamp>2007-10-03T13:44:56Z</timestamp>
 <token>EC-4VL78907RS990801R</token>
 <version>2.300000</version>
</PayPalTxn>
```

Example XML for PayPal elements, for get_express_checkout_details

```
<PayPalTxn>
    <ack>Success</ack>
    <addressid>PayPal</addressid>
    <addressstatus>Confirmed</addressstatus>
    <billingagreementacceptedstatus>Yes
       </billingagreementacceptedstatus>
    <build>1.0006</build>
    <correlationid>6389c035b7785</correlationid>
    <countrycode>GB</countrycode>
    <custom>my_custom_data</custom>
    <email>abc_11_1191417836_per@example.com</email>
    <firstname>Test</firstname>
    <invnum>123-456-abc</invnum>
    <middlename>Arther</middlename>
    <lastname>User</lastname>
    <payerid>8PN3BJPN2WNDE</payerid>
    <payerstatus>unverified</payerstatus>
    <phonenum>+44 123 4567</phonenum>
    <salutation>Mr</salutation>
    <ShippingAddress>…</ShippingAddress>
    <suffix>Esq</suffix>
    <timestamp>2007-10-03T13:44:56Z</timestamp>
    <token>EC-4VL78907RS990801R</token>
    <version>2.300000</version>
</PayPalTxn>
```

Example XML for PayPal elements, for do_express_checkout_payment

```
<PayPalTxn>
    <ack>Success</ack>
    <amt>10.00</amt>
    <billingagreementid>billing123</billingagreementid>
    <build>1.0006</build>
    <correlationid>889dd330d54f0</correlationid>
    <currencycode>GBP</currencycode>
    <exchangerate>1.2222</exchangerate>
    <feeamt>0.54</feeamt>
    <ordertime>2007-10-03T13:45:17Z</ordertime>
    <paymentstatus>Completed</paymentstatus>
    <paymenttype>instant</paymenttype>
    <pendingreason>None</pendingreason>
    <reasoncode>None</reasoncode>
    <settleamt>13.22</settleamt>
    <taxamt>0.00</taxamt>
    <timestamp>2007-10-03T13:45:26Z</timestamp>
    <token>EC-4VL78907RS990801R</token>
    <transactionid>3YT66098JE5874747</transactionid>
    <transactiontype>expresscheckout</transactiontype>
    <version>2.300000</version>
</PayPalTxn>
```

Example XML for PayPal elements, for refund

```
<PayPalTxn>
    <build>1.0006</build>
    <feerefundamt>0.54</feerefundamt>
    <version>2.300000</version>
    <timestamp>2007-10-05T13:09:42Z</timestamp>
    <correlationid>5db496bf4454</correlationid>
    <refundtransactionid>83V1269821897443H</refundtransactionid>
    <grossrefundamt>10.00</grossrefundamt>
    <ack>Success</ack>
    <netrefundamt>9.4</netrefundamt>
</PayPalTxn>
```

Example XML for PayPal elements, for do_authorization

```
<PayPalTxn>
    <ack>Success</ack>
    <amt>10.00</amt>
    <build>1.0006</build>
    <timestamp>2007-10-03T13:45:26Z</timestamp>
    <transactionid>3YT66098JE5874747</transactionid>
    <version>2.300000</version>
</PayPalTxn>
```

Example XML for PayPal elements, for do_capture

```
<PayPalTxn>
    <ack>Success</ack>
    <amt>10.00</amt>
    <authorizationid>123456</authorizationid>
    <build>1.0006</build>
    <exchangerate>1.2222</exchangerate>
    <feeamt>0.54</feeamt>
    <ordertime>2007-10-03T13:45:17Z</ordertime>
    <parenttransactionid>J660-98E5-8747-473YT</parenttransactionid>
    <paymentstatus>Completed</paymentstatus>
    <paymenttype>instant</paymenttype>
    <receipt>34FB-SU1K-LAD9-SDA8</receipt>
    <settleamt>13.22</settleamt>
    <taxamt>0.00</taxamt>
    <timestamp>2007-10-03T13:45:26Z</timestamp>
    <transactionid>JE587-4747-3YT6-6098</transactionid>
    <transactiontype>express-checkout</transactiontype>
    <version>2.300000</version>
</PayPalTxn>
```

Example XML for PayPal elements, for do_void

```
<PayPalTxn>
    <ack>Success</ack>
    <amt>10.00</amt>
    <build>1.0006</build>
    <timestamp>2007-10-03T13:45:26Z</timestamp>
    <transactionid>3YT66098JE5874747</transactionid>
    <version>2.300000</version>
</PayPalTxn>
```

H.1.3.2. Errors

This element will only be returned if a PayPal error or warning occurs

Elements of Errors	
value	**description**
`Error`	See section H.1.3.3

Example XML for Errors element

```
<Errors>
  <Error id=”0”>…</Error>
  <Error id=”1”>…</Error>
   …
</Errors>
```

H.1.3.3. Error

This element will only be returned if a PayPal error or warning occurs. PayPal can return multiple Errors for a single transaction. The ID numbering of these errors will correspond to the number used by PayPal in their message response.

Elements of Errors
`error_code`
`short_message`
`long_message`
`severity_code`

Each error code which is returned will contain the short and long message. Please refer to appendix B of the *Name-Value Pair API Developer Guide and Reference* for a full list of error codes and their explanations.

Example XML for Error element

```
<Error id="0">
  <error_code>81104</error_code>
  <short_message>Missing Parameter</short_message>
  <long_message>CancelURL : Required parameter
      missing</long_message>
  <severity_code>severityCode</severity_code>
</Error>
<Error id="1">
  <error_code>10431</error_code>
  <short_message>Item amount is invalid</short_message>
  <long_message>Item amount is invalid</long_message>
</Error>
```

H.1.4. XML Example Responses

H.1.4.1. Successful Transactions

Example XML Response for a set_express_checkout transaction

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <status>1</status>
  <merchantreference>12345601</merchantreference>
  <reason>ACCEPTED</reason>
  <mode>LIVE</mode>
  <time>1132843641</time>
  <PayPalTxn>
    <ack>Success</ack>
    <build>1.0006</build>
    <timestamp>2007-10-03T13:44:56Z</timestamp>
    <token>EC-4VL78907RS990801R</token>
    <version>2.300000</version>
  </PayPalTxn>
</Response>
```

Example XML Response for a get_express_checkout_details transaction

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <status>1</status>
  <merchantreference>12345601</merchantreference>
  <reason>ACCEPTED</reason>
  <mode>LIVE</mode>
  <time>1132843641</time>
  <PayPalTxn>
    <ack>Success</ack>
    <addressid>PayPal</addressid>
    <addressstatus>Confirmed</addressstatus>
    <BillingAddress>
      <name>Test User</name>
      <street_address1>12 Boddingtons Road</street_address1>
      <street_address2>Regeant's Park</street_address2>
      <city>Wolverhampton</city>
      <region>West Midlands</region>
      <country>United Kingdom</country>
      <country_code>GB</country_code>
      <address_owner>Foobar Ltd</address_owner>
      <address_status>Confirmed</address_status>
    </BillingAddress>
    <billingagreementacceptedstatus>Yes
       </billingagreementacceptedstatus>
    <build>1.0006</build>
    <correlationid>6389c035b7785</correlationid>
    <countrycode>GB</countrycode>
    <custom>my_custom_data></custom>
    <email>abc_11_1191417836_per@example.com</email>
    <firstname>Test</firstname>
    <invnum>123-456-abc</invnum>
    <middlename>Arther</middlename>
    <lastname>User</lastname>
    <payerid>8PN3BJPN2WNDE</payerid>
    <payerstatus>unverified</payerstatus>
    <phonenum>+44 123 4567</phonenum>
    <salutation>Mr</salutation>
    <ShippingAddress>
      <name>Test User</name>
      <street_address1>1 Main Terrace</street_address1>
      <street_address2>Highgate</street_address2>
      <city>Wolverhampton</city>
      <region>West Midlands</region>
      <country>United Kingdom</country>
      <country_code>GB</country_code>
      <postcode>E12 4LQ</postcode>
      <telephone_number>0121 231 3122</telephone_number>
    </ShippingAddress>
    <suffix>Esq</suffix>
    <timestamp>2007-10-03T13:44:56Z</timestamp>
    <token>EC-4VL78907RS990801R</token>
    <version>2.300000</version>
  </PayPalTxn>
</Response>
```

Example XML Response for do_express_checkout_payment transaction

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <status>1</status>
  <merchantreference>12345601</merchantreference>
  <reason>ACCEPTED</reason>
  <mode>LIVE</mode>
  <time>1132843641</time>
  <PayPalTxn>
    <ack>Success</ack>
    <amt>10.00</amt>
    <billingagreementid>billing123</billingagreementid>
    <build>1.0006</build>
    <correlationid>889dd330d54f0</correlationid>
    <currencycode>GBP</currencycode>
    <exchangerate>1.2222</exchangerate>
    <feeamt>0.54</feeamt>
    <ordertime>2007-10-03T13:45:17Z</ordertime>
    <paymentstatus>Completed</paymentstatus>
    <paymenttype>instant</paymenttype>
    <pendingreason>None</pendingreason>
    <reasoncode>None</reasoncode>
    <settleamt>13.22</settleamt>
    <taxamt>0.00</taxamt>
    <timestamp>2007-10-03T13:45:26Z</timestamp>
    <token>EC-4VL78907RS990801R</token>
    <transactionid>3YT66098JE5874747</transactionid>
    <transactiontype>expresscheckout</transactiontype>
    <version>2.300000</version>
  </PayPalTxn>
</Response>
```

Example XML Response for a refund

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <status>1</status>
  <merchantreference>12345601</merchantreference>
  <reason>ACCEPTED</reason>
  <mode>LIVE</mode>
  <time>1132843641</time>
  <PayPalTxn>
    <build>1.0006</build>
    <feerefundamt>0.54</feerefundamt>
    <version>2.300000</version>
    <timestamp>2007-10-05T13:09:42Z</timestamp>
    <correlationid>5db496bf4454</correlationid>
    <refundtransactionid>83V1269821897443H</refundtransactionid>
    <grossrefundamt>10.00</grossrefundamt>
    <ack>Success</ack>
    <netrefundamt>9.4</netrefundamt>
  </PayPalTxn>
</Response>
```

Example XML Response for a do_authorization

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <status>1</status>
  <merchantreference>12345601</merchantreference>
  <reason>ACCEPTED</reason>
  <mode>LIVE</mode>
  <time>1132843641</time>
  <PayPalTxn>
    <ack>Success</ack>
    <amt>10.00</amt>
    <build>1.0006</build>
    <timestamp>2007-10-03T13:45:26Z</timestamp>
    <transactionid>3YT66098JE5874747</transactionid>
    <version>2.300000</version>
  </PayPalTxn>
</Response>
```

Example XML Response for a do_capture

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <status>1</status>
  <merchantreference>12345601</merchantreference>
  <reason>ACCEPTED</reason>
  <mode>LIVE</mode>
  <time>1132843641</time>
  <PayPalTxn>
    <ack>Success</ack>
    <amt>10.00</amt>
    <authorizationid>123456</authorizationid>
    <build>1.0006</build>
    <exchangerate>1.2222</exchangerate>
    <feeamt>0.54</feeamt>
    <ordertime>2007-10-03T13:45:17Z</ordertime>
    <parenttransactionid>J660-98E5-8747-473YT</parenttransactionid>
    <paymentstatus>Completed</paymentstatus>
    <paymenttype>instant</paymenttype>
    <receipt>34FB-SU1K-LAD9-SDA8</receipt>
    <settleamt>13.22</settleamt>
    <taxamt>0.00</taxamt>
    <timestamp>2007-10-03T13:45:26Z</timestamp>
    <transactionid>JE587-4747-3YT6-6098</transactionid>
    <transactiontype>express-checkout</transactiontype>
    <version>2.300000</version>
  </PayPalTxn>
</Response>
```

Example XML Response for a do_void

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <status>1</status>
  <merchantreference>12345601</merchantreference>
  <reason>ACCEPTED</reason>
  <mode>LIVE</mode>
  <time>1132843641</time>
  <PayPalTxn>
    <ack>Success</ack>
    <amt>10.00</amt>
    <build>1.0006</build>
    <timestamp>2007-10-03T13:45:26Z</timestamp>
    <transactionid>3YT66098JE5874747</transactionid>
    <version>2.300000</version>
  </PayPalTxn>
</Response>
```

H.1.4.2. PayPal Warning and Errors

Example XML Response for an error

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <merchantreference>12345601</merchantreference>
  <mode>LIVE</mode>
  <time>1132843641</time>
  <status>561</status>
  <reason>PayPal: Error returned in response</reason>
  <information>The transaction response from PayPal indicated that
      an error has occurred - see the extended PayPal
      information<information>
  <PayPalTxn>
```

```
    <ack>Failure</ack>
    <timestamp>2007-10-03T13:45:26Z</timestamp>
    <correlationid>889dd330d54f0</correlationid>
    <version>2.300000</version>
    <build>1.0006</build>
    <Errors>
      <Error id="0">
        <error_code>81104</error_code>
        <short_message>Missing Parameter</short_message>
        <long_message>CancelURL : Required parameter
          missing</long_message>
        <severity_code>severityCode</severity_code>
      </Error>
    </Errors>
  </PayPalTxn>
</Response>
```

H.1.4.3. Invalid Transactions

When DataCash is sent a transaction, it will be validated before forwarding the information to PayPal. If a transaction fails this validation, the response will not have a `PayPalTxn` element. Examples are shown below. A full list of error codes is available in the Developers Area. Additional examples - and suggestions for how to prevent the errors re-occurring - are published in the Support Centre.

Example XML Response for an invalid transaction

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <merchantreference>12345601</merchantreference>
  <status>560</status>
  <reason>PayPal: Not configured for service</reason>
  <information>Merchant is not configured for the PayPal
     service</information>
  <mode>LIVE</mode>
  <time>1132843641</time>
</Response>
```

Example XML Response for an invalid transaction

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
  <merchantreference>12345601</merchantreference>
  <status>565</status>
  <reason>PayPal: Invalid reference</reason>
  <information>Referenced transactions is not a
    PayPalTxn<information>
  <mode>LIVE</mode>
  <time>1132843641</time>
</Response>
```

H.1.5. Re-Directing the Customer

This section describes how to re-direct your customer from your website to PayPal.

In order to complete this process, you must have submitted a `set_express_checkout` transaction to DataCash, and received a successful response. This response will provide you with a token (section H.1.3.1):

Example Successful XML Response for set_express_checkout, highlighting the token

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>…</datacash_reference>
  <status>1</status>
  <merchantreference>…</merchantreference>
  <reason>ACCEPTED</reason>
  <mode>LIVE</mode>
  <time>…</time>
  <PayPalTxn>
    <token>EC-4VL78907RS990801R</token>
  </PayPalTxn>
</Response>
```

This token is used to construct the URL to which you will re-direct your customer, as shown below:

Example URL for re-direct

```
https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout
    &token=EC-4VL78907RS990801R
```

When redirecting the customer’s browser to the PayPal login page, PayPal recommends that you use the HTTPS response 302 “Object Moved” with the URL above as the value of the Location header in the HTTPS response. Ensure that you use an SSL-enabled server to prevent browser warnings about a mix of secure and insecure graphics.

H.2. Variable End of Day

This section describes the Variable End of Day service for the dynamic processing model only. If you are using the Static processing model, no transactions need to be sent to the DataCash Payment Gateway.

A full description of this service is available on the website:
http://www.datacash.com/services/bankcard/veod/overview.php

H.2.1. Schema Elements for Request

In this section, the fields that can be submitted when using the variable end of day (dynamic) service will be presented, along with sample XML for those fields.

When presenting an eod transaction, the following schema elements should be presented:

- `Request`
  - `Authentication` – section A.1.1
  - `Transaction`
    - `TxnDetails` – the merchant reference – section H.2.1.1
    - `EoDTxn` – the method eod – section H.2.1.2

All elements are required, unless otherwise stated

H.2.1.1. TxnDetails

There is one element to be presented in this parent element

Elements in TxnDetails

Element Name	description	values / limitations
merchantreference	A unique reference number for each transaction	Minimum 6, maximum 30 alphanumeric characters. Must be unique

Example XML for EoDTxn elements

```
<TxnDetails>
  <merchantreference>48tv3498wy3c4dt</merchantreference>
</TxnDetails>
```

H.2.1.2. EoDTxn

There is one element to be presented in this parent element

Elements in EoDTxn		
Element Name	**description**	**values / limitations**
`method`	The transaction type	`eod`

Example XML for EoDTxn elements

```
<EoDTxn>
  <method>eod</method>
</EoDTxn>
```

H.2.2. Example Requests

Example XML Request

```
<Request>
  <Authentication>
    <client>...</client>
    <password>...</password>
  </Authentication>
  <Transaction>
    <TxnDetails>
      <merchantreference>...</merchantreference>
    </TxnDetails>
    <EoDTxn>
      <method>eod</method>
    </EoDTxn>
  </Transaction>
</Request>
```

H.2.3. Schema Elements for Response

Please refer to section H.1.1.3. There are no additional elements which may be returned in the XML response for this service

H.2.4. Example Responses

Example XML Response for a successful transaction

```
<Response>
  <datacash_reference>4900200047685356</datacash_reference>
  <merchantreference>117990881318233</merchantreference>
  <mode>LIVE</mode>
  <reason>ACCEPTED</reason>
  <status>1</status>
  <time>1179908813</time>
</Response>
```

Example XML Response for a rejected transaction

```
<Response>
  <datacash_reference>4000200047685351</datacash_reference>
  <merchantreference>117990875018165</merchantreference>
  <mode>LIVE</mode>
  <reason>VEoD: already received an EoD</reason>
  <status>532</status>
  <time>1179908750</time>
</Response>
```

H.3. RBS Gift Card

H.3.1. Summary of Service

The RBS Gift Card service provides the ability to perform a number of transaction types against RBS Gift Cards:

- Balance Enquiries: To determine the remaining balance
- Redemptions: To perform an authorisation against the card
- Refunds: To refund an amount to the card
- txn_refunds: To refund an amount to the card based on a previously authorized redemption.

H.3.2. Schema Elements for Request

Element Name:	`RbsGiftCardTxn`
Position:	`Request.Transaction`

Elements of RbsGiftCardTxn			
Element Name	**description**	**values / limitations**	**required**

`method`	The method of the transaction.	`balance_enquiry`
		`redeem`
		`refund`
		`txn_refund`
			R

`Card`	The card details of the transaction to be authenticated. This element contains sub elements. See B.1.1.1 for more information. This is not used in the case of a `txn_refund`.	See B.1.1.1 for more information	M
`Reference`	When performing a `txn_refund`, this element should be used to indicate the transaction that is being refunded.	The value contained should be the datacash_reference of the original RBS gift card transaction to refund.	M

H.3.3. Balance Enquiry

The balance enquiry transaction is provided to allow the merchant to check the existing balance of an RBS Gift Card prior to performing a redemption request.

An example balance enquiry transaction is illustrated below:

Example Balance Enquiry Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <merchantreference>12345601</merchantreference>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>

    <RbsGiftCardTxn>
      <method>balance_enquiry</method>
      <Card>
        <pan>xxxxxxxxxxxxxxxx</pan>
        <accesscode>xxxxxxxx</accesscode>
        <expirydate>04/06</expirydate>
        <startdate>01/99</startdate>
      </Card>
    </RbsGiftCardTxn>

  </Transaction>
</Request>
```

An example response to a balance enquiry is shown below:

Example Balance Enquiry Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>

   <datacash_reference>4000900012345671</datacash_reference>
   <status>1</status>
   <merchantreference>12345601</merchantreference>
   <reason>ACCEPTED</reason>

   <RbsGiftCardTxn>
     <authcode>100000</authcode>
     <balance>100.00</balance>
     <currency>GBP</currency>
   </RbsGiftCardTxn>

   <mode>TEST</mode>
   <time>1132843641</time>

</Response>
```

H.3.4. Redemption

The redemption transaction allows the merchant to collect funds from the gift card.
An example redemption transaction is shown below:

Example Redemption Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <amount currency="GBP">100.00</amount>
      <merchantreference>12345601</merchantreference>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>

    <RbsGiftCardTxn>
      <method>redeem</method>
      <Card>
        <pan>xxxxxxxxxxxxxxxx</pan>
        <accesscode>xxxxxxxx</accesscode>
        <expirydate>04/06</expirydate>
        <startdate>01/99</startdate>
        <Cv2Avs>
          <cv2>123</cv2>
        </Cv2Avs>
       </Card>
    </RbsGiftCardTxn>

  </Transaction>
</Request>
```

An example response to a redeem transaction where the card has sufficient funds to complete the transaction is shown below:

Example Redemption Response where card has sufficient funds

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <datacash_reference>4000900012345671</datacash_reference>
   <status>1</status>
   <merchantreference>12345601</merchantreference>
   <reason>ACCEPTED</reason>

   <RbsGiftCardTxn>
     <authcode>100000</authcode>
     <balance>100.00</balance>
   </RbsGiftCardTxn>

   <mode>TEST</mode>
   <time>1132843641</time>
</Response>
```

The remaining balance is presented in the response XML as shown above.

Where the card contains insufficient funds, the response is slightly different. The available funds on the Gift Card will be deducted and the XML response will contain a negative balance. The negative balance will specify the difference between the requested amount and the card balance i.e. the outstanding balance required to complete the transaction.

The response below illustrates a negative balance being returned:

Example Redemption Response where card has insufficient funds

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
   <datacash_reference>4000900012345671</datacash_reference>
   <status>1</status>

   <merchantreference>12345601</merchantreference>
   <reason>ACCEPTED</reason>

   <RbsGiftCardTxn>
     <authcode>100000</authcode>
     <balance>-41.00</balance>
   </RbsGiftCardTxn>

   <mode>TEST</mode>
   <time>1132843641</time>

</Response>
```

Where a negative balance is returned, the merchant must submit another transaction request using another card (either Gift Card or Credit/Debit card) to secure the outstanding balance.

See the developers guide for information on processing Credit / Debit card transactions.

H.3.5. Refunds

The refund transaction allows the merchant to refund the gift card.
An example refund transaction is shown below:

Example Refund Request

```
<Request>

  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <amount currency="GBP">10.59</amount>
      <merchantreference>12345601</merchantreference>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>

    <RbsGiftCardTxn>
      <method>refund</method>
      <Card>
        <pan>xxxxxxxxxxxxxxxx</pan>
        <accesscode>xxxxxxxx</accesscode>
        <expirydate>04/06</expirydate>
        <startdate>01/99</startdate>
      </Card>
    </RbsGiftCardTxn>

  </Transaction>
</Request>
```

An example response to a successful refund transaction is shown below:

Example Refund Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
   <datacash_reference>4000900012345671</datacash_reference>
   <status>1</status>
   <merchantreference>12345601</merchantreference>
   <reason>ACCEPTED</reason>

   <RbsGiftCardTxn>
     <authcode>100000</authcode>
     <balance>100.00</balance>
   </RbsGiftCardTxn>

   <mode>TEST</mode>
   <time>1132843641</time>

</Response>
```

Note: the response formatting is identical to that for redemption transactions.

H.3.6. Txn Refunds

The txn_refund transaction allows the merchant to refund the gift card with a maximum amount equal to that contained within the original redemption transaction as identified by the ‘reference’ provided in the XML.

Note that where multiple txn_refund transactions are submitted against the same redemption, the sum of the refunded amounts must not exceed the originally authorised amount.

An example txn_refund transaction is shown below:

Example txn_refund Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <amount>10.59</amount>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>

    <RbsGiftCardTxn>
      <method>txn_refund</method>
      <reference>4100900012345675</reference>
    </RbsGiftCardTxn>

  </Transaction>
</Request>
```

If the amount is not provided, the amount refunded will be that of the original redemption transaction. Note that ‘currency’ is not permitted in a txn_refund.

An example response to a txn_refund is provided below:

Example txn_refund Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
   <datacash_reference>4000900012345671</datacash_reference>
   <status>1</status>
   <merchantreference>4000900012345670</merchantreference>
   <reason>ACCEPTED</reason>

   <RbsGiftCardTxn>
     <authcode>100000</authcode>
     <balance>100.00</balance>
   </RbsGiftCardTxn>

   <mode>TEST</mode>
   <time>1132843641</time>
</Response>
```

The ‘merchantreference’ returned in response to a txn_refund is the ‘datacash_reference’ of the original redemption transaction.

H.3.7. Unsuccessful Transactions

In the event of an unsuccessful transaction (where a valid response is received indicating failure at RBS), the balance may be returned, and additional data provided in the response message may be returned in the `information` field.

An example of an unsuccessful transaction can be seen below:

Example Response for a Unsuccessful Transaction

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
   <datacash_reference>4000900012345672</datacash_reference>
   <status>7</status>

   <merchantreference>12345602</merchantreference>
   <reason>DECLINED</reason>

   <RbsGiftCardTxn>
     <balance>5.25</balance>
   </RbsGiftCardTxn>

   <information>CARD EXPIRED</information>
   <mode>TEST</mode>
   <time>1132843641</time>

</Response>
```

H.3.8. Invalid Transactions

In the event of a transaction being rejected by the DataCash Payment Gateway without going to RBS for processing, the `RbsGiftCardTxn` block will not be returned. A suitable status code and simple message will be returned, with additional details in the `information` field. This is in-line with the existing DataCash response format.

For example:

Example Response for a Invalid Transaction

```
<?xml version="1.0" encoding="UTF-8"?>

<Response>
   <datacash_reference>4000900012345672</datacash_reference>
   <status>631</status>
   <merchantreference>12345602</merchantreference>

   <reason>Inappropriate card number</reason>
   <information>The card provided is not an RBS Gift
Card</information>
   <mode>TEST</mode>
   <time>1132843641</time>

</Response>
```

Such responses will be received if the XML submitted contains unexpected elements, has missing elements or the data contained within those elements has failed to meet the validation rules.

H.3.9. Failed Transactions

A failed transaction is one where an attempt to submit a request to RBS was made, but a communication error has rendered the outcome either failed or unknown.

An example of a failed transaction is given below for completeness:

Example Response for a Failed Transaction

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
   <datacash_reference>4000900012345672</datacash_reference>
   <status>6</status>

   <merchantreference>12345602</merchantreference>
   <reason>BANK DID NOT REPLY</reason>

   <mode>TEST</mode>
   <time>1132843641</time>

</Response>
```

H.3.10. Notes

H.3.10.1. Accesscode

This element is optional and if presented must be 8 digits in length. If provided, the ‘accesscode’ must match that held against the account otherwise the transaction will be declined.

H.3.10.2. Capturemethod

This can be one of ‘ecomm’ or ‘cnp’ for RBS Gift Card transactions. If this is not specified, the capturemethod will be set by determining the environment of a terminal configured on your account which is capable of processing the card and currency combination specified in the transaction.

Note that the ‘capturemethod’ is mandatory if the account is configured with multiple merchant ID environments.

H.4. PrePay Technology (PPT)

H.4.1. Summary of Service

The PrePay Technology service (hereafter referred to as PPT) provides the ability to perform a number of transaction types against PPT gift cards:

- Top Ups: To load the PPT gift card with funds
- Redemptions: To perform an authorization against the PPT gift card
- Refunds: To refund an amount to the PPT gift card
- Balance Enquiries: To determine the remaining balance on the PPT gift card
- Reversals: Reverse a previous top up, redemption or refund on the PPT gift card
- txn_refunds: To refund an amount to the card based on a previously authorized redemption.

PPT transactions can be performed in ecommerce or cardholder not present environments. Alternatively PPT cards can be swiped or keyed into a card terminal.

H.4.2. Schema Elements for Request

Element Name: PPTCardTxn
Position: Request.Transaction

Elements of PPTCardTxn			
Element Name	**description**	**values / limitations**	**required**
method	The method of the transaction.	top_up redeem refund balance_enquiry reversal txn_refund	R
pan	The PPT card number. Element used for "ecomm", "cnp" and "keyed" capturemethods.	The PPT card number	M
pin	A Pin number can optionally be supplied when performing "top-up", "redeem" and "balance_enquiry" transactions.	4 digit pin number	O
reference	The datacash reference of the related transaction. This element is only valid for "reversal" or "txn_refund" methods.	Datacash reference of the related transaction	M
merchantreference	If subscribed to the merchant reference cancellations service, a "reversal" may be invoked with this element instead of reference	Merchant reference of the related transaction	O
card_details	For the "swiped" capturemethod only, this element is used instead of pan.	The track2 information from the PPT gift card	M
Terminal	For PPT transactions using a terminal ("keyed", "swiped" capturemethods), this element should be present.	See section G.1.1.3	M

H.4.3. Schema Elements for Response

Element Name: PPTCardTxn
Position: Response

Elements of PPTCardTxn		
Element Name	**description**	**values / limitations**
Authcode	The authcode of the transaction, this element can also contain messages from PPT e.g "INVALID PIN".	
card_activated	PPT may return information in their authorisation response that indicates that a card has been activated as a result of the transaction. This element will only be included where card activation has occurred.	"yes"
Balance	The balance remaining on the PPT gift card following the transaction, returned with major and minor units (i.e 0.00).	

H.4.4. Top Ups

Top up transactions are used to load funds onto the PPT gift card. An example of a “swiped” top up request is shown below:

Example PPT Top Up Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <amount currency="GBP">100.00</amount>
      <merchantreference>12345601</merchantreference>
      <capturemethod>swiped</capturemethod>
    </TxnDetails>

    <PPTCardTxn>

      <Terminal id="xxxxxxxx">
        <terminal_capabilities ic_reader="true"
magnetic_stripe_reader="true" manual_card_entry="true" />
        <features_capabilities pin_pad_available="true" />
      </Terminal>

      <method>top_up</method>
      <card_details
type="track2">xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</card_details>
      <pin>xxxx</pin>
    </PPTCardTxn>

  </Transaction>
</Request>
```

An example of a DataCash response to a top up request is shown below. Note that in this case, this was the first use of the gift card and so the “card_activated” element is present.

Example PPT Top Up Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <reason>ACCEPTED</reason>
  <status>1</status>

  <currency>GBP</currency>
  <datacash_reference>4000900012345671</datacash_reference>
  <merchantreference>12345601</merchantreference>

  <PPTCardTxn>
    <authcode>100000</authcode>
    <balance>100.00</balance>
    <card_activated>yes</card_activated>
  </PPTCardTxn>
  <mode>TEST</mode>
  <time>_UNIX_TIME_</time>
</Response>
```

H.4.5. Redemption

Redeem transactions are used to authorise fund removal from the gift card. An example of a “keyed” redemption request is shown below:

Example PPT Redemption Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <amount currency="GBP">40.00</amount>
      <merchantreference>12345602</merchantreference>
      <capturemethod>keyed</capturemethod>
    </TxnDetails>

    <PPTCardTxn>
      <Terminal id="xxxxxxxx">
        <terminal_capabilities ic_reader="true"
magnetic_stripe_reader="true" manual_card_entry="true" />
        <features_capabilities pin_pad_available="true" />
      </Terminal>

      <method>redeem</method>
      <pan>xxxxxxxxxxxxxxxx</pan>
      <pin>xxxx</pin>
    </PPTCardTxn>

  </Transaction>
</Request>
```

Note that the “pin” element is optional.

An example of a response to the above request is shown below:

Example PPT Redemption Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <reason>ACCEPTED</reason>
  <status>1</status>

  <currency>GBP</currency>
  <datacash_reference>4800900012345672</datacash_reference>
  <merchantreference>12345602</merchantreference>

  <PPTCardTxn>
    <authcode>100000</authcode>
    <balance>60.00</balance>
  </PPTCardTxn>

  <mode>TEST</mode>
  <time>_UNIX_TIME_</time>
</Response>
```

H.4.6. Refunds

Refund transactions are used to refund an amount to the gift card, where the refund doesn’t relate to an existing redemption on that gift card. An example of an e-commerce refund request is shown below:

Example PPT Refund Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <amount currency="GBP">30.00</amount>
      <merchantreference>12345603</merchantreference>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>

    <PPTCardTxn>
      <method>refund</method>
      <pan>xxxxxxxxxxxxxxxx</pan>
    </PPTCardTxn>

  </Transaction>
</Request>
```

An example response to the above transaction is shown below:

Example PPT Refund Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <reason>ACCEPTED</reason>
  <status>1</status>

  <currency>GBP</currency>
  <datacash_reference>4800900012345673</datacash_reference>
  <merchantreference>12345603</merchantreference>

  <PPTCardTxn>
    <authcode>100000</authcode>
    <balance>90.00</balance>
  </PPTCardTxn>

  <mode>TEST</mode>
  <time>_UNIX_TIME_</time>
</Response>
```

H.4.7. Balance Enquiries

Balance enquiries are used query the amount of funds on the gift card. An example of an e-commerce balance enquiry request is shown below:

Example PPT Balance Enquiry Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <merchantreference>12345604</merchantreference>
      <capturemethod>ecomm</capturemethod>
    </TxnDetails>

    <PPTCardTxn>
      <method>balance_enquiry</method>
      <pan>xxxxxxxxxxxxxxxx</pan>
    </PPTCardTxn>

  </Transaction>
</Request>
```

An example response to the above transaction is shown below:

Example PPT Balance Enquiry Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <reason>ACCEPTED</reason>
  <status>1</status>

  <currency>GBP</currency>
  <datacash_reference>4800900012345674</datacash_reference>
  <merchantreference>12345604</merchantreference>

  <PPTCardTxn>
    <authcode>100000</authcode>
    <balance>90.00</balance>
  </PPTCardTxn>

  <mode>TEST</mode>
  <time>_UNIX_TIME_</time>
</Response>
```

H.4.8. Reversals

Reversals can be used to reverse a previous top up, redemption or refund on the PPT gift card, by using the datacash_reference (supplied in the response of the previous transaction). An example of an e-commerce reversal request is shown below:

Example PPT Reversal Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>

    <PPTCardTxn>
      <method>reversal</method>
      <reference>4800900012345673</reference>
    </PPTCardTxn>

  </Transaction>
</Request>
```

An example response to the above transaction is shown below:

Example PPT Reversal Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <reason reversal='success'>ACCEPTED</reason>
  <status>1</status>
  <currency>GBP</currency>
  <datacash_reference>4800900012345675</datacash_reference>
  <information>Reversal ACCEPTED status=1
authcode=100000</information>

  <PPTCardTxn>
    <authcode>100000</authcode>
    <balance>60.00</balance>
  </PPTCardTxn>

  <mode>TEST</mode>
  <time>_UNIX_TIME_</time>
</Response>
```

H.4.9. Txn_refunds

Txn_Refund transactions are used to refund an amount to the gift card, where the refund relates to an existing redemption on that gift card. The refund may be for the full amount of the transaction, or a partial amount (if the amount element is missing then the full amount will be assumed).
An example of an e-commerce txn_refund request is shown below:

Example PPT Txn_Refund Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <amount currency="GBP">10.59</amount>
    </TxnDetails>

    <PPTCardTxn>
      <method>txn_refund</method>
      <reference>4800900012345672</reference>
    </PPTCardTxn>

  </Transaction>
</Request>
```

An example response to the above transaction is shown below:

Example PPT Txn_Refund Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <reason>ACCEPTED</reason>
  <status>1</status>

  <currency>GBP</currency>
  <datacash_reference>4800900012345675</datacash_reference>
  <merchantreference>12345602</merchantreference>

  <PPTCardTxn>
    <authcode>100000</authcode>
    <balance>79.41</balance>
  </PPTCardTxn>

  <mode>TEST</mode>
  <time>_UNIX_TIME_</time>
</Response>
```

H.4.10. Unsuccessful Transactions

H.4.10.1. Declined

In the event of a decline by PPT, the balance may or may not be returned, for example:

Example of a PPT Decline response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <reason>DECLINED</reason>
  <status>7</status>

  <currency>GBP</currency>
  <datacash_reference>4800900012345672</datacash_reference>
  <merchantreference>12345602</merchantreference>

  <PPTCardTxn>
    <authcode>DECLINED</authcode>
    <balance>5.25</balance>
  </PPTCardTxn>

  <mode>TEST</mode>
  <time>_UNIX_TIME_</time>
</Response>
```

H.4.10.2. Pin Mismatch

This scenario follows the same format as a Decline above, however the reason field indicates why the transaction was declined, for example:

Example of a PPT PIN mismatch response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <reason>PIN MISMATCH</reason>
  <status>7</status>

  <currency>GBP</currency>
  <datacash_reference>4800900012345672</datacash_reference>
  <merchantreference>12345602</merchantreference>

  <PPTCardTxn>
    <authcode> PIN MISMATCH                 010000010000
</authcode>
    <balance>5.25</balance>
  </PPTCardTxn>

  <mode>TEST</mode>
  <time>_UNIX_TIME_</time>
</Response>
```

H.4.11. Invalid Transactions

In the event of a transaction being rejected by the DataCash Payment Gateway without going to PPT for processing, the `PPTCardTxn` block will not be returned. A suitable status code and simple message will be returned, with additional details in the `information` field. It may not be possible to include currency information in such transactions. For example

Example of a response to an Invalid PPT transaction

```
<?xml version="1.0" encoding="UTF-8"?>

<Response>
   <datacash_reference>4000900012345672</datacash_reference>
   <status>492</status>
   <merchantreference>12345602</merchantreference>

   <reason>PPT: Inappropriate Request Element</reason>
   <information>Field LINEITEMDETAIL may not be included in a PPT
transaction</information>

   <mode>TEST</mode>
   <time>1132843641</time>

</Response>
```

H.4.12. Failed Transactions

A failed transaction will be one where an attempt to submit a request to PPT was made, but a communication error has rendered the outcome either failed or unknown.

On behalf of a merchant, DataCash issue an automatic reversal in this case, and the `reversed` element is used to indicate whether the reversal was successful. This will only be applicable only to error 6 responses.

Elements of Response		
Element Name	**description**	**values / limitations**

`reversed`	Indicates whether an automatic reversal issued by DataCash in the event of a communications failure was successful.	`Y - The original transaction was reversed`
		`N - The original transaction did not reach the acquirer so was not reversed`
		`F - A reversal was attempted but failed`

In the event of a `reversed` flag of `F`, it is not possible for DataCash to know the status of the transaction, and the merchant will have to investigate further with PPT.

For example:

Example of DataCash Response on PPT communications failure

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
   <datacash_reference>4000900012345672</datacash_reference>
   <status>6</status>
   <merchantreference>12345602</merchantreference>
   <reason>BANK DID NOT REPLY</reason>
   <information>Reversal attempt FAILED</information>

   <reversed>F</reversed>

   <currency>EUR</currency>
   <mode>TEST</mode>
   <time>1132843641</time>
</Response>
```

H.5. Fexco Dynamic Currency Conversion

H.5.1. Summary of service

This service allows merchants to take international payments via Fexco’s third party Dynamic Currency Conversion (herin referred to as DCC) product.

Transactions placed using the DCC service can charge a bank card in the currency of the account that the cardholder uses rather than the currency of the merchant processing the txn. The amount changed to the customer is calculated using exchange rates provided by Fexco to the merchant.

H.5.2. Schema Elements for Request

Element Name: `FexcoDCC`
Position: `Request.Transaction.TxnDetails`

Elements of FexcoDCC			
Element Name	**description**	**values / limitations**	**required**
`BaseCurrency`	The currency that the merchant usually trades in	Valid ISO 4217 alphabetic currency code	R
`BaseAmount`	The amount that the merchant has requested for the transaction		R
`ExchangeRate`	The exchange rate provided by FEXCO for the transaction	Must contain a decimal point, and contain no more than 5 decimal places	R

Whenever the FexcoDCC block is present in an XML transaction, the currency attribute of Request.Transaction.TxnDetails.amount is mandatory.

Note that Fexco DCC transactions can only be supplied in bank card transactions where the transaction method is either “auth” or “pre”.

The Fexco DCC service is compatiable with both DataCash MPI and Third Party MPI 3-D Secure transactions. In order to use the Fexco DCC service with these services, the FexcoDCC element must only be supplied in the initial stage (Where the transaction method is “auth” or “pre”).

In the case of Airlines transactions, the per-passenger ticket prices can be in either the DCC (Post conversion) currency or the Base (Pre conversion) currency. No cross check on the totalling of these values against the transaction amount will be made for DCC transactions in the DataCash Payment Gateway.

H.5.3. Example Transaction

The following is an example authorisation. Note that the post conversion amount and currency is specified in the amount element.

For example:

Example of a Fexco DCC Authorisation Request

```
<Request>
  <Authentication>
    <client>xxxxxxxx</client>
    <password>xxxxxxxx</password>
  </Authentication>

  <Transaction>
    <TxnDetails>
      <amount currency="USD">200.00</amount>
      <merchantreference>12345601</merchantreference>
      <capturemethod>ecomm</capturemethod>

      <FexcoDCC>
        <BaseCurrency>GBP</BaseCurrency>
        <BaseAmount>100.00</BaseAmount>
        <ExchangeRate>2.0000</ExchangeRate>
      <FexcoDCC>

    </TxnDetails>

    <CardTxn>
      <method>auth</method>
      <Card>
        <pan>xxxxxxxxxxxxxxxx</pan>
        <expirydate>12/15</expirydate>
      </Card>
    </CardTxn>

  </Transaction>
</Request>
```

I. Appendices

I.1. LID Commodity Codes

These commodity codes can be used for the LID Service, section B.2.1.4

Commodity Code	Description
7461	Accounting/tax Consultancy Services
7464	Advertising And Marketing Services
3100	Agricultural Fertilisers
8800	Aircraft Parts
2200	Alcoholic Beverages
1200	Animal Feeding Stuffs
7300	Barbed Wire/fencing
0900	Beverages - Non-alcoholic
8705	Bicycles/tricycles
9000	Binoculars/telescopes
4900	Books; Printed Material
2500	Building Aggregates
6900	Building Bricks/tiles/stones
8507	Burglar & Fire Alarms
9001	Cameras/flashes
7160	Car Hire Services: Qualifying Cars
6562	Car Leasing Services: Qualifying Cars
6565	Car Leasing Services: Non - Qualifying Cars
5700	Carpets/floor Coverings
5561	Catering Services
6901	Sinks, Baths, Washbasins; Other Plumbing/heating Products
2800	Chemicals/acids
3400	Cleaning Products
7468	Cleaning Services
9100	Clocks, Watches & Parts
6200	Clothing
8701	Commercial Motor Vehicles
6563	Commercial Vehicle Leasing Services
7260	Computer Consultancy Services
8405	Computer Hardware
8406	Computer Software
7163	Computer/office Machine Rental
4560	Construction/demolition Services
0400	Dairy Produce
7261	Data Processing Services
8060	Driving Tuition Services
8509	Electric Lamps
4562	Electrical Contracting
8508	Electrical Switches/fuses/plugs/circuits
8061	Employee Training
9360	Employee Welfare Services
7466	Employment Bureau Services
6561	Equipment Leasing Services
7061	Estate Agency Services
6564	Factoring Services
6760	Financial Administration Services
6560	Financial Services (non Insurance)
8404	Fire Extinguishers
3000	First Aid Products

Commodity Code	Description
0300	Fish/seafood Products
6400	Footwear
6362	Freight / Carriage Charges (non Courier / Delivery Service Providers)
6061	Freight Transport Services
0800	Fruit And Fruit Products
8500	Generators/electric Motors
7000	Glassware
8200	Hand Tools
6500	Headgear
5560	Hotel/restaurant Services
8401	Industrial Engines/plant: Parts
3600	Industrial Explosives
8400	Industrial Mechanical Appliances/parts
7100	Industrial Metals (base/processed)
8402	Industrial Plant/equipment
3800	Insecticides/fungicides/disinfectants
6660	Insurance Services - General
4565	Joinery Services
8403	Kitchen Equipment
4200	Leather Cases And Bags
7460	Legal Services - Lawyers/solicitors
9261	Library/archiving Services
0100	Live Animals
2702	Lubricating Oils
8201	Machine Tools
8202	Machine/hand Tool Parts
7262	Maintenance/repair - Office Equipment
7463	Management Consultancy Services
7462	Market Research Services
9004	Measuring Equipment
0200	Meat/meat Products
9260	Media/news Agency Services
8560	Medical Services
9160	Membership Subscriptions
9003	Microscopes
8700	Motor Cars
8704	Motor Cycles/sidecars
8703	Motor Vehicle Parts (excluding Tyres)
5060	Motor Vehicle Repair/maintenance Services
9200	Musical Instruments
4801	Newsprint
9400	Office Furniture
4800	Office Stationery
7472	Other Business Services
6300	Packing Materials/sacks/bags
7470	Packing Services
8300	Padlocks, Locks, Keys. Safes, Strong Boxes, Other Ironmongery Products
4566	Painting/glazing Services
3200	Paints/varnishes
6060	Passenger Transport
6761	Pension Advisory/administration Services
2701	Petrol/diesel - Road Fuel
9002	Photocopiers
3700	Photographic Film Products
7469	Photographic Services
0600	Plants, Shrubs And Trees

Commodity Code	Description
4564	Plastering Services
4563	Plumbing, Heating And Drainage Services
6460	Courier Services (courier / Delivery Services Providers)
2261	Printing Services
7062	Property Management Services
7060	Property Rentals
2260	Publishing Services
8505	Radios
8600	Railway Locomotive Parts
8502	Record/cassette/cd Players
7161	Rental Of Agricultural Machinery
7162	Rental Of Construction Equipment
7164	Rental Of Sports/recreational Equipment
4561	Roofing Services
5600	Rope, cable, Netting
4000	Rubber Products
7471	Secretarial/translation Services
7467	Site Security Services
2700	Solid Fuel; Heating/ Industrial/ Agricultural Oils
8702	Special Purpose Vehicles
7560	Statutory Charges/licenses
6360	Storage/warehousing Services
8503	Tape Recorders/answering Machines
7465	Technical Analysis Services
6461	Telecommunications Services
8501	Telephones
8506	Television Sets
2400	Tobacco/tobacco Products
9500	Toys, Games And Models
8706	Trailers/trailer Parts
6361	Travel Agency Services
4001	Tyres - Motor Vehicles
6600	Umbrellas And Walking Sticks
9999	Vat: Insurance Repairs
0700	Vegetables And Vegetable Products
8407	Vending Machines
8504	Video Recorders
4802	Wallpaper
3760	Waste Disposal Services
4700	Wood, Timber Products, Pulp And Paper Products

I.2. Real Time Fraud Screening: Schema Elements by ReD Name

The Real Time Fraud Screening Service is outlined in section D.3

In this section, elements are listed by ReD name (not DataCash schema names). This is to enable bespoke merchants to identify the DataCash elements required to trigger all the rules set up by ReD.

The following information should be noted about these fields:

- If no `Recipient` is specified:
  - the `Customer Address` will be used for the `edSHIP` fields
- If only one `Recipient` is specified:
- the `ebSHIP` fields will be used in preference to the `ebRECIPIENT` fields
- If more than one `Recipient` is specified
  - the `ebRECIPIENT` fields will be used
  - the `Customer Address` elements will be ignored

ReD field	DataCash Schema Element	Section
ebBILLADDRESS2	moreaddress	D.3.1.2
ebBILLAPT	apartment	D.3.1.2
ebBILLCITY	city	D.3.1.2
ebBILLCOUNTRY	country	D.3.1.2
ebBILLSTATE	region	D.3.1.2
ebBILLSTREET	streetaddress	D.3.1.2
ebBILLZIPCD	postcode	D.3.1.2
ebCARRIER	carrier	D.3.1.12
ebCINPRESENT	cv2_present	D.3.1.6
ebCORPADDRESS2	moreaddress	D.3.1.1
ebCORPCOUNTRY	country	D.3.1.1
ebCORPFAX	fax	D.3.1.3
ebCORPNAME	name	D.3.1.3
ebCORPPHONE	telephone	D.3.1.3
ebCORPPONBR	pobox	D.3.1.1
ebCORPPURCHDESC	purchase_description	D.3.1.3
ebCORPSTATE	region	D.3.1.1
ebCORPSTREET	streetaddress	D.3.1.1
ebCORPSUITE	apartment	D.3.1.1
ebCORPCITY	city	D.3.1.1
ebCORPZIPCD	postcode	D.3.1.1
ebCUSTAGE	age	D.3.1.4
ebCUSTBDATE	date_of_birth	D.3.1.4
ebCUSTCOOKIES	cookies	D.3.1.5
ebCUSTEMAIL	email	D.3.1.4
ebCUSTFAX	fax	D.3.1.4
ebCUSTFIRSTNAME	forename	D.3.1.4
ebCUSTHOMEPHONE	alt_telephone	D.3.1.4
ebCUSTID	account	D.3.1.5

ebCUSTIP	ip_address	D.3.1.4
ebCUSTLASTNAME	surname	D.3.1.4
ebCUSTMAXAGE	max_age	D.3.1.4
ebCUSTMIDNAME	middlename	D.3.1.4
ebCUSTMINAGE	min_age	D.3.1.4
ebCUSTSALUTATION	salutation	D.3.1.4
ebCUSTSSN	national_id	D.3.1.4
ebCUSTTM	pctime	D.3.1.4
ebCUSTWORKPHONE	telephone	D.3.1.4
ebDEFFORMFILL	formfill	D.3.1.5
ebFORGOTPWD	forgot_password	D.3.1.5
ebGIFTCARDTYPE	attribute card	D.3.1.9, D.3.1.8
ebGIFTMSG	gift	D.3.1.9, D.3.1.8
ebHANDLING	handling	D.3.1.10
ebITEMCARRIER	carrier	D.3.1.12
ebITEMGIFTCARDTYPE	attribute card	D.3.1.9, D.3.1.8
ebITEMGIFTMESSAGE	gift	D.3.1.9, D.3.1.8
ebITEMSHIPCOMMENTS	comments	D.3.1.12
ebITEMSHIPINSTRUCTION	instructions	D.3.1.12
ebITEMSHIPMENTNO	tracking_number	D.3.1.12
ebITEMSHIPMETHOD	method	D.3.1.12
ebMANPARTNO	attribute manpartno	D.3.1.8
ebMANUFACTURER	attribute manufacturer	D.3.1.8
ebPASSWORD	password_access	D.3.1.5
ebPREVIOUSCUST	previous_cust	D.3.1.5
ebPRODCATEGORY	attribute category	D.3.1.8
ebPRODCD	attribute product_code	D.3.1.8
ebPRODDESC	description	D.3.1.8
ebPRODQUANTITY	attribute quantity	D.3.1.8
ebPRODSKU	attribute sku	D.3.1.8
ebPRODTYPE	attribute type	D.3.1.8
ebPRODUNITPRICE	attribute unit_price	D.3.1.8
ebRECIPIENTADDRESS2	moreaddress	D.3.1.1
ebRECIPIENTAPT	apartment	D.3.1.1
ebRECIPIENTCITY	city	D.3.1.1
ebRECIPIENTCOUNTRY	country	D.3.1.1
ebRECIPIENTEMAIL	email	D.3.1.11
ebRECIPIENTFIRSTNAME	forename	D.3.1.11
ebRECIPIENTLASTNAME	surname	D.3.1.11
ebRECIPIENTMIDNAME	middlename	D.3.1.11

ebRECIPIENTPHONE	telephone	D.3.1.11
ebRECIPIENTSALUTATION	salutation	D.3.1.11
ebRECIPIENTSTATE	region	D.3.1.1
ebRECIPIENTSTREET	streetaddress	D.3.1.1
ebRECIPIENTZIPCD	postcode	D.3.1.1
ebREGLOYALTY	loyalty	D.3.1.5
ebREGPROMOS	promos	D.3.1.5
ebRETURNALLOWED	returns_allowed	D.3.1.9
ebSHIPADDRESS2	moreaddress	D.3.1.1
ebSHIPAPT	apartment	D.3.1.1
ebSHIPCITY	city	D.3.1.1
ebSHIPCOMMENTS	comments	D.3.1.12
ebSHIPCOUNTRY	country	D.3.1.1
ebSHIPEMAIL	email	D.3.1.11
ebSHIPFIRSTNAME	forename	D.3.1.11
ebSHIPINSTRUCTION	instructions	D.3.1.12
ebSHIPLASTNAME	surname	D.3.1.11
ebSHIPMENTNO	tracking_number	D.3.1.12
ebSHIPMETHOD	method	D.3.1.12
ebSHIPMIDNAME	middlename	D.3.1.11
ebSHIPPHONE	telephone	D.3.1.11
ebSHIPSALUTATION	salutation	D.3.1.11
ebSHIPSTATE	region	D.3.1.1
ebSHIPSTREET	streetaddress	D.3.1.1
ebSHIPZIPCD	postcode	D.3.1.1
ebSUBTOTAL	subtotal	D.3.1.10
ebTAX	tax	D.3.1.10
ebTRANCATEGORY	transactionsource	D.3.1.13
ebUPC	attribute upc	D.3.1.8
ebWEBSITE	website	D.3.1.5

I.3. Real Time Fraud Screening - Gift Occasions

The Real Time Fraud Screening Service is outlined in section D.3

1 - Celebrate Fall 2 - Grandparent's Day 4 - Independence Day A - Anniversary B - Birthday C - Congratulations D - April Fool's Day E - Easter F - Father's Day G - Graduation H - Holiday I - Season's Greeting J - Passover K - Kwanzaa L - Halloween	M - Mother's Day N - New Year's Day O - Bosses' Day P - St Patrick's Day Q - Sweetest Day R - Christmas S - Baby Shower T - Thanksgiving U - Other V - Valentine's Day W - Wedding X - Secretary's Day Y - Chinese New Year Z - Hanukkah