Internet Open Trading Protocol
Part 2: Specification

Version 0.9
Date 12 January 1998
Status Draft for Public Comment

Note. The intention is that this document will, in due course, be submitted to an appropriate International Standards Authority.
Copyright

© The producers of the OTP Specification have a copyright on this material. Copying and/or duplicating this material is allowed but using the material for any purpose other than contemplating, considering or actually developing an OTP based capability is not allowed. There are plans for an OTP Service Mark and only those who respect the guidelines outlined above will be allowed to apply for an OTP Service Mark for OTP based products or services they develop.
## History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 0.1</td>
<td>20 February 1997</td>
<td>Initial draft for comment</td>
</tr>
<tr>
<td>Version 0.2</td>
<td>14 April 1997</td>
<td>Revised draft including changes arising from comments</td>
</tr>
<tr>
<td>Version 0.2a</td>
<td>24 April 1997</td>
<td>Same as version 0-2 with typographic corrections</td>
</tr>
<tr>
<td>Version 0.3</td>
<td>9 October 1997</td>
<td>Revised draft for comment including revised encoding approach using [XML]</td>
</tr>
<tr>
<td>Version 0.4</td>
<td>31 October 1997</td>
<td>Published draft for limited public review by groups working within OTP dev</td>
</tr>
<tr>
<td>Version 0.9</td>
<td>12 January 1998</td>
<td>Revisions following limited public review – may be used for prototype development</td>
</tr>
<tr>
<td>Version 1.0</td>
<td>Planned 1Q98</td>
<td>Published draft for unlimited public review and use. Production pilots expected from this release</td>
</tr>
</tbody>
</table>
This Version 0.9 of the document is for public comment which should be sent to otp-forum@lists.commerce.net

All email sent to this list will be archived in the Public area of the OTP website http://www.otp.org

Received comments will be incorporated and product developers are advised to expect a revised Version 1.0 by the end of 1Q98.

Any products built from the Version 0.9 and/or the Version 1.0 are expected to prove function and live interoperability.

Therefore it may be necessary for early released products developed with this Version 0.9 to be changed for compatibility with the Version 1.0 of this protocol.

Any responsibility or liability for such changes is expressly disclaimed.
Preface

The interest in electronic commerce is everywhere. As merchants, financial institutions, technology providers and newcomers scramble to understand the market and establish their presence it is clear that there is little overall cohesion to the myriad efforts.

There have been some co-ordinated activities that while not guaranteeing success have demonstrated the benefits of a co-operative multi-enterprise effort to produce something for the advancement of this phenomenon known as electronic commerce.

Examples are the development of:

- SET – Secure Electronic Transaction from MasterCard, Visa, American Express etc
- JEPI – Joint Electronic Payment Initiative from CommerceNet and W3C (the World Wide Web Consortium)
- EMV - Debit/credit cards using chip technology from Europay, MasterCard and Visa
- E-Check – Electronic Checkbook on a smart card from the US based Financial Services Technology Consortium and FSTC members; Federal Reserve Bank, NationsBank, Bank of Boston, Huntington Bancshares, IBM and Sun Microsystems and others.

OTP shares many of the high level objectives of these efforts and in part builds on some of them. The OTP specification provides a unifying framework within which these and others not named or even contemplated can exist and successfully interoperate.

OTP seeks to provide the virtual capability that safely replicates the real world, the paper based, traditional, understood, accepted methods of trading, buying, selling, value exchanging that has existed for many hundreds of years. The negotiation of who will be the parties to the trade, how it will be conducted, the presentment of an offer, the method of payment, the provision of a payment receipt, the delivery of goods and the receipt of goods. These are events that are taken for granted in the course of our real world and OTP has been produced to provide the same for the virtual world, and to prepare and provide for the introduction of new models of trading made possible by the expanding presence of the virtual world.

The other fundamental ideal of this effort is to produce a definition of these trading events in such a way that no matter where produced, two unfamiliar parties using electronic commerce capabilities to buy and sell that conform to the OTP specifications will be able to complete the business safely and successfully.

In summary, OTP supports:

- Familiar trading models
- New trading models
- Global interoperability
Commerce on the Internet – a Different Model

The growth of the Internet and the advent of electronic commerce are bringing about enormous changes around the world in society, politics and government, and in business. The ways in which trading partners communicate, conduct commerce, are governed have been enriched and changed forever.

One of the very fundamental changes about which OTP is concerned is taking place in the way consumers and merchants trade. Characteristics of trading that have changed markedly include:

- **Presence**: Face-to-face transactions become the exception, not the rule. Already with the rise of mail order and telephone order placement this change has been felt in western commerce. Electronic commerce over the Internet will further expand the scope and volume of transactions conducted without ever seeing the people who are a part of the enterprise with whom one does business.

- **Authentication**: An important part of personal presence is the ability of the parties to use familiar objects and dialogue to confirm they are who they claim to be. The seller displays one or several well known financial logos that declaim his ability to accept widely used credit and debit instruments in the payment part of a purchase. The buyer brings government or financial institution identification that assures the seller she will be paid. People use intangibles such as personal appearance and conduct, location of the store, apparent quality and familiarity with brands of merchandise, and a good clear look in the eye to reinforce formal means of authentication.

- **Payment instruments**: Despite the enormous size of bank card financial payments associations and their members, most of the world’s trade still takes place using the coin of the realm or barter. The present infrastructure of the payments business cannot economically support low value transactions and could not survive under the consequent volumes of transactions if it did accept low value transactions.

- **Transaction values**: New meaning for low value transactions arises in the Internet where sellers may wish to offer for example, pages of information for fractions of currency that do not exist in the real world.

- **Delivery**: New modes of delivery must be accommodated such as direct electronic delivery. The means by which receipt is confirmed and the execution of payment change dramatically where the goods or services have extremely low delivery cost but may in fact have very high value. Or, maybe the value is not high, but once delivery occurs the value is irretrievably delivered so payment must be final and non-refundable but delivery nonetheless must still be confirmed before payment. Incremental delivery such as listening or viewing time or playing time are other models that operate somewhat differently in the virtual world.

Benefits of OTP

*Electronic Commerce Software Vendors*

Electronic Commerce Software Vendors will be able to develop e-commerce products which are more attractive as they will inter-operate with any other vendors’ software. However since OTP focuses on how these solutions communicate, there is still plenty of opportunity for product differentiation.
Payment Brands

OTP provides a standard framework for encapsulating payment protocols. This means that it is easier for payment products to be incorporated into OTP solutions. As a result the payment brands will be more widely distributed and available on a wider variety of platforms.

Merchants

There are several benefits for Merchants:

• they will be able to offer a wider variety of payment brands,
• they can be more certain that the customer will have the software needed to complete the purchase
• through receiving payment and delivery receipts from their customers, they will be able to provide customer care knowing that they are dealing with the individual or organisation with which they originally traded
• new merchants will be able to enter this new (Internet) market-place with new products and services, using the new trading opportunities which OTP presents

Banks and Financial Institutions

There are also several benefits for Banks and Financial Institutions:

• they will be able to provide OTP support for merchants
• they will find new opportunities for OTP related services:
  - providing customer care for merchants
  - fees from processing new payments and deposits
• they have an opportunity to build relationships with new types of merchants

Customers

For Customers there are several benefits:

• they will have a larger selection of merchants with whom they can trade
• there is a more consistent interface when making the purchase
• there are ways in which they can get their problems fixed through the merchant (rather than the bank!)
• there is a record of their transaction which can be used, for example, to feed into accounting systems or, potentially, to present to the tax authorities

Baseline OTP

The team working on the OTP see an extended versions of this specification being developed but at this stage feel a need to develop a limited function specification in order that technology providers can soon develop pathway-pilot\(^1\) products that will be placed in the market in order to

\(^1\) “Pathway-Pilot” is used to draw a distinction between pilot products that within our industry are often considered to be temporary with set end dates and our intent to build off these early products as the specification is extended.
understand the real “market place” demands and requirements for electronic trading or electronic commerce. To proceed otherwise would be presumptuous, time consuming, expensive and foolish.

Accordingly the OTP Baseline specification has been produced for pathway-pilot product development, expecting to transact live trades over the open networks by mid ’98.

At the same time that the pathway-pilot products are being developed and brought to market the developers of the OTP specification will be studying and working on the protocol to both preserve the original work (and the initial investment made by early technology suppliers) and to enhance the specification.

An area that is most likely to be extended is the “Trading Exchanges” as it is here that many new possibilities and options are expected: the ability to make multiple payments in one purchase, the ability to pay for units of time connected to a service, the ability to earn and or use loyalty points all are recognised and need to be brought into a fuller and extended OTP while at the same time preserving the initial specification.

Objectives of Document

The objectives of this document are to provide a functional specification of version 0.9 of the Open Trading Protocols which can be used to design and implement systems which support electronic trading on the Internet using the Open Trading Protocols.

An overview of OTP is provided by Part 1: Business Description and should be read first.

Purpose

The purpose of the document is:

• to allow potential developers of products based on the protocol to:
  - comment on the applicability of the protocol to their environment,
  - start development of software/hardware solutions which use the protocol\(^2\).

• to allow the financial services industry to understand a developing electronic commerce trading protocol that encapsulates (without modification) any of the current or developing payment schemes now being used or considered by their merchant customer base.

• to describe the intent of the protocol developers to submit the OTP for consideration as the basis for a governed industry standard for conducting electronic commerce which is independent of the payment method.

Scope of Document

The protocol describes the content, format and sequences of messages that pass among the participants in an electronic trade - consumers, merchants and banks or other financial

\(^2\) Please see disclaimer at the start of this document.
Preface

institutions. These are required to support the electronic commerce transactions outlined in the objectives above.

The protocol is designed to be applicable to any electronic payment scheme\(^3\) since it targets the complete purchase process where the movement of electronic value from the payer to the payee is only one, but important, step of many that may be involved to complete the trade.

Each payment scheme contains some message flows which are specific to that scheme. These scheme-specific parts of the protocol are contained in a set of payment scheme supplements of this standard.

The document does not prescribe the software and processes that will need to be implemented by each participant. It does describe the framework necessary for trading to take place.

This document also does not address any legal or regulatory issues surrounding the implementation of the protocol or the information systems which use them.

Intended Readership

Software and hardware developers; development analysts; business and technical planners; industry analysts; merchants; bank and other value providers; owners, custodians, and users of payment protocols.

\(^3\) A Payment Scheme is a method of making a payment such as MasterCard Credit, Visa Credit, Mondex Cash, Visa Cash, GeldKarte, DigiCash, Cybercoin, Millicent, Proton etc. where rules apply which define the methods, policies and processes used to make payments.
Related Documents

This section contains the descriptions of related documents identified in this specification.


[DSA] The Digital Signature Algorithm (DSA) published by the National Institute of Standards and Technology (NIST) in the Digital Signature Standard (DSS), which is a part of the U.S. government's Capstone project.

[ECCDSA] Elliptic Curve Cryptosystems Digital Signature Algorithm (ECCDSA). Elliptic curve cryptosystems are analogs of public-key cryptosystems such as RSA in which modular multiplication is replaced by the elliptic curve addition operation. See: V.S. Miller. Use of elliptic curves in cryptography. In Advances in Cryptology - Crypto '85, pages 417-426, Springer-Verlag, 1986.

[HTML] Hyper Text Mark Up Language. The Hypertext Markup Language (HTML) is a simple markup language used to create hypertext documents that are platform independent. See RFC 1866 and the World Wide Web (W3C) consortium web site at: http://www.w3.org/MarkUp/


[ISO4217] ISO 4217: Codes for the Representation of Currencies. Available from ANSI or ISO.


[OPS] Open Profiling Standard. A proposed standard which provides a framework with built-in privacy safeguards for the trusted exchange of profile information between individuals and websites. Being developed by Netscape and Microsoft amongst others.


[UTC] Universal Time Coordinated. A method of defining time absolutely relative to Greenwich Mean Time (GMT). Typically of the form: "CCYY-MM-DDTHH:MM:SS.SSSZ+n" where the "+n" defines the number of hours from GMT. See ISO DIS8601.


[XML Namespace] See Design decisions reached at the XML WG meeting in Montreal, Canada, August 22, 1987

Acknowledgements

This protocol was developed primarily by the following organisations:

- AT&T Universal Card Services
- AT&T Unisource
- British Telecom
- Canadian Imperial Bank of Commerce
- CyberCash
- DigiCash
- DOT Matrix
- Fujitsu
- Hitachi
- Hewlett Packard
- General Information Systems
- IBM
- InterTrader
- JCP Inc
- MasterCard International
- Mercantec
- Mondex International
- Netscape/Actra Corp
- Oracle Corporation
- Royal Bank of Canada
- SIZ Computer Science Center of the German Savings Banks
- Smart Card Integrations Ltd.
- Spyrus
- Sun Microsystems
- Verifone
- Wells Fargo Bank
# Table of Contents

1. Introduction .................................................................................................................. 1  
   1.1 Trading Roles ........................................................................................................... 2  
1.2 Trading Exchanges ..................................................................................................... 3  
   1.2.1 Offer Exchange .................................................................................................... 4  
   1.2.2 Payment Exchange .............................................................................................. 6  
   1.2.3 Delivery Exchange .............................................................................................. 8  
   1.2.4 Authentication Exchange .................................................................................... 9  
1.3 Scope of Baseline OTP .............................................................................................. 10  

2. Protocol Structure ....................................................................................................... 12  
   2.1 Overview ................................................................................................................ 13  
   2.2 OTP Message .......................................................................................................... 15  
   2.3 Trans Ref Block ....................................................................................................... 16  
   2.4 ID Attributes .......................................................................................................... 18  
   2.5 Element References .............................................................................................. 20  
   2.6 Signature Overview ............................................................................................... 22  
   2.7 Extending OTP ....................................................................................................... 23  
   2.8 Identifying Languages ........................................................................................... 27  

3. Trading Components .................................................................................................... 28  
   3.1 Protocol Options Component .................................................................................. 29  
   3.2 Authentication Data Component ........................................................................... 31  
   3.3 Authentication Response Component ...................................................................... 31  
   3.4 Order Component .................................................................................................. 32
Preface

3.5 Organisation Component ................................................................. 33
   3.5.1 Organisation Role Element ..................................................... 34
   3.5.2 Contact Information Element ................................................ 35
   3.5.3 Person Name Element ............................................................ 35
   3.5.4 Postal Address Element ........................................................ 36

3.6 Brand List Component ................................................................. 36
   3.6.1 Brand Element ................................................................. 37
   3.6.2 Pay Protocol Element .......................................................... 39

3.7 Brand Selection Component .......................................................... 40

3.8 Pay Amount Component ............................................................... 42

3.9 Payment Scheme Component ......................................................... 43

3.10 Pay Receipt Component ............................................................... 44

3.11 Delivery Component ................................................................. 44
   3.11.1 DelivData Element ............................................................. 45

3.12 Delivery Note Component ........................................................... 47

3.13 Signature Component ................................................................. 48
   3.13.1 SignedData Element ........................................................... 49
   3.13.2 Hash Element ................................................................. 49
   3.13.3 Authenticated Attribute Element ........................................... 50
   3.13.4 Digital Signature Element ................................................... 51
   3.13.5 Unauthenticated Attribute Element ....................................... 52
   3.13.6 Hashes and Signatures ....................................................... 53
   3.13.7 Checking Hashes and Signatures ........................................... 56

3.14 Certificate Component ............................................................... 57

3.15 Fail Reason Component .............................................................. 57
   3.15.1 Fail Text Parameter Element ............................................... 58

3.16 Fail OTP Message Component ...................................................... 59

4. Trading Blocks .................................................................................. 60
   4.1 Trading Protocol Options Block .................................................. 61
   4.2 TPO Selection Block .................................................................. 61
   4.3 Offer Response Block .................................................................. 62
   4.4 Authentication Request Block ..................................................... 63
   4.5 Authentication Response Block .................................................. 63
   4.6 Payment Request Block .............................................................. 63
   4.7 Payment Exchange Block .......................................................... 64
   4.8 Payment Response Block ............................................................ 65
### 5. Open Trading Protocol Transactions

5.1 Baseline Authentication OTP Transaction ................................................................. 68
   5.1.1 TPO (Trading Protocol Options) Block ................................................................. 69
   5.1.2 Authentication Request Block .............................................................................. 69
   5.1.3 Authentication Response Block .............................................................................. 70

5.2 Baseline Deposit OTP Transaction ................................................................................ 70
   5.2.1 Baseline Deposit Variations ...................................................................................... 70
   5.2.2 Baseline Deposit Authentication ............................................................................. 71
   5.2.3 Baseline Deposit Payment Messages ...................................................................... 73
   5.2.4 TPO (Trading Protocol Options) Block ................................................................. 73
   5.2.5 TPO Selection Block ............................................................................................... 74
   5.2.6 Authentication Request Block ................................................................................. 74
   5.2.7 Authentication Response Block .............................................................................. 74
   5.2.8 Offer Response Block ............................................................................................. 74
   5.2.9 Payment Request Block ........................................................................................ 75
   5.2.10 Payment Exchange Block .................................................................................... 76
   5.2.11 Payment Response Block ..................................................................................... 76

5.3 Baseline Purchase OTP Transaction ............................................................................ 76
   5.3.1 Baseline Purchase Variations .................................................................................. 77
   5.3.2 TPO (Trading Protocol Options) Block ................................................................. 84
   5.3.3 TPO Selection Block ............................................................................................... 84
   5.3.4 Offer Response Block ............................................................................................. 84
   5.3.5 Payment Request Block ........................................................................................ 85
   5.3.6 Payment Exchange Block .................................................................................... 86
   5.3.7 Payment Response Block ..................................................................................... 86
   5.3.8 Delivery Request Block ......................................................................................... 86
   5.3.9 Delivery Response Block ...................................................................................... 87

5.4 Baseline Refund OTP Transaction ................................................................................ 87
   5.4.1 Baseline Refund Variations ...................................................................................... 88
   5.4.2 Baseline Refund Authentication ............................................................................. 88
   5.4.3 Baseline Refund Payment Messages ...................................................................... 90
   5.4.4 TPO (Trading Protocol Options) Block ................................................................. 90
   5.4.5 TPO Selection Block ............................................................................................... 91
   5.4.6 Authentication Request Block ................................................................................. 91
   5.4.7 Authentication Response Block .............................................................................. 91
   5.4.8 Offer Response Block ............................................................................................. 91
   5.4.9 Payment Request Block ........................................................................................ 92
   5.4.10 Payment Exchange Block ................................................................................... 93
   5.4.11 Payment Response Block ..................................................................................... 93

5.5 Baseline Withdrawal OTP Transaction ....................................................................... 93
   5.5.1 Baseline Withdrawal Variations ............................................................................. 94
Preface

5.5.2 Baseline Withdrawal Authentication ................................ ................................ ............................. 94
5.5.3 Baseline Withdrawal Payment Messages................................................................................. 97
5.5.4 TPO (Trading Protocol Options) Block.................................................................................. 97
5.5.5 TPO Selection Block ................................................................................................................. 98
5.5.6 Authentication Request Block .................................................................................................. 98
5.5.7 Authentication Response Block ............................................................................................... 98
5.5.8 Offer Response Block ............................................................................................................... 98
5.5.9 Payment Request Block ........................................................................................................... 99
5.5.10 Payment Exchange Block ...................................................................................................... 100
5.5.11 Payment Response Block ..................................................................................................... 100

5.6 Baseline Value Exchange OTP Transaction ............................................................................ 100
5.6.1 Baseline Value Exchange Variations ..................................................................................... 101
5.6.2 TPO (Trading Protocol Options) Block .................................................................................. 104
5.6.3 TPO Selection Block ................................................................................................................. 105
5.6.4 Offer Response Block ............................................................................................................... 105
5.6.5 Payment Request Block (first payment) .................................................................................. 106
5.6.6 Payment Exchange Block (first payment) ............................................................................... 107
5.6.7 Payment Response Block (first payment) ............................................................................... 107
5.6.8 Payment Request Block (second payment) .............................................................................. 107
5.6.9 Payment Exchange Block (second payment) .......................................................................... 108
5.6.10 Payment Response Block (second payment) ......................................................................... 108
5.6.11 Baseline Value Exchange Signatures ..................................................................................... 109

5.7 Restart Transaction (provisional) ............................................................................................... 110

6. Terminating and Resending OTP Transactions ................................ .............................. 111

6.1 Terminating OTP Transactions ............................................................................................... 111
   6.1.1 Consumer Termination of Transactions ............................................................................... 111
   6.1.2 Non-consumer Termination of Transactions ....................................................................... 111

6.2 Resending OTP Messages ........................................................................................................ 112

Appendix A - XML Name Summary ............................................................................................ 114

Appendix B - Failure Codes ........................................................................................................... 115

Appendix C - XML Overview .......................................................................................................... 116

   Document Definition ......................................................................................................................... 116
   Element Declaration ............................................................................................................................. 117
   Attribute declarations .......................................................................................................................... 118

Appendix D - Open Trading Protocol Formal Definition ......................................................... 120
# Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OTP Trading Roles</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Offer Exchange</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Payment Exchange</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>Delivery Exchange</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Authentication Exchange</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>OTP Message Structure</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>An OTP Transaction</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>Example use of OTPMsgId, BlkId and Compld attributes</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>Element References</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>Signature Overview</td>
<td>23</td>
</tr>
<tr>
<td>11</td>
<td>Extra XML Elements</td>
<td>26</td>
</tr>
<tr>
<td>12</td>
<td>Trading Components</td>
<td>28</td>
</tr>
<tr>
<td>13</td>
<td>Signature Component</td>
<td>48</td>
</tr>
<tr>
<td>14</td>
<td>What to Sign</td>
<td>53</td>
</tr>
<tr>
<td>15</td>
<td>Generating Canonical Encoding Format</td>
<td>54</td>
</tr>
<tr>
<td>16</td>
<td>Trading Blocks</td>
<td>60</td>
</tr>
<tr>
<td>17</td>
<td>Baseline Authentication</td>
<td>69</td>
</tr>
<tr>
<td>18</td>
<td>Baseline Deposit with Authentication</td>
<td>71</td>
</tr>
<tr>
<td>19</td>
<td>Baseline Deposit without Authentication</td>
<td>72</td>
</tr>
<tr>
<td>20</td>
<td>Baseline Deposit Payment Messages</td>
<td>73</td>
</tr>
<tr>
<td>21</td>
<td>Brand Dependent Baseline Purchase</td>
<td>78</td>
</tr>
<tr>
<td>22</td>
<td>Brand Independent Baseline Purchase</td>
<td>79</td>
</tr>
<tr>
<td>23</td>
<td>Baseline Purchase, Delivery Response Block and Pay Response Blocks Not Combined</td>
<td>80</td>
</tr>
<tr>
<td>24</td>
<td>Baseline Purchase, Delivery Response Block and Pay Response Block Combined</td>
<td>81</td>
</tr>
<tr>
<td>25</td>
<td>Baseline Purchase, Purchase without Delivery Exchange</td>
<td>82</td>
</tr>
<tr>
<td>26</td>
<td>Baseline Purchase Variations</td>
<td>83</td>
</tr>
<tr>
<td>27</td>
<td>Baseline Refund with Authentication</td>
<td>88</td>
</tr>
<tr>
<td>28</td>
<td>Baseline Refund without Authentication</td>
<td>89</td>
</tr>
<tr>
<td>29</td>
<td>Baseline Refund Payment Messages</td>
<td>90</td>
</tr>
<tr>
<td>30</td>
<td>Baseline Withdrawal with Authentication</td>
<td>95</td>
</tr>
<tr>
<td>31</td>
<td>Baseline Withdrawal without Authentication</td>
<td>96</td>
</tr>
<tr>
<td>32</td>
<td>Baseline Withdrawal Payment Messages</td>
<td>97</td>
</tr>
<tr>
<td>33</td>
<td>Baseline Value Exchange</td>
<td>102</td>
</tr>
<tr>
<td>34</td>
<td>Brand Dependent Value Exchange</td>
<td>103</td>
</tr>
<tr>
<td>35</td>
<td>Baseline Value Exchange Payment Messages</td>
<td>104</td>
</tr>
<tr>
<td>36</td>
<td>Baseline Value Exchange Signatures</td>
<td>109</td>
</tr>
</tbody>
</table>
1. Introduction

The Open Trading Protocols (OTP) define a number of different types of OTP Transactions:

- **Purchase.** This supports a purchase involving an offer, a payment and optionally a delivery
- **Refund.** This supports the refund of value as a result of, typically, an earlier purchase
- **Value Exchange.** This involves two payments which result in the exchange of value from one combination of currency and payment method to another
- **Authentication.** This supports the authentication of one party to make sure that another party is who they appear to be using a challenge-response mechanism
- **Withdrawal.** This supports the withdrawal of electronic cash from a financial institution
- **Deposit.** This supports the deposit of electronic cash at a financial institution

These **OTP Transactions** are "Baseline" transactions since they have been identified as a minimum useful set of transactions. Later versions of OTP may include additional types of transactions\(^4\).

Each of the OTP Transactions above involve:

- a number organisations playing a **Trading Role**, and
- a set of **Trading Exchanges.** Each Trading Exchange involves the exchange of data, between Trading Roles, in the form of a set of **Trading Components.**

Trading Roles, Trading Exchanges and Trading Components are described below.

\(^4\) One of the later transactions will be a "Transaction Restart" OTP transaction. This will provide a general facility for restarting any OTP Transaction which had previously failed, for example, due to a "timeout" of the transport mechanism. This will be particularly useful for the Purchase Transaction if the transaction failed after payment was made but before delivery had been completed. Similarly for the Value Exchange Transaction it will be useful if the transaction fails between the first and second payments.
1.1 Trading Roles

The Trading Roles identify the different parts which organisations can take in a trade. The five Trading Roles used within OTP are illustrated in the diagram below.

![Diagram of OTP Trading Roles]

The roles are:

- **Consumer.** The person or organisation which is to receive and pay for the goods or services.
- **Merchant.** The person or organisation from whom the purchase is being made and who is legally responsible for providing the goods or services and receives the benefit of the payment made.
- **Value Acquirer.** The entity that physically receives the payment from the Consumer on behalf of the Merchant.
- **Deliverer.** The entity that physically delivers the goods or services to the Consumer on behalf of the Merchant.
- **Customer Care Provider.** The entity that is involved with customer dispute negotiation and resolution on behalf of the Merchant.

Roles may be carried out by the same organisation or different organisations. For example:

- in the simplest case one physical organisation (e.g. a merchant) could handle the purchase, accept all the value, deliver the goods and provide customer care.
- at the other extreme, a merchant could handle the purchase but instruct the consumer to pay a bank or financial institution, request that delivery be made by an
overnight courier firm and to contact an organisation which provides 24x7 service if problems arise.

Note that in this specification, unless stated to the contrary, when the words Consumer, Merchant, Value Acquirer, Deliverer or Customer Care Provider are used, they refer to the Trading Role rather than an actual organisation.

As roles occur in different places there is a need for the organisations involved in the trade to exchange data, i.e. to carry out Trading Exchanges, so that the trade can be completed.

1.2 Trading Exchanges

The Open Trading Protocols identify four Trading Exchanges which involve the exchange of data between the Trading Roles. The Trading Exchanges are:

- **Offer.** The Offer Exchange results in the Merchant providing the Consumer with the reason why the trade is taking place. It is called an Offer since the Consumer must accept the Offer if a trade is to continue

- **Payment.** The Payment Exchange results in value of some kind being transferred between the Consumer and the Value Acquirer. This may occur in either direction

- **Delivery.** The Delivery Exchange transmits either the on-line goods, or delivery information about physical goods from the Deliverer to the Consumer, and

- **Authentication.** The Authentication Exchange can be used by any Trading Role to authenticate another Trading Role to check that they are who they appear to be.

OTP Transactions are composed of various combinations of these Trading Exchanges. For example, an OTP **Purchase** transaction includes **Offer**, **Payment**, and **Delivery** Trading Exchanges. As another example, an OTP **Value Exchange** transaction is composed of an **Offer** Trading Exchange and two **Payment** Trading Exchanges.

Trading Exchanges consist of Trading Components that are transmitted between the various Trading Roles. Where possible, the number of round-trip delays in an OTP Transaction is minimised by packing the Components from several Trading Exchanges into combination OTP messages. For example, the OTP **Purchase** transaction combines a Delivery Organisation component with an Offer Response component in order to avoid an extra Consumer request and response.

Each of the OTP Trading Exchanges is described in more detail below. For clarity of description, these describe the Trading Exchanges as though they were standalone operations. For performance reasons, the Trading Exchanges are intermingled in the actual OTP transaction definitions.
1.2.1 Offer Exchange

The goal of the Offer Exchange is for the Merchant to provide the Consumer with information about the trade so that the Consumer can decide whether to continue with the trade. This is illustrated in the diagram below.

An Offer Exchange uses the following Trading Components that are passed between the Consumer and the Merchant:

- the Organisation Component contains information which describes the organisations which are taking a role in the trade:
  - the consumer provides information, about who the consumer is and, if goods or services are being delivered, where the goods or services are to be delivered to
  - the merchant augments this information by providing information about the merchant, the value acquirer, the customer care provider and, if goods or services are being delivered, the deliverer

- the Order Component contains descriptions of the goods or services which will result from the trade if the consumer agrees to the offer. This information is sent by the Merchant to the consumer who should verify it
Chapter 1 Introduction

- the Pay Amount Component generated by the Merchant, contains details of how much to pay, the currency and the payment direction, for example the consumer could be asking for a refund. Note that there may be more than one payment in a trade.
- the Delivery Component, also generated by the Merchant, is used if goods or services are being delivered. This contains information about how delivery will occur, for example by post or using e-mail.
- the "Offer" Signature Component, if present, digitally signs all of the above components to ensure their integrity.

The exact content of the information provided by the Merchant to the Consumer will vary depending on the type of OTP Transaction. For example:

- low value purchases may not need a signature
- the amount to be paid may vary depending on the payment brand and payment protocol used
- some offers may not involve the delivery of any goods
- a value exchange will involve two payments
- a merchant may not offer customer care.

Information provided by the consumer to the merchant could be provided using a variety of methods, for example, it could be provided:

- using [HTML] pages as part of the "shopping experience" of the consumer.
- using the Open Profiling Standard [OPS] which has recently been proposed,
- in the form of Organisation and Order Components in a later version of OTP.
1.2.2 Payment Exchange

The goal of the Payment Exchange is for a payment to be made from the Consumer to a Value Acquirer or vice versa using a payment brand and payment protocol selected by the Consumer. A secondary goal is to optionally provide the Consumer with a digitally signed Payment Receipt which can be used to link the payment to the reason for the payment as described in the Offer Exchange.

Payment Exchanges can work in a variety of ways. The most general case where the trade is dependent on the payment brand and protocol used is illustrated in the diagram below. Simpler payment exchanges are possible.

A Payment Exchange uses the following Trading Components that are passed between the Consumer, the Merchant and the Value Acquirer:

- The Brand List Component contains a list of payment brands (for example, MasterCard, Visa, Mondex, GeldKarte) and payment protocols (for example SET Version 1.0, Secure Channel Credit Debit⁵). The Merchant sends the Brand List to

---

⁵ Secure Channel Credit Debit (SCCD), is the name used for a credit or debit card payment where unauthorised access to account information is prevented through use of secure channel transport mechanisms such as SSL.
Chapter 1 Introduction

the Consumer. The consumer compares the payment brands and protocols on offer with those that the Consumer supports and makes a selection.

- The Brand Selection Component contains the Consumer's selection. Payment brand, protocol and possibly protocol-specific information is sent back to the Merchant. This information may be used to change information in the Offer Exchange. For example, a merchant could choose to offer a discount to encourage the use of a store card.

- The Organisation Components are generated by the Merchant. They contain details of the Merchant and Value Acquirer Roles:
  - the Merchant role is required so that the Value Acquirer can identify which Merchant initiated the payment. Typically, the result of the Value Acquirer accepting (or making) a payment on behalf of the Merchant will be a credit or debit transaction to the Merchant's account held by the Value Acquirer. These transactions are outside the scope of OTP
  - the Value Acquirer role is required so that the Value Acquirer can check that it is the correct Value Acquirer to be used for the payment

- The optional Authentication Data Component contains challenge data which is used by the payment protocol to authenticate the consumer. Authentication may not always occur

- The Pay Amount Component contains details of how much to pay, the currency and the payment direction, and identifies the Authentication Data Component to use.

- The "Offer" Signature Component, if present, digitally signs all of the above components to ensure their integrity. Note that the Brand List and Brand Selection Components are not signed until the payment information is created (step 3 in the diagram)

- The Pay Scheme Component contains messages from the payment protocol used in the Trade. For example they could be SET messages, Mondex messages, GeldKarte Messages or one of the other payment methods supported by OTP. The content of the Pay Scheme Component is defined in the supplements that describe how OTP works with various payment protocols.

- The Pay Receipt Component contains a record of the payment. The content depends upon the payment protocol used.

- The "Pay Receipt" Signature Component provides proof of payment by digitally signing both the Pay Receipt Component and the Offer Signature. The signature on the offer digitally signs the Order, Organisation and Delivery Components contained in the Offer. This signature effectively binds the payment to the offer.

The example of a Payment Exchange above is the most general case. Simpler cases are also possible. For example, if the amount paid is not dependent on the payment brand and protocol selected then the payment information generated by step 3 can be sent to the Consumer at the same time as the Brand List Component generated by step 1. These and other variations are described in the Baseline Purchase OTP Transaction (see section 5.3).
1.2.3 Delivery Exchange

The goal of the Delivery Exchange is to cause purchased goods to be delivered to the consumer either online or via physical delivery. A second goal is to provide a "delivery note" to the consumer, providing details about the delivery, such as shipping tracking number. A future goal is to have a signed delivery that can be used for customer care in the case of problems with physical delivery. This is illustrated in the diagram below.

A Delivery Exchange uses the following Trading Components that are passed between the Consumer, the Merchant and the Deliverer:

- The Organisation Component(s) contain details of the Deliver To, Deliverer and Merchant Roles:
  - the Deliver To role indicates where the goods or services are to be delivered to
  - the Deliverer role is required so that the Deliverer can check that she is the correct Deliverer to do the delivery
  - the Merchant role is required so that the Deliverer can identify which Merchant initiated the delivery

- The Order Component, contains information about the goods or services to be delivered

- The Delivery Component contains information about how delivery will occur, for example by post or using e-mail.

- The "Offer" Signature Component, if present, digitally signs all of the above components to ensure their integrity.
Chapter 1 Introduction

- The "Pay Receipt" Signature Component provides proof of payment by digitally signing the Pay Receipt Component and the Offer Signature. This is used by the deliverer to check that delivery is authorised.

- The Delivery Note Component contains customer care information related to a physical delivery, or alternatively the actual "electronic goods". The Consumer’s software does not interpret information about a physical delivery but should have the ability to display the information, both at the time of the delivery and later if the Consumer selects the Trade to which this delivery relates from a transaction list.

1.2.4 Authentication Exchange

The goal of the Authentication Exchange is to allow one organisation, for example a financial institution, to be able to check that another organisation, for example a consumer, is who they appear to be. It uses a "challenge-response" mechanism. This is illustrated in the diagram below.

![Authentication Exchange Diagram]

An Authentication Exchange uses the following Trading Components that are passed between the two organisations:

- the Authentication Data Component which contains the challenge data to be used in the "challenge-response" mechanism and indicates the authentication method to be used. It is sent by one organisation to the other.

- the Authentication Response Component which contains the challenge response generated by the recipient of the Authentication Data Component. It is sent back to the first organisation for verification.
1.3 Scope of Baseline OTP

This specification describes the OTP Transactions which make up Baseline OTP. As described in the preface, OTP will evolve over time. This section defines the initial conformance criteria for implementations that claim to “support OTP.”

The main determinant on the scope of an OTP implementation is the roles which the solution is designed to support. The roles within OTP are described in more detail in section 1.1 Trading Roles. To summarise the roles are: Merchant, Consumer, Value Acquirers, Deliverer and Customer Care Provider.

Value Acquirers who can be of three types:

- those who acquire value as part of a purchase or provide value as part of a refund,
- those who acquire value as part of a deposit transaction, or
- those that issue value via a withdrawal transaction

The following table defines, for each role, the OTP Transactions and Trading Blocks which must be supported for that role.

<table>
<thead>
<tr>
<th>Role</th>
<th>OTP Transactions and Trading Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merchant</td>
<td>Must support the following types of OTP Transactions:</td>
</tr>
<tr>
<td></td>
<td>• Purchase</td>
</tr>
<tr>
<td></td>
<td>• Refund</td>
</tr>
<tr>
<td></td>
<td>Must support the following Trading Blocks:</td>
</tr>
<tr>
<td></td>
<td>• TPO</td>
</tr>
<tr>
<td></td>
<td>• TPO Selection</td>
</tr>
<tr>
<td></td>
<td>• Offer Response</td>
</tr>
<tr>
<td></td>
<td>May support the following OTP Transactions:</td>
</tr>
<tr>
<td></td>
<td>• Authentication</td>
</tr>
<tr>
<td></td>
<td>• Value Exchange</td>
</tr>
<tr>
<td></td>
<td>If the Baseline Authentication OTP Transaction is supported, must support the following Trading</td>
</tr>
<tr>
<td></td>
<td>Blocks:</td>
</tr>
<tr>
<td></td>
<td>• Authentication Request</td>
</tr>
<tr>
<td></td>
<td>• Authentication Response</td>
</tr>
<tr>
<td>Consumer</td>
<td>Must support the following types of OTP Transactions:</td>
</tr>
<tr>
<td></td>
<td>• Purchase</td>
</tr>
<tr>
<td></td>
<td>• Value Exchange</td>
</tr>
<tr>
<td></td>
<td>Must support the following types of OTP Transaction if required by a Payment Method as defined</td>
</tr>
<tr>
<td></td>
<td>in the supplement for the Payment Method:</td>
</tr>
<tr>
<td></td>
<td>• Refund</td>
</tr>
<tr>
<td></td>
<td>• Authentication</td>
</tr>
<tr>
<td></td>
<td>• Withdrawal</td>
</tr>
<tr>
<td></td>
<td>• Deposit</td>
</tr>
<tr>
<td></td>
<td>Must support the following Trading Blocks:</td>
</tr>
<tr>
<td></td>
<td>• TPO</td>
</tr>
<tr>
<td></td>
<td>• TPO Selection</td>
</tr>
<tr>
<td></td>
<td>• Offer Response</td>
</tr>
<tr>
<td></td>
<td>• Payment Request</td>
</tr>
</tbody>
</table>
Chapter 1 Introduction

- Payment Exchange
- Payment Response
- Delivery Request
- Delivery Response

If the Baseline Authentication OTP Transaction is supported, must support the following Trading Blocks:
- Authentication Request
- Authentication Response

<table>
<thead>
<tr>
<th>Value Acquirer (purchase/refund)</th>
<th>Must not support the initiation of any OTP Transactions.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Must support the following Trading Blocks:</td>
</tr>
<tr>
<td></td>
<td>• Payment Request</td>
</tr>
<tr>
<td></td>
<td>• Payment Exchange</td>
</tr>
<tr>
<td></td>
<td>• Payment Response</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value Acquirer (issuing value)</th>
<th>Must support the following types of OTP Transactions:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Authentication</td>
</tr>
<tr>
<td></td>
<td>• Withdrawal</td>
</tr>
</tbody>
</table>

Must support the following Trading Blocks:
- TPO
- TPO Selection
- Offer Response
- Authentication Request
- Authentication Response
- Payment Request
- Payment Exchange
- Payment Response

<table>
<thead>
<tr>
<th>Value Acquirer (acquiring value)</th>
<th>Must support the Deposit OTP Transaction.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>May support the Authentication OTP Transaction</td>
</tr>
</tbody>
</table>

Must support the following Trading Blocks:
- TPO
- TPO Selection
- Offer Response
- Payment Request
- Payment Exchange
- Payment Response

If the Baseline Authentication OTP Transaction is supported, must support the following Trading Blocks:
- Authentication Request
- Authentication Response

<table>
<thead>
<tr>
<th>Deliverer</th>
<th>Must not support the initiation of any OTP Transactions.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Must support the following Trading Blocks:</td>
</tr>
<tr>
<td></td>
<td>• Delivery Request</td>
</tr>
<tr>
<td></td>
<td>• Delivery Response</td>
</tr>
</tbody>
</table>

| Customer Care Provider           | Must not support any OTP Transactions as Customer Care Provider Transactions are not within the scope of Baseline OTP. |

An OTP solution **must** support all the OTP Transactions and Trading Blocks required by **at least one role** as described above in order for that solution to be described as "supporting OTP".
2. Protocol Structure

The previous section provided an introduction which explained:

- **Trading Roles** which are the different roles which organisations can take in a trade: Consumer, Merchant, Value Acquirer, Deliverer and Customer Care Provider, and

- **Trading Exchanges** where each Trading Exchange involves the exchange of data, between Trading Roles, in the form of a set of Trading Components.

This section describes:

- how Trading Components are constructed into Trading Blocks and the OTP Messages which are physically sent in the form of [XML] documents\(^6\) between the different Trading Roles,

- how OTP Messages are exchanged between Trading Roles to create an OTP Transaction

- the XML definitions of an OTP Message including a Trans Ref Block - an XML element which identifies an OTP Transaction and the OTP Message within it

- the definitions of the XML ID Attributes which are used to identify OTP Messages, Trading Blocks and Trading Components and how these are referred to using Element References from other XML elements such as

- OTP Signatures Components which use digital signature techniques to preserve the integrity of OTP Messages and provide the trust relationships required by OTP

- how extra XML Elements and new user defined values for existing OTP codes can be used when Extending OTP, and finally

- how it is planned to use abbreviated versions of XML Element and Attribute Names in version 1.0 of the protocol in order to reduce the physical length of OTP Messages.

\(^6\) See Appendix C XML Overview for a simple explanation of the main features of XML
Chapter 2 Protocol Structure

2.1 Overview

2.1.1 OTP Message Structure

The structure of an OTP Message and its relationship with Trading Blocks and Trading Components is illustrated in the diagram below.

![OTP Message Structure Diagram]

The diagram also introduces the concept of a Trans Ref Block. This block contains, amongst other things a globally unique identifier for the OTP Transaction. Also each block and component is given an ID Attribute (see section 2.4) which is unique within an OTP Transaction. Therefore the combination of the ID attribute and the globally unique identifier in the Trans Ref Block is sufficient to uniquely identify any Trading Block or Trading Component.
2.1.2 OTP Transactions

A predefined set of OTP Messages exchanged between the Trading Roles constitute an **OTP Transaction**. This is illustrated in the diagram below.

In the above diagram the Internet is shown as the transport mechanism. This is not necessarily the case. OTP Messages can be transported using a variety of transport mechanisms.

The OTP Transactions (see section 5) in this version of OTP are specifically:

- **Baseline Purchase.** This supports a purchase involving an offer, a payment and optionally a delivery
- **Baseline Refund.** This supports the refund of value as a result of, typically, an earlier purchase
- **Baseline Value Exchange.** This involves two payments which result in the exchange of value from one combination of currency and payment method to another.
- **Baseline Deposit.** This supports the deposit of electronic cash at a financial institution
- **Baseline Withdrawal.** This supports the withdrawal of electronic cash from a financial institution
- **Baseline Authentication.** This supports the authentication by one organisation of another to make sure that the first organisation is who they appear to be.
2.2 OTP Message

As described earlier, OTP Messages are [XML] documents which are physically sent between the different organisations that are taking part in a trade.

The XML definition of an OTP Message is as follows.

```xml
<!ELEMENT OtpMessage (TransRefBlk,
    ( TpoBlk |
      TpoSelectionBlk
    AuthReqBlk |
    AuthRespBlk |
    OfferRespBlk |
    PayReqBlk |
    PayExchBlk |
    PayRespBlk |
    DelivReqBlk |
    DelivRespBlk |
    FailBlk
  )+ )>
<!ATTLIST OtpMessage
  Status (Original|Resend|Fail) #REQUIRED >
```

Attributes:

- **Status**
  Indicates the reason why the OTP message was sent:
  - **Original** indicates that this is the first time this message has been sent.
  - **Resend** indicates that this message was sent previously but is being resent for some reason (for example, see section 6.2 Resending OTP Message, or)
  - **Fail** indicates that the sender of the message is terminating the OTP transaction (see 4.11 Fail Trading Block). In this instance the `MsgSeq` in the Message Id Component (see section 2.3.2) must be set to `LastMsg`.

Content:

- **TransRefBlk**
  This contains information which describes an OTP Message within an OTP Transaction (see section 2.3 immediately below)

- **TpoBlk,**
  - **TpoSelectionBlk,**
  - **AuthReqBlk,**
  - **AuthRespBlk,**
  - **OfferRespBlk,**
  - **PayReqBlk,**
  - **PayExchBlk,**
  - **PayRespBlk,**
  - **DelivReqBlk,**
  - **DelivRespBlk,**
  - **FailBlk**

These are the Trading Blocks.

The Trading Blocks present within an OTP Message, and the content of a Trading Block itself is dependent on the type of OTP Transaction being carried out - see the definition of each transaction in Section 5 Open Trading Protocol Transactions.

Full definitions of each Trading Block are described in section 4.
2.2.1 XML Document Prolog

The OTP Message is the root element of the XML document. It therefore needs to be preceded by an appropriate XML Document Prolog. For example:

```xml
<?XML Version='1.0'?>
<!DOCTYPE otp:OtpMessage >
<otp:OtpMessage Status='Original'>
   ...
</orp:OTPMessage>
```

2.3 Trans Ref Block

A Trans Ref Block contains information which describes an OTP Message within an OTP Transaction. The Trans Ref Block contains:

- a Transaction Id Component which globally uniquely identifies the OTP Transaction. The Transaction Id Components are the same across all OTP messages that comprise a single OTP transaction, and
- a Message Id Component which provides control information about the OTP Message as well as uniquely identifying the OTP Message within an OTP Transaction.

The definition of a Trans Ref Block is as follows:

```xml
<!ELEMENT TransRefBlk (TransId, MsgId) >
<!ATTLIST TransRefBlk
    BlkId ID #REQUIRED >
```

Attributes:

- **BlkId**: An identifier which uniquely identifies the Trans Ref Block within the OTP Transaction (see section 2.4 ID Attributes).

Content:

- **TransId**: See 2.3.1 Transaction Id Component immediately below.
- **MsgId**: See 2.3.2 Message Id Component immediately below.

2.3.1 Transaction Id Component

This contains information which globally uniquely identifies the OTP Transaction. Its definition is as follows:

```xml
<!ELEMENT TransId EMPTY>
<!ATTLIST TransId
    CompId ID #REQUIRED
    OtpTransId NM_TOKEN #REQUIRED >
```
Chapter 2 Protocol Structure

Attributes:

- **CompId**: An identifier which uniquely identifies the Transaction Id Component within the OTP Transaction.
- **OtpTransId**: Contains data which uniquely identifies the OTP Transaction. It must conform to the rules for Message Ids in [RFC 822].

### 2.3.2 Message Id Component

The Message Id Component provides control information about the OTP Message as well as uniquely identifying the OTP Message within an OTP Transaction. Its definition is as follows.

```xml
<!ELEMENT MsgId EMPTY >
<!ATTLIST MsgId
  OtpMsgId ID #REQUIRED
  Version NMTOKEN #FIXED '0.9'
  RespOtpMsg NMTOKEN #IMPLIED
  MsgSeq (FirstMsg|ContMsg|LastMsg) #REQUIRED
  TimeStamp CDATA #IMPLIED >
```

Attributes:

- **OtpMsgId**: An identifier which uniquely identifies the OTP Message within the OTP Transaction (see section 2.4 ID Attributes). Note that if the same OTP Message is resent (see the Status attribute on the OTP Message) then the OtpMsgId remains the same.
  
  Note that there is no CompId attribute for the Message Id Component since the OtpMsgId attribute serves the same purpose.

- **Version**: This identifies the version of OTP to which the OTP Message and its contents conforms. Note that in this "draft for public comment" version of Baseline OTP the version is always "0.9".

- **RespOtpMsg**: This contains the OTPMsgId of the OTP Message to which this OTP Message is a response. In this way all the OTP Messages in an OTP Transaction are unambiguously linked together. This field is required on every OTP Message except the first OTP Message in an OTP Transaction i.e. where MsgSeq is set to FirstMsg.

- **MsgSeq**: Indicates the position of the OTP Message within in the OTP Transaction. Its value may be:
  - FirstMsg which indicates it is the first message
  - ContMsg which indicates it is a message sent in response to an earlier message that has been received, or
  - LastMsg which indicates that this is the last message and no further message in reply is expected

- **TimeStamp**: Where the device sending the message has an internal clock, it is set to the time at which the OTP Message was created in [UTC] format.
2.4 ID Attributes

OTP Messages, Blocks (i.e. Trans Ref Blocks and Trading Blocks) and Trading Components (including the Trans Id Component and the Signature Component) are each given an XML "ID" attribute which is used to identify an instance of these XML elements. These identifiers are used so that one element can be referenced by another. The IDs are given the following names:

- **OTPMsgId** - the ID of an OTP Message
- **BlkId** - the ID of a Block
- **CompId** - the ID of a Component

The values of each **OTPMsgId**, **BlkId**, and **CompId** attribute are unique within an OTP transaction i.e. the set of OTP Messages which have the same globally unique Transaction ID Component. This means that it is possible to use these IDs to refer to and locate the content of any OTP Message, Block or Component from any other OTP Message, Block or Component in the same OTP Transaction using **Element References** (see section 2.5).

This section defines the rules for setting the values for the ID attributes of OTP Messages Blocks and Components.

2.4.1 OTP Message ID Definition

The **OTPMsgId** of an OTP Message must be unique within an OTP Transaction. It's definition is as follows:

```
OtpMsgId_value ::= OTPMsgIdPrefix OtpMsgIdSuffix
OtpMsgIdPrefix ::= NameChar (NameChar)*
OtpMsgIdSuffix ::= Digit (Digit)*
```

- **OTPMsgIdPrefix** - The same prefix is used for all messages sent by the Merchant or Consumer role as follows:
  - "M" - Merchant
  - "C" - Consumer
  - The prefix for the other roles in a trade is contained within the Organisation Component for the role and are typically set by the Merchant. The following is recommended as a guideline and must not be relied upon:
    - "V" - First (only) Value Acquirer
    - "W" - Second Value Acquirer
    - "D" - Deliverer
  - As a guideline, prefixes should be limited to one character.

- **OTPMsgIdSuffix** - The suffix consists of one or more digits. The suffix must be unique within a Trading Role within an OTP Transaction. The following is recommended as a guideline and must not be relied upon:
  - the first OTP Message sent by a trading role is given the suffix "1"
  - the second and subsequent OTP Messages sent by the same trading role are incremented by one for each message
  - no leading zeroes are included in the suffix
  - Put more simply the first OTP Message sent by a Consumer would have an
2.4.2 Block and Component ID Definitions

Block and Component IDs must also be unique within an OTP Transaction. Their definition is as follows:

```
BlkId_value ::= OTPMsgId "." IdSuffix
CompId_value ::= OTPMsgId "." IdSuffix
IdSuffix ::= Digit (Digit)*
```

- **OTPMsgId**: The OTPMsgId used is the ID of the OTP Message where the Block or Component is first used.
- **IdSuffix**: The suffix consists of one or more digits. The suffix must be unique within the OTPMsgId used to generate the BlkId or CompId. The following is recommended as a guideline and must not be relied upon:
  - the first Block or Component sent by a trading role is given the suffix "1"
  - the second and subsequent BlkIds or CompIds are incremented by one for each new Block or Component added to an OTP Message
  - no leading zeroes are included in the suffix

Put more simply, the first new Block or Component added to the second OTP Message sent, for example, by a consumer would have a BlkId or CompId of "C2.1", the second "C2.2", the third "C2.3" etc.

**Digit** has the same definition as the [XML] definition of Digit.
2.4.3 Example of use of ID Attributes

The diagram below illustrates how OtpMsgId, BlkId, and CompId attribute values are used.

2.5 Element References

A Trading Component or one of its child XML elements, may contain an XML attribute that refers to another Block (i.e. a Trans Ref Block or a Trading Block) or Trading Component (including a Transaction Id and Signature Component). These Element References are used for a many purposes, a few examples include:

- identifying an XML element whose hash value is included in a Signature Component,
- referring to the Value Acquirer Organisation Component which is used when making a Payment

An Element Reference always contains the value of an OtpMsgId, a BlkId, or a CompId.
Identifying the OTP Message, Trading Block or Trading Component which is referred to by an Element Reference, involves finding the XML element which:

- belongs to the same OTP Transaction (i.e. the Transaction Id Components of the OTP Messages match\(^7\), and
- where the value of the ID attribute of the element matches the value of the Element Reference.

An example of “matching” an Element Reference is illustrated in the example below.

**1st OTP Message**  
(e.g. from Merchant to Consumer)

**2nd OTP Message**  
(e.g. from Consumer to Value Acquirer)

---

**Note:**  
Element Reference attributes are defined as "NMTOKEN" rather than "IDREF" (see [XML]). This is because an IDREF requires that the XML element referred to is in the same XML Document. With OTP this is not necessarily the case.

---

\(^7\) The term “match” in this specification has the same definition as the [XML] definition of match
2.6 Signature Overview

This section provides an overview of how signatures are used within OTP Transactions. The detailed definitions of how signatures are calculated and encoded as XML elements are contained within sections 3.13 Signature Component and section 3.14 Certificate Component which cover:

- how OTP Signatures sign one or more hashes of Blocks or Components within any OTP Message in an OTP Transaction,

- rules designed to ensure that signatures are reproducible including:
  - identifying what to sign
  - a canonical encoding format, and
  - rules for calculating hashes and signatures
  - guidelines for checking hashes and signatures

Note: The method of calculating and encoding signatures described in this specification was developed since no approach to digitally signing XML documents was available at the time this specification was being written.

It is envisaged that the signing method described here may, in due course, be replaced by an XML standard for digitally signing XML documents, once one becomes established.

Suggestions and comments from other organisations working on digitally signing XML documents are particularly welcome.

Section 2.4 ID Attributes and section 2.5 Element References describe how Trading Blocks and Trading Components within an OTP message are identified and referenced.

8 OTP signatures support both asymmetric (public-key) signatures and symmetric (secret key) message authentication codes (MACs)
Chapter 2 Protocol Structure

Element References are used in generating signatures as illustrated in the diagram below.

Figure 10 Signature Overview

All signatures contained within OTP Messages must always sign:

- The Transaction Id Component (see section 2.3.1) in the OTP message. This binds the globally unique OtpTransId to other components which make up the OTP Transaction.
- the Trans Ref Block (see section 2.3). This binds the OtpTransId with information about the OTP Message contained inside the Message Id Component (see section 2.3.2).

Both are required because the OTP protocol structure requires verification of the Transaction Id Component but not the Trans Ref Block in some situations.

Note that the Components and Blocks being signed may be in any OTP Message in the OTP Transaction.

2.7 Extending OTP

Baseline OTP defines a minimum protocol which systems supporting OTP must be able to accept. As new versions of OTP are developed, additional types of OTP Transactions will be defined. In addition to this, Baseline and future versions of OTP will support user extensions to OTP through two mechanisms:

- extra XML elements, and
• new user-defined values for existing OTP codes.

2.7.1 Extra XML Elements

The XML element and attribute names used within OTP constitute an [XML Namespace\(^9\)]. This allows OTP to support the inclusion of additional XML elements within OTP messages through the use of [XML Namespaces].

Extra XML elements may be included at any level within an OTP message including:

• new Trading Blocks
• new Trading Components
• new XML elements within a Trading Component.

The following rules apply:

• any new XML element must be declared according to the rules for [XML Namespaces]. This means that:
  - the namespace must be declared to the XML parser
  - each element must have a start and end tags which conform to the rules for XML Namespaces

• new XML elements which are either Trading Blocks or Trading Components must contain the appropriate ID attributes. These are:
  - for Trading Blocks, BlkId, and
  - for Trading Components, CompId

In order to make sure that extra XML elements can be processed properly, OTP reserves the use of a special attribute, `otp:critical`, which takes the values `True` or `False` and may appear in extra elements added to an OTP message.

The purpose of this attribute is to allow an OTP aware application to determine if the OTP transaction can safely continue. Specifically:

• if an extra XML element has an "otp:critical" attribute with a value of "True" and an OTP aware application does not know how to process the element and its child elements, then the OTP transaction must fail (see Fail Trading Block section 4.11)

\(^9\) In drafts of the [XML] specification, the concept of "Namespaces" have been discussed. However they are not present in the XML documentation submitted for approval (see XML draft dated 8 December 1997) although it appears as if they may be included in version 1.1 of XML. It is considered by the authors of this document that OTP would be an ideal example of a Namespace so that other XML elements with potentially the same name can be included unambiguously in XML documents which conform to this specification. If Namespaces, or an equivalent, is not developed for XML as a whole then OTP is likely to propose its own equivalent. The Views of other organisations on this topic are sought.
Chapter 2 Protocol Structure

• if an extra XML element has an "otp:critical" attribute with a value of "False" then the OTP transaction may continue if the OTP aware application does not know how to process it. In this case:
  - any extra XML elements contained within an XML element defined within the OTP namespace, must be included with that element whenever the OTP XML element is used or copied by OTP
  - the content of the extra element must be ignored except that it must be included when it is hashed as part of the generation of a signature

• if an extra XML element has no "otp:critical" attribute then it must be treated as if it had an "otp:critical" attribute with a value of "True"

• if an XML element contains an "otp:critical" attribute, then the value of that attribute is assumed to apply to all the child elements within that element

In order to ensure that documents containing "otp:critical" are valid, it is declared as part of the DTD for the extra element as:

otp:critical   (True | False ) #IMPLIED

2.7.2 Opaque Embedded Data

There are five areas where OTP expects that opaque data will be embedded within OTP messages.

• the content of the Order Component (see section 3.4),
• the content of the Brand Element (see section 3.6.1),
• the content of the Pay Protocol Element (see section 3.6.2),
• the content of the 3.7 (see section 3.7) and
• the content of the Payment Scheme Component (see section 3.9).

The embedded data is called “opaque” because it is not processed by OTP, but is instead passed to or from order or payment processing software. For example, the Payment Scheme Component carries data that is generated or handled by software specific to a payment protocol.

2.7.2.1 Embedded XML data

The opaque embedded data may be in a variety of formats, including XML. The otp:critical attribute may be used to allow the same XML element name to be used in both OTP and within the content of the additional data.
For example, there may be a payment method which uses its own form of digital signatures and uses an XML element name which is also used by OTP. For example suppose that the element name \texttt{SignedData} was used by both OTP and by a payment method called \texttt{"xzpay"}. It would then be encoded as follows:

\begin{verbatim}
<otp:OtpMessage ... >
  ...
  <OtpSig ...>
    <SignedData ...>
      ...
    </SignedData>
  </OtpSig>
  ...
  <PaySchemeData Format='XML' ...>
    <xzpay:PayMsg1 otp:critical='True'...>
      ...
    </PayMsg1>
  </PaySchemeData>
  ...
</OtpMessage>
\end{verbatim}

\textit{OTP SignedData element, within the \texttt{"otp"} namespace.}

\textit{First message of payment protocol. The \texttt{"xzpay:"} prefix identifies this element and its children as part of the \texttt{"xzpay"} namespace.}

\textit{OTP SignedData element, within the \texttt{"xzpay"} namespace.}

\textbf{Figure 11 Extra XML Elements}

\textbf{Note:} The Order Component is the part of OTP that contains the definition of the goods or services being purchased. OTP is not prescriptive about the content and structure of the order. It is anticipated that the mechanism described here will be used by other organisations to support encapsulation of, for example, XML based EDI invoices within OTP messages. The author's of this specification seek suggestions on this topic.

The opaque embedded data may also be in a format not compliant with XML, such as binary or basic SGML without XML-style ending tags. To avoid confusion, such data should be bracketed with XML’s character data escape (CDATA) sequence. The non-XML opaque data should be prefixed with the character string "<![CDATA[", and suffixed with "]]>".

\subsection{2.7.2.2 Embedded non XML data}

Data embedded may not be XML, for example a SET payment protocol message of some type. In this case the embedded data should make use of the XML CDATA structures (see [XML] specifications) to ensure that the embedded data is not parsed by any XML parser as markup.

CDATA sections are terminated by the character sequence "]]>". This means that the embedded data must not contain this character sequence anywhere except the end of the CDATA section. However, there may be a legitimate reason why the sequence "]]>" should be included in the middle of a CDATA section.
2.7.3 User Defined Codes

User defined codes provide a simple way to identify additional values for the codes contained within this specification.

The definition of a user defined code is as follows:

user_defined_code ::= ( "x-" | "X-" ) domain_name "::" name

- **domain_name**: A name which identifies the organisation which is creating the user defined code (see [DNS]). The purpose of this field is to reduce the probability of two organisations creating the same user-defined name.
- **name**: A name specified by the organisation which owns the domain_name which identifies the user defined code within the domain_name.

User defined codes are identified in this specification as "x-ddd:nnn". The values of User Defined Codes must conform to the rules for the specific code (see explanations of the individual codes).

2.8 Identifying Languages

OTP uses [XML] Language Identification to specify which languages used within the content and attributes of OTP Messages.

The following principles have been used in order to determine which XML elements contain an XML:Lang attribute:

- a mandatory XML:Lang attribute is contained on every Trading Component which contains attributes or content which may need to be displayed or printed in a particular language.
- an optional XML:Lang attribute is included on child elements of these Trading Components. In this case the value of XML:Lang, if present, overrides the value for the Trading Component.

XML:Lang attributes which follow these principles are included in the Trading Components and their child XML elements defined in section 3.
Chapter 3 Trading Components

This section describes the Trading Components used within OTP. Trading Components are the child XML elements which occur immediately below a Trading Block as illustrated in the diagram below.

The Trading Components described in this section are listed below in approximately the sequence they are likely to be used:

- Protocol Options Component
- Authentication Component
- Authentication Response Component
- Order Component
- Organisation Component
- Brand List Component
- Brand Selection Component
- Pay Amount Component
- Payment Scheme Component
- Pay Receipt Component
- Delivery Component
Chapter 3 Trading Components

- Delivery Note Component
- Signature Component
- Certificate Component
- Fail Reason Component
- Fail OTP Message Component

Note that the following components are listed in other sections of this specification:
- Transaction Id Component (see section 2.3.1)
- Message Id Component (see section 2.3.2)

### 3.1 Protocol Options Component

Protocol options are options which apply to the OTP Transaction as a whole. Essentially it is used to identify what type of OTP Transaction is being carried out. For Baseline OTP it will identify one of the "Baseline" OTP Transactions (see section 5. Open Trading Protocol Transactions) by name.

The definition of a Protocol Options Component is as follows.

```xml
<!ELEMENT ProtocolOptions (TradeSequence?) >
<!ATTLIST ProtocolOptions
  CompId ID #REQUIRED
  XML:Lang NMTOKEN #REQUIRED
  OtpTransType (BaselineAuthentication |BaselineDeposit | BaselinePurchase | BaselineRefund | BaselineWithdrawal | BaselineValueExchange ) #REQUIRED
  ShortDesc CDATA #REQUIRED
  SuccessNetLocn CDATA #REQUIRED
  CancelNetLocn CDATA #REQUIRED
  ErrorNetLocn CDATA #REQUIRED >

<!ELEMENT TradeSequence (#PCDATA)>`
**OtpTransType**

This is the type of OTP Transaction being carried out. For Baseline OTP it identifies a "standard" OTP Transaction and implies the sequence and content of the OTP Messages exchanged between the Trading Roles. The valid values for Baseline OTP are:

- BaselineAuthentication
- BaselineDeposit
- BaselinePurchase
- BaselineRefund
- BaselineWithdrawal
- BaselineValueExchange

In later versions of OTP, this list will be extended to support different types of standard OTP Transaction based on market demand. It will also support the type **Dynamic** which indicates that the sequence is non-standard and defined by the **TradeSequence** element.

**ShortDesc**

This contains a short description of the OTP Transaction in the language defined by XML:Lang. Its purpose is to provide an explanation of what type of OTP Transaction is being conducted by the parties involved.

It is used to facilitate selecting an individual transaction from a list of similar transactions, for example from a database of OTP transactions which has been stored by a Consumer, Merchant, etc.

**SuccessNetLocn**

This contains the net location that should be displayed after the OTP Transaction has successfully completed. The content of this attribute is dependent on the Transport Mechanism (e.g. HTTP) that is used. See the Transport Mechanism supplement.

**CancelNetLocn**

This contains the net location that should be displayed by the Consumer after the OTP Transaction has been cancelled by one of the Trading Roles.

For this purpose, cancel is defined as sending or receiving a Fail Trading Block (see section 4.11) where the **FailType** attribute of all the **FailReasons** in the block are set to **Cancel**.

The content of this attribute is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.

**ErrorNetLocn**

This contains the net location that should be displayed after the OTP Transaction has failed due to an error.

For this purpose, an error is defined as sending or receiving a Fail Trading Block (see section 4.11) where the **FailType** attribute of at least one of the **FailReasons** in the block is set to **Error**, or there has been an irrecoverable loss of communication.

The content of this attribute is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.

**Content:**

**TradeSequence**

This element is not used in Baseline OTP. In Full OTP it will contain a definition of the sequence, content and dependency of the OTP Messages to be used by the OTP Transaction.
3.2 Authentication Data Component

This Trading Component contains data about how an Authentication within the OTP Transaction will occur. Its definition is as follows.

```xml
<!ELEMENT AuthData ANY>
<!ATTLIST AuthData
  CompId ID #REQUIRED
  AuthMethod NMTOKEN #REQUIRED
  ContentFormat NMTOKEN #REQUIRED >
```

Attributes:

- **CompId**: An identifier which uniquely identifies the Authentication Data Component within the OTP Transaction.
- **AuthMethod**: This identifies the content of the Authentication Data Component. Valid values are:
  - `sha1`: This indicates that the recipient of the Authentication Data Component should generate a hash. See 3.3 Authentication Response Component.
  - `pay:ppp`: A payment protocol specific authentication method. The *ppp* identifies a payment protocol associated with a payment exchange which is part of the OTP Transaction. In this case the content and format of the `AuthData` element is defined in the appropriate Payment Scheme supplement. For example if a payment method "xzpay" provided an authentication method, then this attribute would have the value "pay:xzpay".
  - `x-ddd:nnn`: a user defined authentication scheme type see section (2.7.3 User Defined Codes).
- **ContentFormat**: Identifies the structure of the content of the Authentication Data Component. Its values may be:
  - `Xml`: The data is structured using the Extensible Markup Language [XML]
  - `Mime`: The data consists of a [MIME] message
  - `Pcdata`: The data contains data conforming to the [XML] rules for PCDATA but contains none of the structural features of XML
  - `Base64`: The data consists of binary information encoded using [Base64]

Contents:

- **ANY**: This contains the challenge data in a format defined by `ContentFormat` that is to be responded to using the method indicated by `AuthMethod`.

3.3 Authentication Response Component

This Authentication Response Component contains the results of an authentication. Its definition is as follows.

```xml
<!ELEMENT AuthResp ANY >
<!ATTLIST AuthResp
  CompId ID #REQUIRED
  ContentFormat NMTOKEN #REQUIRED >
```
Chapter 3 Trading Components

Attributes:

CompId
An identifier which uniquely identifies the Authentication Response Component within the OTP Transaction.

ContentFormat
Identifies the structure of the content of the Authentication Response Component. Its values may be:

- **Xml** The data is structured using the Extensible Markup Language [XML]
- **Mime** The data consists of a [MIME] message
- **Pcdata** The data contains data conforming to the XML rules for PCDATA but contains none of the structural features of XML
- **Base64** The data consists of binary information encoded using [Base64]

Content:

**ANY**
This contains the response to the content of the Authentication Data Component see section 3.2 in the format defined by ContentFormat.

For a payment specific scheme, it may contain scheme-specific data. Refer to the scheme-specific supplemental documentation.

If the **AuthMethod** attribute of the Authentication Data Component is "sha1" then the hash is generated using the methods defined in section 3.13.6 Hashes and Signatures, with the following variations:

- the Authentication Data Component is converted into its canonical form (see Hashes and Signatures)
- the resulting Authentication Data Component is concatenated with the recipient's secret key (the Authentication Data Component is concatenated first)

The concatenated result is hashed and encoded, using [Base64], and placed in the **AuthResp** content.

3.4 Order Component

An Order Component contains information about an order. Its definition is as follows.

```xml
<!ELEMENT Order ANY >
<!ATTLIST Order
  CompId ID #REQUIRED
  XML:Lang NMTOKEN #REQUIRED
  OrderRef CDATA #REQUIRED
  ShortDesc CDATA #REQUIRED
  OkFrom CDATA #REQUIRED
  OkTo CDATA #REQUIRED
  ContentFormat NMTOKEN #IMPLIED >
```

Attributes:

CompId
An identifier which uniquely identifies the Order Component within the OTP Transaction.

XML:Lang
Defines the language used by attributes or child elements within this component, unless overridden by an XML:Lang attribute on a child element. See section 2.8 Identifying Languages.
3.5 Organisation Component

The Organisation Component provides information about an individual or an organisation. This can be used for a variety of purposes. For example:

- to describe the merchant who is selling the goods,
- to identify who made a purchase,
- to identify who will take delivery of goods,
- to provide a customer care contact,
- to describe who will be the value acquirer.

Its definition is as follows.

```
<!ELEMENT Org (OrgRole+, ContactInfo?, PersonName?, PostalAddress?)>
<!ATTLIST Org
    CompId ID #REQUIRED
    XML:Lang NMTOKEN #REQUIRED
```
Chapter 3 Trading Components

Organisation Component

OrgId CDATA #REQUIRED
OtpMsgPrefix NMTOKEN #REQUIRED
LegalName CDATA #IMPLIED
ShortDesc CDATA #IMPLIED
LogoNetLocn CDATA #IMPLIED

Attributes:

CompId
An identifier which uniquely identifies the Organisation Component within the OTP Transaction.

XML:Lang
Defines the language used by attributes or child elements within this component, unless overridden by an XML:Lang attribute on a child element. See section 2.8 Identifying Languages.

OrgId
This is a code which the creator of the Organisation Component may use to identify the organisation. It must be unique within an OTP Transaction. It may be, for example, a customer reference or a value acquirer reference. If it is used in this way, then it may remove the need to specify the optional elements as the code can be used to look up the information in a database.

OtpMsgPrefix
Contains the prefix which must be used for all OTP Messages sent by the Organisation in this OTP Transaction. The values to be used are defined in 2.3.2 Message Id Component.

LegalName
For organisations which are companies this is their legal name in the language defined by XML:Lang.

ShortDesc
A short description of the organisation in the language defined by XML:Lang. It is typically the name by which the organisation is commonly known. For example, if the trading name was "Blue Meadows Financial Services Inc.". Then its short name would likely be "Blue Meadows".

It is used to facilitate selecting an individual organisation from a list of organisations, for example from a database of organisations involved in OTP Transactions which has been stored by a consumer.

LogoNetLocn
The net location which can be used to download the logo for the organisation.

The content of this attribute is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.

Content:

OrgRole See 3.5.1 Organisation Role Element below.
ContactInfo See 3.5.2 Contact Information Element below.
PersonName See 3.5.3 Person Name below.
PostalAddress See 3.5.4 Postal Address below.

3.5.1 Organisation Role Element

This identifies the role of an individual or organisation in the OTP Transaction. Note, an organisation may have more than one role and several roles may be present in one organisation element. Its definition is as follows:

<!ELEMENT OrgRole EMPTY >
<!ATTLIST OrgRole
Role NMTOKEN #REQUIRED >
Attributes:
Role

The role of the organisation. Valid values are:

- **Consumer**. The person or organisation that is acting in the role of a consumer in the OTP Transaction.
- **Merchant**. The person or organisation that is acting in the role of merchant in the OTP Transaction.
- **ValueAcquirer**. The financial institution or other organisation which is a value acquirer for the OTP Transaction.
- **Deliever**. The person or organisation that is delivering the goods or services for the OTP Transaction.
- **DelivTo**. The person or organisation that is receiving the delivery of goods or services in the OTP Transaction.
- **CustCare**. The organisation and/or individual who will provide customer care for an OTP Transaction.
- **x-ddd:nnn** a user defined role (see section 2.7.3 User Defined Codes).

### 3.5.2 Contact Information Element

This contains information which can be used to contact an organisation or an individual. All attributes are optional however at least one item of contact information should be present. Its definition is as follows.

```xml
<!ELEMENT ContactInfo EMPTY >
<!ATTLIST ContactInfo
  XML:Lang NMToken #IMPLIED
  Tel CDATA #IMPLIED
  Fax CDATA #IMPLIED
  Email CDATA #IMPLIED
  Web CDATA #IMPLIED >
```

Attributes:
- **XML:Lang** Defines the language used by attributes within this element. See section 2.8 Identifying Languages.

### 3.5.3 Person Name Element

This contains the name of an individual person. All fields are optional however as a minimum either the **GivenName** or the **FamilyName** should be present. Its definition is as follows.

```xml
<!ELEMENT PersonName EMPTY >
<!ATTLIST PersonName
  XML:Lang NMToken #IMPLIED
  Title CDATA #IMPLIED
  GivenName CDATA #IMPLIED
  Initials CDATA #IMPLIED
  FamilyName CDATA #IMPLIED >
```

Attributes:
- **XML:Lang** Defines the language used by attributes within this element. See section 2.8 Identifying Languages.
3.5.4 Postal Address Element

This contains an address which can be used, for example, for the physical delivery of goods, services or letters. Its definition is as follows.

```xml
<!ELEMENT PostalAddress EMPTY >
<!ATTLIST PostalAddress
  XML:Lang NMTOKEN #IMPLIED
  AddressLine1 CDATA #IMPLIED
  AddressLine2 CDATA #IMPLIED
  CityOrTown CDATA #IMPLIED
  StateOrRegion CDATA #IMPLIED
  PostalCode CDATA #IMPLIED
  Country CDATA #IMPLIED >
```

Attributes:

- **XML:Lang**: Defines the language used by attributes within this element. See section 2.8 Identifying Languages.

3.6 Brand List Component

Brand List Components are contained within the Trading Protocol Options Block (see section 4.1) of the OTP Transaction. They contain lists of payment brands and payment protocols from which a particular brand and protocol may be selected. Its definition is as follows.

```xml
<!ELEMENT BrandList (Brand+, PayProtocol+) >
<!ATTLIST BrandList
  CompId ID #REQUIRED
  XML:lang NMTOKEN #REQUIRED
  ShortDesc CDATA #REQUIRED
  PayDirection (Debit | Credit) #REQUIRED >
```

Attributes:

- **CompId**: An identifier which uniquely identifies the Brand List Component within the OTP Transaction.
- **XML:Lang**: Defines the language used by attributes or child elements within this component, unless overridden by an XML:Lang attribute on a child element. See section 2.8 Identifying Languages.
- **ShortDesc**: A text description in the language defined by XML:Lang giving details about the purpose of the Brand List. This information must be displayed to the receiver of the Brand List in order to assist with making the selection.
- **PayDirection**: Indicates the direction in which the payment for which a Brand is being selected is to be made. Its values may be:
  - **Debit**: The sender of the Payment Request Block (e.g. the Consumer) to which this Brand List relates will make the payment to the Value Acquirer, or
  - **Credit**: The sender of the Payment Request Block to which this Brand List relates will receive a payment from the Value Acquirer.
Chapter 3 Trading Components

Content:

Brand

This describes a Brand. The sequence of the Brand elements (see section 3.6.1) within the Brand List does not indicate any preference. It is recommended that software which processes this Brand List presents Brands in a sequence which the receiver of the Brand List prefers.

PayProtocol

This contains information about a Payment Protocol (see section 0) which may be used with a particular Brand.

Example:

```xml
<BrandList CompId='M1.3' XML:Lang='en-us' ShortDesc='Select payment method for goods ordered' PayDirection='Debit'>
  <Brand Id='1' Name='MC' ShortDesc='MasterCard Credit' Logo='...' ProtocolChoices='SET' ContentFormat='Base64'>1298erh18dh238djwoire</Brand>
  <Brand Id='2' Name='Visa' ShortDesc='Visa Credit' Logo='...' ProtocolChoices='SET SCCD' ContentFormat='Base64'>238djqw1298erh18dhoire</Brand>
  <PayProtocol Id='SET' Name='SET' Version='1.0' PayReqNetLocn='http://www.merchant.com/set' SecPayReqNetLocn='https://www.merchant.com/set' ContentFormat='Base64'>8ueu26e482hd82he82</PayProtocol>
  <PayProtocol Id='SCCD' Name='SCCD' Name='Secure Channel Credit Debit' Version='1.0' SecPayReqNetLocn='https://www.merchant.com/sccd' ContentFormat='Base64'>482hd82h8ueu26ee82</PayProtocol>
</BrandList>
```

3.6.1 Brand Element

A Brand Element describes a brand that can be used for making a payment. One or more of these elements is carried in each Brand List Component that has the PayDirection attribute set to Debit. Exactly one Brand Element may be carried in a Brand List Component that has the PayDirection attribute set to Credit.

```xml
<!ELEMENT Brand ANY >
<!ATTLIST Brand
  Id Id #REQUIRED
  XML:Lang NMTOKEN #IMPLIED
  Name CDATA #REQUIRED
  ShortDesc CDATA #REQUIRED
  Logo CDATA #REQUIRED
  ProtocolChoices IDREFS #REQUIRED
  ContentFormat NMTOKEN #IMPLIED >
```
Attributes:

Id
Element identifier, potentially referenced in a Brand Selection Component contained in a later Payment Request message and uniquely identifies the element within the Brand List.

XML:Lang
Defines the language used by attributes or child elements within this component, unless overridden by an XML:Lang attribute on a child element. See section 2.8 Identifying Languages.

Name
A payment brand name, in the language defined by XML:Lang such as “MasterCard”, “Visa”, “JCB”, “Mondex”, “GeldKarte”. These will generally be trademarked names declared by the trademark owners.

ShortDesc
Text description in the language defined by XML:Lang giving more details about the brand. The description is prescribed by the organisation that owns the brand.

LogoNetLocn
The Net Location specifying where a brand logo can be obtained.

The content of this attribute is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.

ProtocolChoices
Identifies the protocols supported for this brand. Specified as a space-separated list of ID’s of Protocol elements contained within the Brand List.

The sequence of the elements in the list indicates the preference of the sender of the Brand List. The software which is processing the Brand List should use the first protocol in protocol choices which the recipient of the Brand List can support.

ContentFormat
This attribute is required if the content of the Brand element is non-empty. It defines the format of the content of the Brand element. Valid values are

- Xml The data is structured using the Extensible Markup Language [XML]
- Mime The data consists of a [MIME] message
- PCDATA The data contains data conforming to the [XML] rules for PCDATA but contains none of the structural features of XML
- Base64 The data consists of binary information encoded using [Base64]
- x-ddd:nnn The structure of the data is user defined (see section 2.7.3 User Defined Codes).

Content:
ANY

Information about the brand which may be used by the payment protocol. The format of this data is defined by the ContentFormat attribute and the Language by the XML:Lang attribute. The content of this information is defined in the supplement for a payment protocol which describes how the payment protocol works with OTP.

Example:

```xml
<Brand Id='2' Name='Visa' ShortDesc='Visa Credit' Logo='...' ProtocolChoices='SET1 SET2 SCCD' ContentFormat='Base64'>
  238djqwl298erhl8dhoire
</Brand>
```
### 3.6.2 Pay Protocol Element

A Pay Protocol element specifies details of a protocol that can be used for a payment. One or more of these elements is carried in each Brand List, regardless of whether the payment is a credit or a debit.

```xml
<!ELEMENT PayProtocol ANY >
<!ATTLIST PayProtocol
  Id Id #REQUIRED
  Name CDATA #REQUIRED
  Version NMTOKEN #REQUIRED
  PayReqNetLocn CDATA #IMPLIED
  SecPayReqNetLocn CDATA #IMPLIED
  ContentFormat NMTOKEN #IMPLIED >
```

**Attributes:**

- **Id**: Element identifier, potentially referenced in a Brand element; or in a Brand Selection Component contained in a later Payment Request message which uniquely identifies the Pay Protocol element within the Brand List.

- **Name**: A well-known protocol name, such as “SET”. Protocol names are declared by their designers, who are expected to pick unique names.

- **Version**: Indicates the version of the protocol which will be used. Protocol versions are declared by the designer of the protocol. Typical values are "1.0".

- **PayReqNetLocn**: The Net Location indicating where an unsecured Payment Request message should be sent if this protocol choice is used.

  The content of this attribute is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.

- **SecPayReqNetLocn**: The Net Location indicating where a secured Payment Request message should be sent if this protocol choice is used.

  The content of this attribute and the type of secure channel used is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.

- **ContentFormat**: This attribute is required if the content of the Pay Protocol Element is non-empty. It defines the format of the content of the Pay Protocol element. Valid values are
  - `Xml`: The data is structured using the Extensible Markup Language [XML]
  - `Mime`: The data consists of a [MIME] message
  - `Pcdata`: The data contains data conforming to the [XML] rules for PCDATA but contains none of the structural features of XML
  - `Base64`: The data consists of binary information encoded using [Base64]
  - `x-ddd:nnn`: The structure of the data is user defined (see section 2.7.3 User Defined Codes).

**Content:**

- **ANY**: Information about the protocol which is used by the payment protocol in the format defined by ContentFormat. The content of this information is defined in the supplement for a payment protocol which describes how the payment protocol works with OTP. An example of its use could be to include a payment protocol message.
The following rules apply to net locations:

- either `PayReqNetLocn` or `SecPayReqNetLocn` or both must be present
- if only one of the two Net Locations is present, then the one present must be used
- if both are present, then the sender of the message may use either depending on preference

Example:

```xml
<PayProtocol Id='SET1' Name='SET' Version='1.0'
  PayReqNetLocn='http://www.merchant.com/etill/set1' >
  8ueu26e482hd82he82
</PayProtocol>

<PayProtocol Id='SET2' Name='SET' Version='2.0'
  PayReqNetLocn='http://www.merchant.com/etill/set2' >
  26e482hd82he828ueu
</PayProtocol>

<PayProtocol Id='SCCD' Name='SCCD' Version='1.0'
  PayReqNetLocn='http://www.merchant.com/etill/sccd1' >
  82hd82he8226e48ueu
</PayProtocol>
```

### 3.7 Brand Selection Component

A Brand Selection Component identifies the choice of payment brand and protocol. This element is used:

- in Payment Request messages within Purchase and Value Exchange trades
- to, optionally, inform a merchant in a purchase of the payment brand being used, and

In Baseline OTP, the integrity of Brand Selection Components is not guaranteed. However, modification of Brand Selection Components can only cause denial of service if the payment protocol itself is secure against message modification, duplication, and swapping attacks.

```xml
<!ELEMENT BrandSelection ANY >
<!ATTLIST BrandSelection
  CompId Id #REQUIRED
  BrandListRef NMTOKEN #REQUIRED
  BrandRef NMTOKEN #REQUIRED
  PayProtocolRef NMTOKEN #REQUIRED
  ContentFormat NMTOKEN#IMPLIED>
```
Attributes:

CompId
An identifier which uniquely identifies the Brand Selection Component within the OTP Transaction.

BrandListRef
The ID of the Brand List Component from which a Brand is being selected

BrandRef
The ID of a Brand element within the Brand List Component that is being selected that is to be used in the payment.

PayProtocolRef
The ID of a Pay Protocol element within the Brand List Component which is to be used when making the payment.

ContentFormat
This attribute is required if the content of the Brand Selection element is non-empty. It defines the format of the content of the Brand Selection element. Valid values are
- Xml The data is structured using the Extensible Markup Language [XML]
- Mime The data consists of a [MIME] message
- Pcdata The data contains data conforming to the [XML] rules for PCDATA but contains none of the structural features of XML
- Base64 The data consists of binary information encoded using [Base64]
- x-ddd:nnn The structure of the data is user defined (see section 2.7.3 User Defined Codes).

Content:
ANY
This contains any additional data that may be required by a particular payment brand or protocol in the format defined by ContentFormat. See the payment method supplement for OTP for rules on how this is used.

The following rules apply:

- The BrandListRef must contain the ID of a Brand List Component in the same OTP Transaction
- Every Brand List Component in the Trading Protocol Options Block must be referenced by one and only one Brand Selection Component
- The BrandRef must refer to the ID of a Brand contained within the Brand List Component referred to by BrandListRef
- The PayProtocolRef must refer to one of the Pay Protocol Element IDs mentioned in ProtocolChoices

Example:

```xml
<BrandSelection CompId='c1.2'
  TransRef='19971103230201-435@merchant.com'
  BrandListRef='m1.1' BrandRef='MC' PayProtocolRef='SET1' />
</BrandSelection>
```
3.8 Pay Amount Component

A Pay Amount Component contains information used to control how a payment is carried out. It provides information on, for example:

- how much value the value acquirer is to be paid or pay out,
- whether or not a payment receipt will be provided.

Its definition is as follows.

```xml
<!ELEMENT PayAmount (StartAfter*) >
<!ATTLIST PayAmount
  CompId ID #REQUIRED
  Amount CDATA #REQUIRED
  CurrCodeType NMTOKEN 'ISO4217'
  CurrCode CDATA #REQUIRED
  PayDirection (Debit|Credit) #REQUIRED
  OkFrom CDATA #REQUIRED
  OkTo CDATA #REQUIRED
  SignedPayReceipt (True|False) #REQUIRED
  BrandListRef NMTOKEN #REQUIRED
  VaOrgRef NMTOKEN #REQUIRED
  AuthDataRef NMTOKEN #IMPLIED
  VaPaySigRef NMTOKEN #IMPLIED >
<!ELEMENT StartAfter (#PCDATA)>```

Attributes:

- **CompId**: An identifier which uniquely identifies the Pay Amount Component within the OTP Transaction.

- **Amount**: Indicates the amount to be paid in whole and fractional units of the currency. For example $245.35 would be expressed "245.35". Note that values smaller than the smallest denomination are allowed. For example one tenth of a cent would be "0.001".

- **CurrCodeType**: Indicates the domain of the CurrCode. This field is included so that the currency code may support non-standard "currencies" such as frequent flyer points, trading stamps, etc. Its values may be:
  - ISO4217 indicates the currency code conforms to [ISO 4217]
  - x-ddd:nnn a user defined currency code type (see section 2.7.3 User Defined Codes).

- **CurrCode**: A code which identifies the currency to be used in the payment. The domain of valid currency codes is defined by CurrCodeType

- **PayDirection**: Indicates the direction in which the payment is to be made. Its values may be:
  - Debit: The sender of the Payment Request Block in which this Pay Amount Component is contained will make the payment to the Value Acquirer, or
  - Credit: The sender of the Payment Request Block in which this Pay Amount Component is contained will receive a payment from the Value Acquirer.

- **OkFrom**: The date and time in [UTC] format after which a Value Acquirer may accept for processing a Payment Request Block (see section 4.6) containing the Pay Amount Component.
Chapter 3 Trading Components

OkTo
The date and time in [UTC] format before which a Value Acquirer may for processing accept a Payment Request Block containing the Pay Amount Component.

SignedPayReceipt
This indicates whether or not the Payment Response Block will contain a signature to bind the payment receipt to the reason (e.g. an offer) why the payment was being made. Valid values are:

• True a signature will be included, or
• False a signature will not be included.

BrandListRef
An Element Reference (see section 2.5) of a Brand List Component (see section 3.6) within the TPO Trading Block for the OTP Transaction.

VaOrgRef
An element reference (see section 2.4) of an Organisation Component (see section 3.5) for the Value Acquirer for this payment

AuthDataRef
An element reference (see section 2.4) of an Authentication Data Component (see section) which is to be used for authentication of the Trading Role which sends the Payment Request Block containing the Pay Amount Component to the Value Acquirer. If not present, then no authentication is to take place.

VaPaySigRef
If present, then this field contains the element reference (see section 2.4) of the Signature which should be checked by the Value Acquirer to make sure that the payment request is valid.

If missing, then it indicates that the Value Acquirer does not make any checks on signatures.

Content:

StartAfter
Contains Element References (see section 2.5) of other Pay Amount Components which describe payments which must be complete before this payment can start. If no StartAfter is present then there are no dependencies and the payment can start immediately

3.9 Payment Scheme Component

A Payment Scheme Component contains payment protocol information for a specific payment scheme which is transferred between the parties involved in a payment for example a [SET] message. Its definition is as follows.

<!ELEMENT PaySchemeData ANY >
<!ATTLIST PaySchemeData
  CompId ID #REQUIRED
  ContentFormat NMTOKEN #REQUIRED >

Attributes:

CompId
An identifier which uniquely identifies the Payment Scheme Component within the OTP Transaction.

ContentFormat
Identifies the structure of the content of the Payment Scheme Component. Its values may be:

• Xml The data is structured using the Extensible Markup Language [XML]
• Mime The data consists of a [MIME] message
• Pcdata The data contains data conforming to the XML rules for PCDATA
but contains none of the structural features of [XML]
- **Base64** The data consists of binary information encoded using [Base64]

### 3.10 Pay Receipt Component

A Pay Receipt Component contains payment scheme specific data which can be used after the payment has completed to verify the payment occurred. Its definition is as follows.

```xml
<!ELEMENT PayReceipt ANY >
<!ATTLIST PayReceipt
  CompId ID #REQUIRED
  PayRef NMTOKEN #REQUIRED
  ContentFormat NMTOKEN #REQUIRED >
```

**Attributes:**

- **CompId**
  An identifier which uniquely identifies the Pay Receipt Component within the OTP Transaction.

- **PayRef**
  Contains an Element Reference (see section 2.5) to the Pay Amount Component (see section 3.8) to which this payment receipt applies.

- **ContentFormat**
  Identifies the structure of the content of the Pay Receipt Component. Its values may be:
  - **Xml** The data is structured using the Extensible Markup Language [XML]
  - **Mime** The data consists of a [MIME] message
  - **Pcdata** The data contains data conforming to the XML rules for PCDATA but contains none of the structural features of [XML]
  - **Base64** The data consists of binary information encoded using [Base64]

**Content:**

- **ANY**
  Contains the payment scheme specific record of the payment which can be used for receipt purposes. The format of the data is defined by the `ContentFormat` attribute. Each payment scheme defines its supplement the structure of the content.

### 3.11 Delivery Component

The Delivery Element contains information required to deliver goods or services. Its definition is as follows.

```xml
<!ELEMENT Delivery (DelivData?) >
<!ATTLIST Delivery
  CompId ID #REQUIRED
  XML:Lang NMTOKEN #REQUIRED
```

**Content:**

- **ANY**
  Contains the payment scheme specific record of the payment which can be used for receipt purposes. The format of the data is defined by the `ContentFormat` attribute. Each payment scheme defines in its supplement the structure of the content.
Chapter 3 Trading Components

NoDelivExch (True|False) #REQUIRED
DelivAndPayResp CDATA #REQUIRED >

Attributes:
CompId
An identifier which uniquely identifies the Delivery Component within the OTP Transaction.

XML:Lang
Defines the language used by attributes or child elements within this component, unless overridden by an XML:Lang attribute on a child element. See section 2.8 Identifying Languages.

NoDelivExch
Indicates if this OTP Transaction includes the messages associated with a Delivery Exchange. Valid values are:
- True indicates it does include a Delivery Exchange
- False indicates it does not include a Delivery Exchange
If set to true then a DelivData element must be present. If set to false it may be absent.

DelivAndPayResp
Indicates if the Delivery Response Block (see section 4.10) and the Payment Response Block (see section 4.8) are combined into one OTP Message. Valid values are:
- True indicates both blocks will be in the same OTP Message, and
- False indicates each block will be in a different OTP Message
DelivAndPayResp should not be true if NoDelivExch is True.

In practice combining the Delivery Response Block and Payment Response Block is only practical if the Merchant, the Value Acquirer and the Deliverer are the same organisation since:
- the Value Acquirer must have access to Order Component information so that they know what to deliver, and
- the Value Acquirer must be able to carry out the delivery

Content:
DelivData
Contains details about how the delivery will be carried out. See 3.11.1 DelivData Element below.

3.11.1 DelivData Element

The DelivData element contains information about where and how goods are to be delivered. Its definition is as follows.

<!ELEMENT DelivData ANY >
<!ATTLIST DelivData
XML:Lang NMTOKEN #IMPLIED
OkFrom CDATA #REQUIRED
OkTo CDATA #REQUIRED
DelivMethod NMTOKEN #REQUIRED
DelivToRef NMTOKEN #REQUIRED
DelivReqNetLocn CDATA#REQUIRED
SecDelivReqNetLocn CDATA #REQUIRED
ContentFormat NMTOKEN#REQUIRED >
Chapter 3 Trading Components

Attributes:

XML:Lang
Defines the language used by attributes or child elements within this component, unless overridden by an XML:Lang attribute on a child element. See section 2.8 Identifying Languages.

OkFrom
The date and time in [UTC] format after which the Deliverer may accept for processing a Delivery Request Block (see section 4.9).

OkTo
The date and time in [UTC] format before which the Deliverer may accept for processing a Delivery Request Block.

DelivMethod
Indicates the method by which goods or services may be delivered. Valid values are:
- Post the goods will be delivered by post or courier
- Web the goods will be delivered electronically as a response to the Delivery Request Block
- Email the goods will be delivered electronically by e-mail
- x-ddd:nnn a user defined delivery method see 2.7.3 User Defined Codes.

DelivToRef
The Element Reference (see section 2.4) of an Organisation Component within the OTP Transaction which has a role of DelivTo. The information in this block is used to determine where delivery is to be made. It must be compatible with DelivMethod. Specifically if the DelivMethod is:
- Post, then there must be a Postal Address Element containing sufficient information for a postal delivery,
- Web, then there are no specific requirements. The information will be sent in a web page back to the Consumer
- Email, then there must be Contact Information Element with a valid e-mail address

DelivReqNetLocn
This contains the Net Location to which an unsecured Delivery Request Block (see section 4.9) which contains the Delivery Component should be sent.

The content of this attribute is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.

SecDelivReqNetLocn
This contains the Net Location to which a secured Delivery Request Block (see section 4.9) which contains the Delivery Component should be sent.

The content of this attribute and the type of secure channel used is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.

ContentFormat
Defines the format of the content of the DelivData element. Valid values are:
- Xml The data is structured using the Extensible Markup Language [XML]
- Pcdata The data contains data conforming to the XML rules for PCDATA but contains none of the structural features of [XML]
- Mime The data consists of a [MIME] message
- x-ddd:nnn The structure of the data is user defined (see section 2.7.3 User Defined Codes).

Content:
ANY Narrative text in the format defined by ContentFormat, and in the language defined by XML:Lang, which provides additional information to the deliverer which the merchant wants to provide
The following rules apply to net locations:

- either DelivReqNetLocn or SecDelivReqNetLocn or both must be present
- if only one of the two Net Locations is present, then the one present must be used
- if both are present, then the sender of the message may use either depending on preference

### 3.12 Delivery Note Component

A Delivery Note contains delivery instructions about the delivery of goods or services or potentially the actual Delivery Information itself. It is information which the person or organisation receiving the Delivery Note can use when delivery occurs.

```xml
<!ELEMENT DeliveryNote ANY >
<!ATTLIST DeliveryNote
  CompId ID #REQUIRED
  XML:Lang NMTOKEN #REQUIRED
  ContentFormat NMTOKEN#REQUIRED >
```

**Attributes:**

- **CompId**: An identifier which uniquely identifies the Delivery Note Component within the OTP Transaction.
- **XML:Lang**: Defines the language used by attributes or child elements within this component, unless overridden by an `XML:Lang` attribute on a child element. See section 2.8 Identifying Languages.
- **ContentFormat**: Identifies the structure of the content of the Delivery Note Component. Its values may be:
  - *Xml*: The data is structured using the Extensible Markup Language [XML]
  - *Mime*: The data consists of a [MIME] message
  - *NetLocation*: The data consists of a Net Location, for example a URL. The exact content is dependent on the Transport Mechanism being used. See the Transport Mechanism supplement
  - *Pcdata*: The data contains data conforming to the [XML] rules for PCDATA but contains none of the structural features of XML
  - *Base64*: The data consists of binary information encoded using [Base64]
  - *x-ddd:nnn*: The structure of the data is user defined (see section 2.7.3 User Defined Codes).

**Content:**

- **DeliveryNote**: Contains the actual delivery note information its structure is identified by `ContentFormat` and the language by `XML:Lang`.

If the content of the Delivery Message is a Mime message\(^\text{10}\) then the Delivery Note may trigger an application which causes the actual delivery to occur.

---
\(^{10}\) This may also apply if the content is XML
### 3.13 Signature Component

Each Signature Component digitally signs one or more Blocks (i.e. Trans Ref Blocks or Trading Blocks) or Components including other Signature Components.

The structure of a Signature Component is illustrated in the diagram below.

The Signature Component:

- hashes one or more Blocks or Components in one or more OTP Messages within the same OTP Transaction
- concatenates these hashes and any additional information to be signed in the form of authenticated attributes into a SignedData element, and
- signs the SignedData element using the optional certificate identified in the CertRef attribute of the DigSig element.

Note that a SignedData Element may be signed by more than one DigSig element.

The definition of a Signature Component is as follows.

```xml
<!ELEMENT OtpSig (SignedData, DigSig+, UnAuthAttr*)>
<!ATTLIST OtpSig
  CompId ID #REQUIRED
  XML:Lang NM_TOKEN #REQUIRED>
```
Attributes:

CompId
An identifier which uniquely identifies the Signature Component within the OTP Transaction.

XML:Lang
Defines the language used by attributes or child elements within this component, unless overridden by an XML:Lang attribute on a child element. See section 2.8 Identifying Languages.

Content:

SignedData
An initial non-empty list of element hashes containing an Element Reference (see section 2.5) and the hash of the corresponding element. For its definition see section 3.13.1 SignedData Element

DigSig
Contains the digital signature calculated over the data in SignedData. For its definition see section 3.13.4 Digital Signature Element.

UnAuthAttr
Contains additional data to be associated with this signature. For its definition see section 3.13.5 Unauthenticated Attribute Element.

3.13.1 SignedData Element

This contains the data which is to be signed it consists of:

• one or more hashes of Blocks or Components, and
• any authenticated attributes to be associated with the signature

The definition of a SignedData element is as follows.

<!ELEMENT SignedData(Hash+, AuthAttr*)>

Content:

Hash
A hash of a Block or a Component which is to be signed. For its definition see 3.13.2 Hash Element.

AuthAttr
Additional material it is desired to secure with this signature. For example the date of signing for this particular signature. For its definition see section 3.13.3.

3.13.2 Hash Element

The Hash Element identifies the Block or Component to be hashed, the type of Hash to be created and contains in its content, the actual hash value.

The definition of a Hash Element is as follows.

<!ELEMENT Hash (#PCDATA)>
<!ATTLIST Hash
 HashType NMTOKEN 'SHA1'
 ElRef NMTOKEN #REQUIRED>
Chapter 3 Trading Components

3.13.3 Authenticated Attribute Element

The AuthAttr element contains additional material it is desired to secure with this signature. For example the date of signing for this particular signature.

The definition of an AuthAttr element is as follows.

```
<!ELEMENT AuthAttr ANY>
<!ATTLIST AuthAttr
  XML:Lang NMTOKEN #IMPLIED
  ContentFormat NMTOKEN #REQUIRED >
```

Attributes:

- **XML:Lang**
  - Defines the language used by the content of the attribute. See section 2.8 Identifying Languages.

- **ContentFormat**
  - Defines the format of the content of the Authenticated Attribute element. Valid values are
    - **Xml** The data is structured using the Extensible Markup Language [XML]
    - **Mime** The data consists of a [MIME] message
    - **Pcdata** The data contains data conforming to the [XML] rules for PCDATA but contains none of the structural features of XML
    - **Base64** The data consists of binary information encoded using [Base64]
    - **x-ddd:nnn** The structure of the data is user defined (see section 2.7.3 User Defined Codes).

Content:

- **ANY**
  - Contains any information required in the format defined by ContentFormat.
Chapter 3 Trading Components

3.13.4 Digital Signature Element

This contains the actual digital signature resulting from signing the information contained within SignedData.

The definition of a Digital Signature element is as follows.

```xml
<!ELEMENT DigSig #PCDATA>
<!ATTLIST   DigSig
    HashType       NMTOKEN 'SHA1'
    SigAlg         NMTOKEN 'RSA'
    SignerKeyid    NMTOKEN #IMPLIED
    ElRef          NMTOKEN #IMPLIED
    XREF           CDATA   #IMPLIED >
```

Attributes:

- **HashType**
  - The hash algorithm used to calculate the hash in the content of the element. For valid values see the HashType attribute of the Hash Element (see section 3.13.2).

- **SigAlg**
  - The algorithm used to calculate digital signature from the hash. Valid values are:
    - RSA signature uses the [RSA] algorithm
    - DSA signature uses the [DSA] algorithm
    - ECCDSA signature uses the [ECCDSA] algorithm
    - HMAC signature uses the [HMAC] algorithm
    - x-ddd:nnn a user defined signature algorithm see section (2.7.3 User Defined Codes)

    When SignerKeyID is used, the SigAlg will typically be based on HMAC or some other message authentication code (MAC) using a symmetric key that is exchanged outside of OTP. A public-key signature could also be used, without publishing the public key value.

    When ElRef or XRef is used, the SigAlg should be a public-key algorithm.

- **SignerKeyID**
  - Contains an identifier indicating the key that should be used to verify the signature.

    The definitions of this identifier and of the referenced key are private between the signer and whoever verifies the signature. A SignerKeyID is used when only specific message receivers need to verify the signature. A certificate identified by ElRef or XRef should be used when the public in general needs to verify the signature.

- **ElRef**
  - The Element Reference (see section 2.5) of a Certificate Component that contains optional certificates, certificate revocation lists, or similar key authenticating items. An ElRef is used when the signature is a public key signature that may be verified by anyone.

- **XRef**
  - This contains an external reference in the form of a Net Location to the location where the certificate may be located. An XRef is used when the signature is a public key signature that may be verified by anyone.

    The format and content of the Net Location is dependent on the Transport Mechanism being used. See the appropriate Transport Mechanism supplement.
Contents:

PCDATA

Contains the results [Base64] encoded of hashing the SignedData element using the algorithm specified in the HashType attribute of the Signature element followed by the processing of the resultant hash using the algorithm specified in the SigAlg attribute of Digital signature.

Rules:

- At least one of SignerKeyID, ElRef, or XRef must be present.
- If SignerKeyID is present, then ElRef and XRef should not be included.
- If only ElRef is present then it must point to a Certificate Component within the same OTP Transaction.
- If only XRef is present then the certificate to be used must be retrieved.

If both ElRef and XRef are present, then the Certificate Component identified by ElRef should be used.

3.13.5 Unauthenticated Attribute Element

This contains additional information to be associated with the signature.

The definition of a UnAuthAttr element is as follows.

```xml
<!ELEMENT UnAuthAttr ANY>
<!ATTLIST UnAuthAttr
  XML:Lang NMTOKEN #IMPLIED
  ContentFormat NMTOKEN #REQUIRED
```

Attributes:

- **XML:Lang**
  Defines the language used by the content of the attribute. See section 2.8 Identifying Languages.

- **ContentFormat**
  Defines the format of the content of the UnAuthenticated Attribute element. Valid values are
  - Xml The data is structured using the Extensible Markup Language [XML]
  - Mime The data consists of a [MIME] message
  - PCDATA The data contains data conforming to the [XML] rules for PCDATA but contains none of the structural features of XML
  - Base64 The data consists of binary information encoded using [Base64]
  - x-ddd:nnn The structure of the data is a user defined (see section 2.7.3 User Defined Codes).

Content:

ANY

Additional material to be associated with this signature in the format defined by ContentFormat.
3.13.6 Hashes and Signatures

In outline signing of data consists of:

- calculating hashes of one or more Blocks or Components in one or more OTP Messages in the same OTP Transaction and including those hashes with other data to be signed in a SignedData element (see section 3.13.1)
- hashing and then encrypting (signing) the SignedData element and placing the result in the content of a DigSig element (see section 3.13.4).

Both of these involve calculations over data which must be in a canonical form if the results of the signing calculations are to be consistent and reproducible.

This section describes:

- the scope of what to hash or sign, within an element
- the canonical encoding format of an element to be used
- calculating hashes and signatures from the canonical encoding format
- guidance on "signature hygiene", and
- guidance for checking signatures

3.13.6.1 What to sign

The data signed is always an "entire" XML element. This means:

- for non EMPTY elements, starting with, and including, the leftmost "<" of the start tag and finishing with, and including, the rightmost ">" of the end tag of the element.
- for EMPTY elements, starting with, and including the leftmost "<" of the element and finishing with, and including, the rightmost ">" of the element.

For example:

```
First character to Include
<Order ...>

...<Order>

Last character to Include
```

* Figure 14 What to Sign*
3.13.6.2 Canonical Encoding Format

The canonical format of an XML element is created by firstly deriving the logical content and structure of the underlying XML document by parsing it, and then generating the canonical physical form of the element based on the logical structure. This is illustrated in the diagram below.

For the XML element being generated or any of its child elements:

- convert all characters in the element to [UTF16] format\(^1\).
- apply all external entities and all character and entity references in the element so that they are completely resolved

\(^1\) OTP does not assume any internal or external encoding format for OTP Messages. This is an implementation decision.
Chapter 3 Trading Components

- exclude comments and processing instructions (PIs),
- reduce all attributes to their canonical form using the attribute type in the DTD. Replace all single and double quotes present in attributes with &apos; and &quot; respectively so that attributes can be enclosed in double quotes
- create attributes, using their default value, which are not present in the original but have default values in the DTD
- sort the original and generated attributes in ascending attribute name order according to the UTF-16 encoding of the attribute name (i.e. not the native character ordering)
- for whitespace inside markup but not inside attribute values, generate it as minimally as possible. Specifically:
  - remove non essential whitespace, and
  - represent required whitespace by a single space character
- generate the content of all start tags using only the element name and the attributes as described above. If the element is an "empty" element then generate it using the single empty tag format, with a trailing slash. Generate end tags using only the element name, with no added whitespace.
- remove all whitespace in the element content
- all character data is generated inside a CDATA section. Any CDATA end sequences ("]]&gt;") within the data are replaced by "]]><![CDATA[&gt;" in order to escape the CDATA end sequence
- start tags, end tags, empty tags, CDATA sections, and text sections are assembled in the same order as the original document.

3.13.6.3 Calculating Hashes and Signatures

Hashes are calculated by:
- identifying "what to sign" - see above
- converting the data to be signed into a byte stream in the canonical encoding format defined above
- calculating the hash or signature according to the rules for a Hash Element (see section 3.13.2) or a DigSig Element (see section assuming 3.13.4) assuming "big-endian" byte order
- encoding the result using [Base64] encoding
- placing the encoded result, most significant byte first, in either:
  - the content of the Hash Element, or
  - the content of the DigSig Element.

---

This is consistent with both network byte order standards and Unicode/ISO10646 guidelines for the representation of multi-byte items.
3.13.6.4 Signature Hygiene

As a matter of cryptographic hygiene, signers should never sign data completely determined or predictable by another party. Every signature should include the hash of an element locally-generated by the signer and containing some externally unpredictable data. This ensures that an attacker cannot induce someone to sign a message that may be converted to another message with the same hash or signature value.

3.13.7 Checking Hashes and Signatures

Signature Components may be copied from one Trading Block to another. However the [XML] elements whose hashes are contained within the Signature Component are not necessarily copied as well.

For example, the Signature Component contained within an Offer Response Block (see section 4.3) may contain hashes of, inter-alia, the Order, the Pay Amount, and Delivery Components. This signature is then copied into a Payment Request Block (see section 4.6) together with the Pay Amount Component but not the Order or the Delivery Components.

This means that it may not be possible to completely check all the hashes within a Signature Component.

Checking of a signature therefore consists of:

• verifying that the SignedData element contains each Hash Element required by the usage of that signature in Trading Block in a particular OTP Transaction (see section 5)

• for each Hash Element (see section 3.13.2) in the signature:
  - check if the [XML] element identified by the ElRef attribute of the Hash Element refers to an element available from the same or a previous OTP message in the same OTP transaction, which has been received or sent by the particular Trading Role
  - if the [XML] element is available check that the hash value of the XML element contained inside the content of the Hash Element is correctly calculated (see Calculating Hashes and Signatures above)

• for the signature as a whole verifying that the content of the Digital Signature Element has been correctly calculated (see Calculating Hashes and Signatures above).

---

13 The Order and Delivery Components are not copied since the information they contain is not needed to carry out a Payment Exchange. This increases privacy for the Consumer since the Value Acquirer does not know, because she does not need to know, the reason why a payment is being made.

14 Only limited verification of the hashes in a signature is possible since, for example, the value acquirer will not have access to all the Trading Components being signed, e.g. the Order Component.
3.14 Certificate Component

A Certificate Component contains a digital certificate which is to be used in order to create or check a Signature Component (see section 3.13).

```xml
<!ELEMENT OtpCert ANY >
<!ATTLIST OtpCert
  CompId ID #REQUIRED
  CertType NMTOKEN 'x509v3'
  ContentFormat CDATA #REQUIRED>
```

Attributes:

- **CompId**: An identifier which uniquely identifies the Signature Component within the OTP Transaction.
- **CertType**: The Type of Certificate contained within the Trading Component. Valid values are:
  - x509v3 (see [X509])
  - x-nnn:ccc a user defined certificate type see section (2.7.3 User Defined Codes)
- **ContentFormat**: Defines the encoding used within the content of the Certificate Component. Its valid values are:
  - Xml The data is structured using the Extensible Markup Language [XML]
  - Mime The data consists of a [MIME] message
  - PCDATA The data contains data conforming to the [XML] rules for PCDATA but contains none of the structural features of XML
  - Base64 The data consists of binary information encoded using [Base64]
  - x-ddd:nnn The structure of the data is user defined (see section 2.7.3 User Defined Codes).

Content:

- **ANY**: The actual certificate of the type specified by CertType in the format specified by ContentFormat.

3.15 Fail Reason Component

The Fail Reason Component contains information about why the OTP Transaction failed. Its definition is as follows:

```xml
<!ELEMENT FailReason (FailTextParm*) >
<!ATTLIST FailReason
  CompId NMTOKEN #REQUIRED
  XML:Lang NMTOKEN #REQUIRED
  FailType (Cancel|Error) #REQUIRED
  FailCode NMTOKEN #REQUIRED
  FailText CDATA #REQUIRED
  OtpMsgRef NMTOKEN #REQUIRED
  BlkRef NMTOKEN #IMPLIED
  CompRef NMTOKEN #IMPLIED >
```
Chapter 3 Trading Components

Attributes:

- **CompId**: An identifier which uniquely identifies the Fail Reason Component within the OTP Transaction.
- **XML:Lang**: Defines the language used by attributes or child elements within this component, unless overridden by an XML:Lang attribute on a child element. See section 2.8 Identifying Languages.
- **FailType**: Indicates whether the failure is an error or a cancel.
- **FailCode**: Contains a failure code which indicates the nature of the failure. Codes which apply to an individual Trading Block message will be defined in the section which describes that Trading Block message.
- **FailText**: Contains the textual explanation in the language defined by XML:Lang that is associated with the FailCode. FailText may contain "parameters" in the format "&ppp;" which should be replaced by the matching FailTextParm parameter value (see below).
- **OtpMsgRef**: This is the ID of the OTP Message to which this FailReason applies, and:
  - which resulted in the OTP Transaction being cancelled, or
  - in which an error was found
- **BlkRef**: If the problem is associated with a specific Trading Block, then this is the Element Reference of the Trading Block:
  - which resulted in the transaction being cancelled, or
  - in which an error was found
- **CompRef**: If the problem is associated with a specific Trading Component, then this is the Element Reference of the Trading Component:
  - which resulted in the transaction being in the OTP Transaction being cancelled, or
  - in which an error was found

Content:

- **FailTextParm**: This contains a parameter which replaces the parameter in the FailText with the same name (see below).

3.15.1 Fail Text Parameter Element

The Fail Text Parameter Element contains a text field which must be used to replace the matching parameter within the FailText attribute of the FailReason element when the FailText is displayed or used. Its definition is as follows:

```xml
<!ELEMENT FailTextParm EMPTY>
<!ATTLIST FailTextParm
XML:Lang NMTOKEN #IMPLIED
ParmName NMTOKEN #REQUIRED
ParmValue CDATA #REQUIRED>
```

Attributes:

- **XML:Lang**: Defines the language used by attributes within this element. See section 2.8 Identifying Languages.
- **ParmName**: Contains the name of the parameter which is to be replaced in the language defined by XML:Lang.
ParmValue Contains the value of the parameter to be used in the language defined by $XML:Lang$.

The following rules apply:

- for each $FailTextParm$:
  - search for the value of the $ParmName$ attribute within the value of the $FailText$ attribute of the $FailReason$ element
  - if the search is successful, replace the matching text in the $FailText$, by the value of the $ParmValue$ attribute

- if the search in the previous step is unsuccessful, then leave $FailText$ unchanged

- when searching $FailText$ search for "&" plus the value of $ParmName$, i.e. the value of $ParmName$ does not include the "&"

- as a guideline, the value of $ParmName$ should, wherever possible, be the name of the XML attribute or element whose value or content is contained within $ParmValue$.

For example, the $FailText$ might have the following value ...

$FailText='Payment Brand "$Brand" not supported'$

... then, if the $FailTextParm$ was encoded as ...

$<FailTextParm ParmName='Brand' ParmValue='xzpay'/>$

... the $FailTextParm$ would be displayed as ...

'Payment Brand "xzpay" not supported'

### 3.16 Fail OTP Message Component

The Fail OTP Message Component contains the complete OTP message which resulted in the OTP Transaction failing. Its definition is as follows.

```xml
<!ELEMENT FailOtpMsg ANY >
<!ATTLIST FailReason
  CompId NMTOKEN #REQUIRED >
```

Attributes:
- **CompId**
  An identifier which uniquely identifies the Fail OTP Message Component within the OTP Transaction.

Content:
- **ANY**
  An [XML] CDATA Section containing the full content of the OTP Message last received which resulted in the error or cancellation.
4. Trading Blocks

Trading Blocks consist of one or more Trading Components and optionally one or more Signature Components. One or more Trading Blocks may be contained within the OTP Messages which are physically sent in the form of [XML] documents between the different organisations that are taking part in a trade.

This is illustrated in the diagram below.

**Figure 16 Trading Blocks**

Trading Blocks are defined as part of the definition of an OTP Message (see section 2.1.1). The definition of an OTP Message element is repeated here:

```xml
<!ELEMENT OtpMessage (TransRefBlk, ( TpoBlk | TpoSelectionBlk | AuthReqBlk | AuthRespBlk | OfferRespBlk | PayReqBlk | PayExchBlk | PayRespBlk | DelivReqBlk | DelivRespBlk | FailBlk )+)>
```
Chapter 4 Trading Blocks

The remainder of this section defines the Trading Blocks in this version of OTP. They are:

- TPO (Trading Protocol Options) Block
- TPO Selection Block
- Offer Response Block
- Authentication Request Block
- Authentication Response Block
- Payment Request Block
- Payment Exchange Block
- Payment Response Block
- Delivery Request Block
- Delivery Response Block
- Failure Block

The Trans Ref Block is described in section 2.3.

4.1 Trading Protocol Options Block

The TPO Trading Block contains options which apply to the OTP Transaction. The definition of a TPO Trading Block is as follows.

```xml
<!ELEMENT TpoBlk ( ProtocolOptions, BrandList* )>
<!ATTLIST TpoBlk
  BlkId ID #REQUIRED >
```

Attributes:

- **BlkId**: An identifier which uniquely identifies the Trading Protocol Options Block within the OTP Transaction (see section 2.4 ID Attributes).

Content:

- **ProtocolOptions**: The Protocol Options Component (see section 3.1) defines the options which apply to the whole OTP Transaction (see section 5).
- **BrandList**: This Brand List Component contains one or more payment brands and protocols which may be selected (see section 3.6).

4.2 TPO Selection Block

The TPO Selection Block contains the results of selections made from the options contained in the Trading Protocol Options Block (see section 4.1). The definition of a TPO Selection Block is as follows.

```xml
<!ELEMENT TpoSelectionBlk ( BrandSelection+ )>
```
4.3 Offer Response Block

The Offer Response Block contains details of the goods, services, amount, delivery instructions or financial transaction which is to take place. Its definition is as follows.

```xml
<!ELEMENT OfferRespBlk (AuthData*, Order, PayAmount*, Org*, Delivery?, OtpSig*, OtpCert*) >
<!ATTLIST OfferRespBlk
  BlkId ID #REQUIRED >
```

Attributes:
- **BlkId**: An identifier which uniquely identifies the Offer Response Block within the OTP Transaction.

Content:
- **AuthData**: The Authentication Data Component contains information about how Authentication associated with the Offer will occur. See section 3.2.
- **Order**: The Order Component contains details about the goods, services or financial transaction which is taking place see section 3.4.
- **PayAmount**: The Pay Amount Components contain information about the payments which are to be made see section 3.8.
- **Org**: The Organisation Components (see section 3.5) identify the organisations and their roles in the OTP Transaction. The roles and organisations which must be present will depend on the particular type of OTP Transaction. See the definition of each transaction in section 5. Open Trading Protocol Transactions.
- **Delivery**: The Delivery Component contains details of the delivery to be made (see section 3.11).
- **OtpSig**: The Signature Components contain Signatures (see section 3.13). The data which is digitally signed depends on the type of OTP Transaction and is defined in section 5. Open Trading Protocol Transactions.
- **OtpCert**: The Certificate Component (see section 3.14) which contains the optional certificate which is used by the Signature Component.
4.4 Authentication Request Block

This Authentication Request Block contains the challenge data which is used to authenticate one party by another. Its definition is as follows.

```xml
<!ELEMENT AuthReqBlk (AuthData) >
<!ATTLIST AuthReqBlk
  BlkId ID #REQUIRED >
```

Attributes

- **BlkId**: An identifier which uniquely identifies the Authentication Request Block within the OTP Transaction.

Content

- **AuthData**: The Authentication Data Component contains data about how Authentication associated with the Offer will occur. See section 3.2.

4.5 Authentication Response Block

The Authentication Response Block contains the response which results from processing the Authentication Request Block. Its definition is as follows.

```xml
<!ELEMENT AuthRespBlk (AuthResp) >
<!ATTLIST AuthRespBlk
  BlkId ID #REQUIRED >
```

Attributes:

- **BlkId**: An identifier which uniquely identifies the Authentication Response Block within the OTP Transaction.

Content:

- **AuthResp**: The Authentication Response Component which contains the results of processing the challenge data in the Authentication Data Component - see section 3.3.

4.6 Payment Request Block

The Payment Request Block contains information which requests that a payment is started. Its definition is as follows.

```xml
<!ELEMENT PayReqBlk (AuthData?, BrandList, BrandSelection, PayAmount, PaySchemeData?, Org*, OtpSig?, OtpCert?) >
<!ATTLIST PayReqBlk
  BlkId ID #REQUIRED >
```
Chapter 4 Trading Blocks

Attributes:

BlkId
An identifier which uniquely identifies the Payment Request Block within the OTP Transaction.

Content:

AuthData
The optional Authentication Data Component contains data about how Authentication associated with the payment, if any, will occur. See section 3.2.

BrandList
The Brand List Component contains a list of one or more payment brands and protocols which may be selected (see section 3.6).

BrandSelection
This identifies the choice of payment brand and payment protocol to be used in a payment within the OTP Transaction. There is one Brand Selection Component (see section 3.7) for each payment to be made in the OTP Transaction.

PayAmount
The Pay Amount Components contain information about the payment which is being made see section 3.8.

PaySchemeData
The Payment Scheme Component contains payment scheme specific data see section 3.9.

Org
The Organisation Component contains details of organisations involved in the payment (see section 3.5). The Organisations present are dependent on the OTP Transaction - see section 5.

OtpSig
The Signature Component contains a digital signature (see section 3.13) which is to be used by the value acquirer to verify that:

- the Payment Request Block has not been changed
- the value acquirer has a relationship with the organisation on whose behalf the value is being acquired.

OtpCert
The Certificate Component (see section 3.14) which contains the optional certificate which is used by the Signature Component.

4.7 Payment Exchange Block

The Payment Exchange Block contains payment scheme specific data which is exchanged between two of the roles in a trade. Its definition is as follows.

<!--ELEMENT PayExchBlk (PaySchemeData) -->
<!--ATTLIST PayExchBlk
   BlkId ID #REQUIRED -->

Attributes:

BlkId
An identifier which uniquely identifies the Payment Exchange Block within the OTP Transaction.

Contents:

PaySchemeData
This Trading Component contains payment scheme specific data see section 3.9 Payment Scheme Component.
4.8 Payment Response Block

This Payment Response Block contains a Pay Receipt, an optional signature and an optional payment protocol message. Its definition is as follows.

```xml
<!ELEMENT PayRespBlk (PayReceipt, PaySchemeData?, OtpSig*, OtpCert?) >
<!ATTLIST PayRespBlk
    BlkId ID #REQUIRED >
```

Attributes:
- **BlkId**: An identifier which uniquely identifies the Payment Response Block within the OTP Transaction.

Contents:
- **PayReceipt**: Contains payment scheme specific data which can be used to verify the payment occurred. See section 3.10 Pay Receipt Component.
- **PaySchemeData**: Contains payment scheme specific data see section, for example a payment protocol message. See 3.9 Payment Scheme Component.
- **OtpSig**: Optionally contains signatures which bind the record of the payment (i.e. the PayReceipt) to the reason the payment occurred. The signatures required are explained in the definition of each OTP Transaction in section 5.
- **OtpCert**: The Certificate Component (see section 3.14) which contains the optional certificate which is used by the Signature Component.

4.9 Delivery Request Block

The Delivery Request Block contains details of the goods or services which are to be delivered together with a signature which can be used to check that delivery is authorised. Its definition is as follows.

```xml
<!ELEMENT DelivReqBlk (Order, Org*, Delivery, OtpSig?, OtpCert?) >
<!ATTLIST DelivReqBlk
    BlkId ID #REQUIRED >
```

Attributes:
- **BlkId**: An identifier which uniquely identifies the Delivery Request Block within the OTP Transaction.

Contents:
- **Order**: The Order Component contains details about the goods, services or financial transaction which is taking place see section 3.4.
- **Org**: The Organisation Components (see section 3.5) identify the organisations and their roles in the OTP Transaction. The roles and organisations which must be present will depend on the particular type of OTP Transaction. See the definition of each transaction in section 5. Open Trading Protocol Transactions.
- **Delivery**: The Delivery Component contains details of the delivery to be made (see
section 3.11).

The Signature Components contain Signatures (see section 3.13). The data which is digitally signed depends on the type of OTP Transaction and is defined in section 5. Open Trading Protocol Transactions

The Certificate Component (see section 3.14) which contains the optional certificate which is used by the Signature Component.

### 4.10 Delivery Response Block

The Delivery Response Block contains a Delivery Note containing details on how the goods will be delivered. Its definition is as follows.

```xml
<!ELEMENT DelivRespBlk (DeliveryNote) >
<!ATTLIST DelivRespBlk
   BlkId ID #REQUIRED >
```

**Attributes:**

- **BlkId**: An identifier which uniquely identifies the Delivery Response Block within the OTP Transaction.

**Contents:**

- **DeliveryNote**: The Delivery Note Component contains details about how the goods or services will be delivered (see section 3.12).

### 4.11 Fail Trading Block

Each of the definitions of an OTP Transaction assume that they will reach a normal conclusion and will not terminate abnormally. This will not necessarily be the case since any of the parties to an OTP Transaction may terminate a transaction at any point for one of two reasons:

- **an Error**: The OTP Message did not conform to this specification and therefore prevents the transaction from continuing, or
- **a Cancel**: The OTP transaction is not acceptable to one or other of the Trading Roles involved for some reason.

This means that, instead of an OTP Message being received containing the expected Trading Blocks, an OTP Message containing a Fail Trading Block may be received instead.

**Note:** This version of the OTP specification does not include any "fail codes" which can be used with the Fail Trading Block. These will be produced once initial feedback on this specification has been received and considered.

The structure of a Fail Trading Block is as follows.

```xml
<!ELEMENT FailBlk (FailReason+, FailOtpMsg+) >
```
Chapter 4 Trading Blocks

<!ATTLIST FailBlk
  BlkId ID #REQUIRED >

Attributes:
  BlkId
  An identifier which uniquely identifies the Fail Trading Block within the OTP Transaction.

Content:
  FailReason
  Contains information which explains the reason why the OTP Transaction failed (see below).
  FailOtpMsg
  Contains the full content of any OTP Message which resulted in the error or cancellation.

The following processing rules apply:

- the OTP Message which contains a Fail Trading Block is sent back to the net location which sent the OTP Message which caused the Fail Trading Block to be created
- once the OTP Message containing the Fail Trading Block has been sent, the consumer directs the browser to use the net locations specified in the Protocol Options Component (see section 3.1)
- only one Fail Trading Block may be contained in an OTP Message
- if an OTP Message contains a Fail Trading Block then no other types of Trading Blocks can be contained in the same OTP Message
- FailReasons with a FailType of Error or Cancel may be contained within the same Fail Trading Block
- the Status attribute of the OTP Message transaction reference element must be set to Fail
- the MsgSeq attribute on the OTP Message is set to LastMsg.

Sending back the Fail Trading Block to the net location which sent the OTP Message which caused the Fail Trading Block to be created means, for example:

- if the Merchant sent an OTP Message to the Consumer which the consumer did not accept, for example an Offer, then the Consumer sends the Fail Trading Block back to the Merchant
- if the Consumer is sent an OTP Message by the Value Acquirer which the Consumer did not like for some reason, for example a Pay Response, then the Consumer sends the Fail Trading Block back to the Value Acquirer.

Both of the above examples mean that it is possible to receive a message after an OTP Transaction would be complete for a specific role. For example:

- a Merchant might expect payment to be accepted by a Value Acquirer, however, if the Consumer did not like the Offer Response sent by the Merchant, then the consumer would send a message back to the Merchant
- if a Consumer did not like the Pay Response received from a Value Acquirer, then the Value Acquirer would receive a Fail Trading Block in response
5. Open Trading Protocol Transactions

The Baseline Open Trading Protocol supports the following types of OTP Transaction:

- Baseline Authentication
- Baseline Deposit
- Baseline Purchase
- Baseline Refund
- Baseline Withdrawal
- Baseline Transaction Status Inquiry, and
- Baseline Value Exchange

Each of these transactions are described in more detail in the following sections providing descriptions of:

- the Trading Blocks in each OTP Transaction
- the Trading Components in each Trading Block, and
- how the Trading Components are signed

5.1 Baseline Authentication OTP Transaction

The Baseline Authentication OTP Transaction supports the remote authentication of one party by another using a variety of authentication methods. A typical use is likely to be when the Baseline Authentication OTP Transaction takes place as an early part of a session where strong continuity exists. For example, a Financial Institution could:

- set up a secure channel (e.g. using SSL) with a customer
- authenticate the customer using the Baseline Authentication OTP Transaction, and then
- provide the customer with access to account information and other services with the confidence that they are communicating with a bona fide customer.

The Baseline Authentication OTP Transaction consists of just the Authentication Trading Exchange (see section 1.2.4).

The Authentication Exchange is implemented by a set of predefined OTP Messages (see section 2.2) which are exchanged between the Trading Roles (see section 1.1). Each OTP Message contains Trading Blocks (see section 3.15) which contain the Trading Components (see section 3) which are required by the Trading Exchanges.

The Trading Blocks used by the Baseline Authentication OTP Transaction are:

- TPO (Trading Protocol Options) Block
- Authentication Request Block, and
Chapter 5 Open Trading Protocol Transactions

- Authentication Response Block

There are no variations in the Baseline Authentication OTP Transaction.

The OTP Messages used in a Baseline Authentication are illustrated in the diagram below.

5.1.1 TPO (Trading Protocol Options) Block

The TPO (Trading Protocol Options) Block (see section 5.3.2) must contain the following Trading Component:

- one Protocol Options Component which defines the options which apply to the whole OTP Transaction. See Section 3.1.

There are no Brand List Components in the TPO Block (see section 3.6).

5.1.2 Authentication Request Block

The Authentication Request Block (see section 4.4) must contain the following Trading Component:

- one Authentication Data Component (see section 3.2)
5.1.3 Authentication Response Block

The Authentication Response Block (see section 4.5) must contain the following Trading Component:
- one Authentication Response Component (see section 3.3).

5.2 Baseline Deposit OTP Transaction

The Baseline Deposit OTP Transaction supports the deposit of electronic cash with a Financial Institution\(^{15}\). It consists of the following Trading Exchanges:
- an optional Authentication Exchange (see section 1.2.4),
- an Offer Exchange (see section 1.2.1), and
- a Payment Exchange (see section 1.2.2).

These Trading Exchanges are implemented by a set of predefined OTP Messages (see section 2.2) which are exchanged between the Trading Roles (see section 1.1). Each OTP Message contains Trading Blocks (see section 3.15) which contain the Trading Components (see section 3) which are required by the Trading Exchanges.

The Trading Blocks used by the Baseline Purchase OTP Transaction are:
- TPO (Trading Protocol Options) Block
- TPO Selection Block
- Authentication Request Block
- Authentication Response Block
- Offer Response Block
- Payment Request Block
- Payment Exchange Block
- Payment Response Block

5.2.1 Baseline Deposit Variations

The Baseline Deposit OTP Transaction occurs in two basic forms:
- Baseline Deposit with Authentication. Where the Consumer making the deposit is authenticated before the deposit is made, and

---

\(^{15}\) The Financial Institution has, in OTP terminology, a role of merchant in that a service (i.e. a deposit of electronic cash) is being offered in return for a fee, for example bank charges of some kind. The term “Financial Institution” is used in the diagrams and in the text for clarity.
5.2.2 Baseline Deposit Authentication

In Baseline Deposit with Authentication an Authentication Exchange occurs before the Offer Exchange containing the details of the deposit is provided by the Financial Institution.

In Baseline Deposit without Authentication, there is no Authentication Exchange and the Financial Institution provides details about the deposit immediately at the start of the OTP Transaction.

These two alternatives are illustrated in the two diagrams below. The first diagram illustrates the case when an Authentication Exchange is included.

![Diagram of Baseline Deposit with Authentication](image)

Figure 18 Baseline Deposit with Authentication

Note that the above diagram:

- describes the general case where a Merchant can accept a deposit in several different types of electronic cash. In practice usually only one form of electronic cash may be accepted. However, there may be several different protocols which may be used for the same "brand" of electronic cash.
Internet Open Trading Protocol Part 2: Specification, V 0.9

Chapter 5 Open Trading Protocol Transactions

- the financial institution may use the results of the authentication to identify not only the consumer but also the account to which the value is to be deposited. If no single account can be identified, then it must be obtained by other means. For example:
  - the consumer could specify the account number in the initial dialogue (see step 1), or
  - the consumer could have been identified earlier, for example using a Baseline Authentication OTP Transaction, and an account selected from a list provided by the Financial Institution.

The second diagram illustrates the case when an Authentication Exchange is not included.

The Baseline Deposit without authentication might be used:

- if a previous OTP transaction, for example a Baseline Withdrawal or a Baseline Authentication, authenticated the consumer, and a secure channel has been maintained, therefore the authenticity of the consumer is known
- if authentication is achieved as part of a proprietary payment protocol and is therefore included in the Payment Exchange
- if authentication of the consumer has been achieved by some other means outside of the scope of OTP, for example, by using a pass phrase.

OTP aware applications supporting the Consumer Trading Role must check for the existence of an Authentication Request Block in the first OTP Message to determine whether the Baseline Deposit includes an Authentication Exchange or not.
5.2.3 Baseline Deposit Payment Messages

Once the Offer Response Trading Block has been received, the sequence of OTP Messages illustrated in Figure 20 occurs. These are the same whether or not an Authentication of the Consumer has occurred. Note that these continue where the previous diagrams (Figure 18 and Figure 19) finish.

The remainder of this sub-section on the Baseline Deposit OTP Transaction defines the contents of each Trading Block. For most Trading Blocks, the content does not alter with the variations described above. Where differences apply, these are stated.

5.2.4 TPO (Trading Protocol Options) Block

The TPO (Trading Protocol Options) Block (see section 5.3.2) must contain the following Trading Components:

- one Protocol Options Component which defines the options which apply to the whole OTP Transaction. See Section 3.1.
- one Brand List Component (see section 3.6) which contains the payment brand and protocols which may be selected for use in the Payment Exchange.
5.2.5 TPO Selection Block

The TPO Selection Block (see section 4.2) is only used by Baseline Deposit with Authentication. It contains:

- one Brand Selection Component (see section 3.7) for use in the Payment Exchange. It contains the results of the consumer selecting a Payment Brand and Payment Protocol from the list provided in the Brand List Component.

5.2.6 Authentication Request Block

The Authentication Request Block (see section 4.4) must contain the following Trading Component:

- one Authentication Data Component (see section 3.2)

5.2.7 Authentication Response Block

The Authentication Response Block (see section 4.5) must contain the following Trading Component:

- one Authentication Response Component (see section 3.3).

5.2.8 Offer Response Block

The Offer Response Block (see section 4.3) must contain the following components:

- zero or one Authentication Data Component (see section 3.2) An Authentication Data Component is required for each Payment Exchange, where its Pay Amount Component contains an AuthDataRef attribute
- one Order Component (see section 3.4) which contains details about the deposit, for example the amount of value being deposited and any fees which might apply
- one Pay Amount Component (see section 3.8) which contains information about the payment which is to be made
- Organisation Components (see section 3.5) with the following roles:
  - the Merchant\textsuperscript{16} who is accepting the deposit
  - the Consumer who is making the deposit
  - the ValueAcquirer for the payment. The "ID" of the Value Acquirer Organisation Component is contained within the VaOrgRef attribute of the Pay Amount Component
- one Delivery Component (see section 3.11) with the NoDelivExch attribute set to True.

\textsuperscript{16} A role of Merchant is used in that a service (i.e. a deposit of electronic cash) is being offered in return for a fee, for example bank charges of some kind. The term "Financial Institution" is used in the diagrams and in the text for clarity.
Chapter 5 Open Trading Protocol Transactions

If the Baseline Deposit Offer Response is being digitally signed then the Offer Response Block must contain an "Offer" Signature Component (see section 3.13). The Signature Component contains hashes of the following XML elements:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Offer Response Block within the OTP Transaction. It contains information that identifies the OTP Message and OTP Transaction
- the Transaction Id Component (see section 2.3.1) which globally uniquely identifies the OTP Transaction
- the following components of the Offer Response Block:
  - the Authentication Data Component if present
  - the Order Component
  - the Pay Amount Component
  - all the Organisation Components present, and
  - the Delivery Component,
- the following components of the TPO Block:
  - the Protocol Options Component, and
  - the Brand List Component

If the Baseline Deposit is a Baseline Deposit with Authentication then the Signature Component additionally contains a hash of the following:

- the Brand Selection Component contained in the TPO Selection Block.

5.2.9 Payment Request Block

The Payment Request Block (see section 4.6) contains:

- the following components copied from the Offer Response Block:
  - the Authentication Data Component if present
  - the Pay Amount Component
  - the Organisation Components with the roles of Merchant and ValueAcquirer
  - the "Offer" Signature Component if present
- the following component from the TPO Block:
  - the Brand List Component
- one Brand Selection Component either:
  - copied from the Offer Response Block if the deposit is a Baseline Deposit with Authentication, or
  - created by the Consumer, containing the payment brand and payment protocol selected, if the deposit is a Baseline Deposit without Authentication
- one Payment Scheme Component (see section 3.9) if required by the payment method used (see the Payment Method supplement to determine if this is needed).

If signatures are used then the Value Acquirer should check that the Signature Component is valid (see section 3.13.7 Checking Hashes and Signatures).
5.2.10 Payment Exchange Block

The Payment Exchange Block (see section 4.7) contains:

- one Payment Scheme Component (see section 3.9) which contains payment method specific data. See the Payment Method supplement for the payment method being used to determine what this should contain.

5.2.11 Payment Response Block

The Payment Response Block (see section 4.8) contains:

- one Pay Receipt Component (see section 3.10) which contains scheme specific data which can be used to verify the payment occurred
- one Payment Scheme Component (see section 3.9) if required which contains payment method specific data. See the Payment Method supplement for the payment method being used to determine what this should contain
- the "Offer" Signature Component (see section 3.13) from the Payment Request Block if present.

If a signed pay receipt is being provided, indicated by the SignedPayReceipt attribute of the Pay Amount Component of the Offer Response Block being set to True, then the Payment Response Block additionally contains one extra "Pay Receipt" Signature Component which contains hashes of the following:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Payment Response Block,
- the Transaction Id Component (see section 2.3.1) within the Trans Ref Block that globally uniquely identifies the OTP Transaction,
- the Pay Receipt Component from the Payment Response Block and
- the "Offer" Signature Component from the Payment Request Block if present.

5.3 Baseline Purchase OTP Transaction

The Baseline Purchase OTP Transaction supports the purchase of goods or services using any payment method. It consists of the following Trading Exchanges:

- an Offer Exchange (see section 1.2.1),
- a Payment Exchange (see section 1.2.2), and
- an optional Delivery Exchange (see section 1.2.3)

These Trading Exchanges are implemented by a set of predefined OTP Messages (see section 2.2) which are exchanged between the Trading Roles (see section 1.1). Each OTP Message contains Trading Blocks (see section 3.15) which contain the Trading Components (see section 3) which are required by the Trading Exchanges.
Chapter 5 Open Trading Protocol Transactions

The Trading Blocks used by the Baseline Purchase OTP Transaction are:

- TPO (Trading Protocol Options) Block
- TPO Selection Block
- Offer Response Block
- Payment Request Block
- Payment Exchange Block
- Payment Response Block
- Delivery Request Block
- Delivery Response Block

If the Baseline Purchase OTP Transaction fails, then in a future version of OTP there will be a Restart OTP Transaction (see section 5.7 Restart Transaction (provisional)) which, if possible, will allow the transaction to be restarted. This will be particularly useful if the transaction fails between the payment and the delivery since it would permit the delivery to occur.

5.3.1 Baseline Purchase Variations

The Baseline Purchase OTP Transaction occurs in two basic forms:

- Brand Dependent Purchase. Where the content of the offer, e.g. the order details, amount, delivery details, etc., are dependent on the payment brand and protocol selected by the consumer, and
- Brand Independent Purchase. Where the content of the offer is not dependent on the payment brand and protocol selected.

Further variation is supported in that:

- the Delivery Exchange is optional, and
- the Delivery Response Block may be sent to the consumer either:
  - at the same time as the Payment Response Block, or
  - after the Payment Response Block as the result of the Consumer sending the Deliverer a Delivery Request Block.

5.3.1.1 Brand Dependent Purchases

In a Brand Dependent Purchase the TPO Block and the Offer Response Block are sent separately by the Merchant to the Consumer, i.e.:

- the Brand List Component is sent to the Consumer in a TPO Block,
- the Consumer selects a Payment Brand and Payment Protocol from the Brand List Component
- the Consumer sends the selected brand and protocol back to the Merchant in a TPO Selection Block, and
- the Merchant uses the information received to define the content of and then send the Offer Response Block to the Consumer.
In a Brand Independent Purchase the TPO Block and the Offer Response Block are sent together by the Merchant to the Consumer in the same OTP Message at the start of the OTP Transaction.

These two alternatives are illustrated in the two diagrams below. The first diagram illustrates a Brand Dependent Purchase.

**Figure 21 Brand Dependent Baseline Purchase**
Chapter 5 Open Trading Protocol Transactions

The second diagram illustrates the Brand Independent Purchase.

1. Consumer decides to trade and sends information about what to purchase to the Merchant, e.g. using HTML

2. Merchant decides which payment brand and protocols to offer, places them in a Pay Options Component in a TPO Block, creates an Offer Response Block containing details about goods ordered, price, etc and sends both to Consumer

3. OTP aware application started. Consumer selects the payment brand and payment protocol to use, records selection in a Pay Selection Component, checks Offer is OK, combines the Pay Selection Component with information from the TPO Block and Offer Response Block to create a Pay Request Block and sends it to the Value Acquirer

4. Value Acquirer processes Pay Request Block and carries out payment protocol ...

Figure 22 Brand Independent Baseline Purchase

A Brand Independent Purchase always occurs when only one payment brand and protocol is being offered to the Consumer by the Merchant. It is also likely to, but will not necessarily, occur when multiple brands are being offered, the Value Acquirer is the same, and all brands use the same set of protocols.

Note that the TPO Block and the Offer Response Block may be sent in separate OTP messages even if the Offer Response Block does not change. However this increases the number of messages in the transaction and is therefore likely to increase transaction response times.

The content of the Offer Response Block may vary, for example, since:

- the price to pay may vary as a result of selecting a particular payment brand, for example a store credit card brand may offer a discount;
- the value acquirer may differ depending on the payment brand selected.

OTP aware applications supporting the Consumer Trading Role must check for the existence of an Offer Response Block in the first OTP Message to determine whether the Baseline Purchase is brand dependent or not.

5.3.1.2 Combining Delivery Response Block and Pay Response Block

The Delivery Response Block and the Pay Response Block may be sent:

- separately by the Value Acquirer to the Consumer, i.e.:
- the Pay Response Block containing a Pay Receipt and optional signature for the payment is sent by the Value Acquirer to the Consumer,
- the Consumer combines these components from the Pay Response Block with components from the Offer Response Block, to create a Delivery Request Block
- the Consumer sends the Delivery Request Block to the Deliverer
- the Deliverer processes the Delivery Request Block and sends a Delivery Response Block back to the Consumer, or

- together, from the Value Acquirer to the Consumer, when the Payment Exchange is complete.

These two alternatives are illustrated in the two diagrams below.

The first diagram illustrates when the Delivery Response Block and the Pay Response Block are sent to the Consumer in separate OTP Messages. Note, these diagrams continue where the previous diagrams (Figure 21 and Figure 22) finish.
The Delivery Response Block and the Payment Response Block may be combined into the same OTP Message only if the Value Acquirer has the information available so that she can send the Delivery Response Block. This is likely to, but will not necessarily, occur when the Merchant, the Value Acquirer and the Deliverer Roles are combined.

The DelivAndPayResp attribute of the Delivery Component (see section 3.11) contained within the Offer Response Block (see section 4.3) is set to True if the Delivery Response Block and the Payment Response Block are combined into the same OTP Message and is set to False if the Delivery Response Block and the Payment Response Block are sent in separate OTP Messages.
The final variation of the Baseline Purchase OTP Transactions is a purchase without a delivery step. This is illustrated in the following diagram which continue where the earlier diagrams (Figure 21 and Figure 22) finish.

**Figure 25 Baseline Purchase, Purchase without Delivery Exchange**

The NoDelivExch attribute of the Delivery Component (see section 3.11) contained in the Offer Response Block (see section 4.3) is set to True if the Delivery Exchange is omitted and is set to False if the Delivery Exchange is included.
5.3.1.4 Combining Variations

The diagram below shows how the different variations in the Baseline Purchase Transaction may be combined.

![Diagram showing variations in Baseline Purchase Transaction]

The remainder of this sub-section on the Baseline Purchase OTP Transaction defines the contents of each Trading Block. For most Trading Blocks, the content does not alter with the variations described above. Where differences apply, these are stated.
5.3.2 TPO (Trading Protocol Options) Block

The TPO (Trading Protocol Options) Block (see section 5.3.2) must contain the following Trading Components:

- one Protocol Options Component which defines the options which apply to the whole OTP Transaction. See Section 3.1.
- one Brand List Component (see section 3.6) which contains one or more payment brands and protocols which may be selected for use in the Payment Exchange.

5.3.3 TPO Selection Block

The TPO Selection Block (see section 4.2) is only used by Brand Dependent Purchase. It contains:

- one Brand Selection Component (see section 3.7) for use in the Payment Exchange. It contains the results of the consumer selecting a Payment Brand and Payment Protocol from the list provided in the Brand List Component.

5.3.4 Offer Response Block

The Offer Response Block (see section 4.3) contains the following components:

- zero or one Authentication Data Component (see section 3.2) An Authentication Data Component is required for each Payment Exchange, where its Pay Amount Component contains an AuthDataRef attribute.
- one Order Component (see section 3.4) which contains details about the goods, services which are being purchased
- one Pay Amount Component (see section 3.8) which contains information about the payment which is to be made
- Organisation Components (see section 3.5) with the following roles:
  - Merchant who is providing the goods or services
  - Consumer who is making the purchase
  - ValueAcquirer for the payment. The "ID" of the Value Acquirer Organisation Component is contained within the VaOrgRef attribute of the Pay Amount Component
- one Delivery Component (see section 3.11) which contains details of the delivery to be made.

If the Baseline Purchase includes a Delivery Exchange then the Offer Response Block must also contain:

- Organisation Components with the following roles:
  - Deliverer who will be delivering the goods or services
  - DelivTo i.e. the person or organisation which is to take delivery
Chapter 5 Open Trading Protocol Transactions

If the Baseline Purchase Offer Response is being digitally signed then the Offer Response Block must contain an "Offer" Signature Component (see section 3.13). The Signature Component contains hashes of the following XML elements:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Offer Response Block within the OTP Transaction. It contains information that identifies the OTP Message and OTP Transaction
- the Transaction Id Component (see section 2.3.1) which globally uniquely identifies the OTP Transaction
- the following components of the Offer Response Block:
  - the Authentication Data Component if present
  - the Order Component
  - the Pay Amount Component
  - all the Organisation Components present, and
  - the Delivery Component,
- the following components of the TPO Block:
  - the Protocol Options Component, and
  - the Brand List Component

If the Baseline Purchase is a Brand Dependent Purchase then the Signature Component additionally contains a hash of the following:

- the Brand Selection Component contained in the TPO Selection Block.

5.3.5 Payment Request Block

The Payment Request Block (see section 4.6) contains:

- the following components copied from the Offer Response Block:
  - the Authentication Data Component if present
  - the Pay Amount Component
  - the Organisation Components with the roles of Merchant and ValueAcquirer
  - the Signature Component if present
- the following component from the TPO Block:
  - the Brand List Component
- one Brand Selection Component either:
  - copied from the Offer Response Block if the purchase is a Brand Dependent Purchase, or
  - created by the Consumer, containing the payment brand and payment protocol selected, if the purchase is a Brand Independent Purchase
- one Payment Scheme Component (see section 3.9) if required by the payment method used (see the Payment Method supplement to determine if this is needed).

If signatures are used then the Value Acquirer should check that the Signature Component is valid (see section 3.13.7 Checking Hashes and Signatures).
5.3.6 Payment Exchange Block

The Payment Exchange Block (see section 4.7) contains:

- one Payment Scheme Component (see section 3.9) which contains payment method specific data. See the Payment Method supplement for the payment method being used to determine what this should contain.

5.3.7 Payment Response Block

The Payment Response Block (see section 4.8) contains:

- one Pay Receipt Component (see section 3.10) which contains scheme specific data which can be used to verify the payment occurred
- one Payment Scheme Component (see section 3.9) if required which contains payment method specific data. See the Payment Method supplement for the payment method being used to determine what this should contain
- the "Offer" Signature Component (see section 3.13) from the Payment Request Block if present.

If a signed pay receipt is being provided, indicated by the SignedPayReceipt attribute of the Pay Amount Component of the Offer Response Block being set to True, then the Payment Response Block additionally contains one extra "Pay Receipt" Signature Component which contains hashes of the following:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Payment Response Block,
- the Transaction Id Component (see section 2.3.1) within the Trans Ref Block that globally uniquely identifies the OTP Transaction,
- the Pay Receipt Component from the Payment Response Block and
- the "Offer" Signature Component from the Payment Request Block if present.

5.3.8 Delivery Request Block

The Delivery Request Block (see section 4.9) contains:

- the following components copied from the Offer Response Block:
  - the Order Component (see section 3.4)
  - the Organisation Component (see section 3.5) with the roles of: Merchant, Deliverer and DeliverTo
  - the Delivery Component (see section 3.11)
  - the Signature Component (see section 3.13), if present
- the following component copied from the Pay Response Block:
  - the "Pay Receipt" Signature Component

If signatures are used then the Deliverer should check that all Signature Components are valid (see section 3.13.7 Checking Hashes and Signatures).
5.3.9 Delivery Response Block

The Delivery Response Block contains:

• one Delivery Note Component (see section 3.12) which contains delivery instructions about the delivery of goods or services

5.4 Baseline Refund OTP Transaction

In business terms the refund process typically consists of:

• a request for a refund being made by the Consumer to the Merchant, typically supported by evidence to demonstrate:
  - the original trade took place, for example by providing a receipt for the original transaction
  - using some type of authentication, that the consumer requesting the refund is the consumer, or a representative of the consumer, who carried out the original trade
  - the reason why the merchant should make the refund

• the merchant agreeing (or not) to the refund. This may involve some negotiation between the Consumer and the Merchant, and, if the merchant agrees,

• the refund of value by the Merchant to the Consumer.

The Baseline Refund OTP Transaction supports a subset of the above, specifically it supports:

• the optional authentication of the Consumer using an Authentication Exchange (see section 1.2.4), and

• the refund of value from the Merchant to the Consumer using the following two Trading Exchanges:
  - an Offer Exchange (see section 1.2.1), and
  - a Payment Exchange (see section 1.2.2).

These Trading Exchanges are implemented by a set of predefined OTP Messages (see section 2.2) which are exchanged between the Trading Roles (see section 1.1). Each OTP Message contains Trading Blocks (see section 3.15) which contain the Trading Components (see section 3) which are required by the Trading Exchanges.

The Trading Blocks used by the Baseline Purchase OTP Transaction are:

• TPO (Trading Protocol Options) Block
• TPO Selection Block
• Authentication Request Block
• Authentication Response Block
• Offer Response Block
• Payment Request Block
• Payment Exchange Block
• Payment Response Block
5.4.1 Baseline Refund Variations

The Baseline Refund OTP Transaction occurs in two basic forms:

- Baseline Refund with Authentication. Where the Consumer requesting the refund is authenticated before the refund is made, and
- Baseline Refund without Authentication. Where the Consumer is not authenticated before the refund is made.

5.4.2 Baseline Refund Authentication

In Baseline Refund with Authentication an Authentication Exchange occurs before the Offer Exchange containing the details of the refund is provided by the Merchant.

In Baseline Refund without Authentication, there is no Authentication Exchange and the Merchant provides details about the refund immediately at the start of the OTP Transaction.

These two alternatives are illustrated in the two diagrams below. The first diagram illustrates the case when an Authentication Exchange is included.

![Figure 27 Baseline Refund with Authentication](image-url)
Chapter 5 Open Trading Protocol Transactions

The second diagram illustrates the case when an Authentication Exchange is not included.

1. Consumer requests payment of previously agreed refund, sends information about refund to the Merchant, such as a reference number, using, for example, HTML.

2. Merchant sets the payment brand and decides which protocols to offer, generates an Offer Response Block containing information about the refund and sends to the Consumer.

3. OTP aware application started. Consumer selects the payment protocol to use, records selection in a Pay Selection Component, checks Offer is OK, combines the Pay Selection Component with information from the TPO Block and Offer Response Block to create a Pay Request Block and sends it to the Value Acquirer.

Figure 28 Baseline Refund without Authentication

The Baseline Refund without authentication might be used:

- when authentication of the consumer has been achieved by some other means, for example, the consumer has entered some previously supplied code in order to identify herself and the refund to which applies. The code could be supplied, for example on a web page or by e-mail.

- when a previous OTP transaction, for example a Baseline Authentication, authenticated the consumer, and a secure channel has been maintained, therefore the authenticity of the consumer is known and therefore the previously agreed refund can be identified.

OTP aware applications supporting the Consumer Trading Role must check for the existence of an Authentication Request Block in the first OTP Message to determine whether the Baseline Refund includes an Authentication Exchange or not.
5.4.3 Baseline Refund Payment Messages

Once the Offer Response Trading Block has been received, the sequence of OTP Messages illustrated in Figure 29 occurs. These are the same whether or not an Authentication of the Consumer has occurred. Note that these continue where the previous diagrams (Figure 27 and Figure 28) finish.

The remainder of this sub-section on the Baseline Refund OTP Transaction defines the contents of each Trading Block. For most Trading Blocks, the content does not alter with the variations described above. Where differences apply, these are stated.

5.4.4 TPO (Trading Protocol Options) Block

The TPO (Trading Protocol Options) Block (see section 5.3.2) must contain the following Trading Components:

- one Protocol Options Component which defines the options which apply to the whole OTP Transaction. See Section 3.1.
- one Brand List Component (see section 3.6) which contains the payment brand and protocols which may be selected for use in the Payment Exchange.
5.4.5 TPO Selection Block

The TPO Selection Block (see section 4.2) is only used by Baseline Refund with Authentication. It contains:

- one Brand Selection Component (see section 3.7) for use in the Payment Exchange. It contains the results of the consumer selecting a Payment Brand and Payment Protocol from the list provided in the Brand List Component.

5.4.6 Authentication Request Block

The Authentication Request Block (see section 4.4) must contain the following Trading Component:

- one Authentication Data Component (see section 3.2)

5.4.7 Authentication Response Block

The Authentication Response Block (see section 4.5) must contain the following Trading Component:

- one Authentication Response Component (see section 3.3).

5.4.8 Offer Response Block

The Offer Response Block (see section 4.3) must contain the following components:

- zero or one Authentication Data Component (see section 3.2) An Authentication Data Component is required for each Payment Exchange, where its Pay Amount Component contains an AuthDataRef attribute
- one Order Component (see section 3.4) which contains details about the refund, for example the amount of value being refunded and any conditions which might apply
- one Pay Amount Component (see section 3.8) which contains information about the payment which is to be made
- Organisation Components (see section 3.5) with the following roles:
  - the Merchant who is making the refund
  - the Consumer who is requesting the refund
  - the ValueAcquirer for the payment. The "ID" of the Value Acquirer Organisation Component is contained within the VaOrgRef attribute of the Pay Amount Component
- one Delivery Component (see section 3.11) with the NoDelivExch attribute set to True.
If the Baseline Refund Offer Response is being digitally signed then the Offer Response Block must contain an "Offer" Signature Component (see section 3.13). The Signature Component contains hashes of the following XML elements:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Offer Response Block within the OTP Transaction. It contains information that identifies the OTP Message and OTP Transaction
- the Transaction Id Component (see section 2.3.1) which globally uniquely identifies the OTP Transaction
- the following components of the Offer Response Block:
  - the Authentication Data Component if present
  - the Order Component
  - the Pay Amount Component
  - all the Organisation Components present, and
  - the Delivery Component,
- the following components of the TPO Block:
  - the Protocol Options Component, and
  - the Brand List Component

If the Baseline Refund is a Baseline Refund with Authentication then the Signature Component additionally contains a hash of the following:

- the Brand Selection Component contained in the TPO Selection Block.

### 5.4.9 Payment Request Block

The Payment Request Block (see section 4.6) contains:

- the following components copied from the Offer Response Block:
  - the Authentication Data Component if present
  - the Pay Amount Component
  - the Organisation Components with the roles of: Merchant and ValueAcquirer
  - the "Offer" Signature Component if present
- the following component from the TPO Block:
  - the Brand List Component
- one Brand Selection Component either:
  - copied from the Offer Response Block if the refund is a Baseline Refund with Authentication, or
  - created by the Consumer, containing the payment brand and payment protocol selected, if the refund is a Baseline Refund with Authentication
- one Payment Scheme Component (see section 3.9) if required by the payment method used (see the Payment Method supplement to determine if this is needed).

If signatures are used then the Value Acquirer should check that the Signature Component is valid (see section 3.13.7 Checking Hashes and Signatures).
5.4.10 Payment Exchange Block

The Payment Exchange Block (see section 4.7) contains:

- one Payment Scheme Component (see section 3.9) which contains payment method specific data. See the Payment Method supplement for the payment method being used to determine what this should contain.

5.4.11 Payment Response Block

The Payment Response Block (see section 4.8) contains:

- one Pay Receipt Component (see section 3.10) which contains scheme specific data which can be used to verify the payment occurred
- one Payment Scheme Component (see section 3.9) if required which contains payment method specific data. See the Payment Method supplement for the payment method being used to determine what this should contain
- the "Offer" Signature Component (see section 3.13) from the Payment Request Block if present.

If a signed pay receipt is being provided, indicated by the SignedPayReceipt attribute of the Pay Amount Component of the Offer Response Block being set to True, then the Payment Response Block additionally contains one extra "Pay Receipt" Signature Component which contains hashes of the following:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Payment Response Block,
- the Transaction Id Component (see section 2.3.1) within the Trans Ref Block that globally uniquely identifies the OTP Transaction,
- the Pay Receipt Component from the Payment Response Block and
- the "Offer" Signature Component from the Payment Request Block if present.

5.5 Baseline Withdrawal OTP Transaction

The Baseline Withdrawal OTP Transaction supports the withdrawal of electronic cash from a Financial Institution\(^\text{17}\). It consists of the following Trading Exchanges:

- an optional Authentication Exchange (see section 1.2.4),
- an Offer Exchange (see section 1.2.1), and
- a Payment Exchange (see section 1.2.2).

\(^{17}\) The Financial Institution has, in OTP terminology, a role of merchant in that a service (i.e. a withdrawal of electronic cash) is being offered in return for a fee, for example bank charges of some kind. The term "Financial Institution" is used in the diagrams and in the text for clarity.
These Trading Exchanges are implemented by a set of predefined OTP Messages (see section 2.2) which are exchanged between the Trading Roles (see section 1.1). Each OTP Message contains Trading Blocks (see section 3.15) which contain the Trading Components (see section 3) which are required by the Trading Exchanges.

The Trading Blocks used by the Baseline Purchase OTP Transaction are:

- TPO (Trading Protocol Options) Block
- TPO Selection Block
- Authentication Request Block
- Authentication Response Block
- Offer Response Block
- Payment Request Block
- Payment Exchange Block
- Payment Response Block

### 5.5.1 Baseline Withdrawal Variations

The Baseline Withdrawal OTP Transaction occurs in two basic forms:

- Baseline Withdrawal with Authentication. Where the Consumer making the withdrawal is authenticated before the withdrawal is made, and
- Baseline Withdrawal without Authentication. Where the Consumer is not authenticated before the withdrawal is made.

### 5.5.2 Baseline Withdrawal Authentication

In Baseline Withdrawal with Authentication an Authentication Exchange occurs before the Offer Exchange containing the details of the withdrawal is provided by the Financial Institution.

In Baseline Withdrawal without Authentication, there is no Authentication Exchange and the Financial Institution provides details about the withdrawal immediately at the start of the OTP Transaction.

These two alternatives are illustrated in the two diagrams below. The first diagram illustrates the case when an Authentication Exchange is included.
1. Consumer decides to withdraw electronic cash and sends information about how much to withdraw to the Financial Institution, e.g. using HTML.

2. The Financial Institution decides which payment brand and protocols to offer, generates an Authentication Request Block containing challenge data and the method of authentication and sends to the Consumer.

3. OTP aware application started. The consumer selects the payment brand and payment protocol to use, records selection in a Pay Selection Component, generates an Authentication Response Component and sends back to the Financial Institution.

4. The Financial Institution checks the Authentication Response against the challenge data in the Authentication Request Block, uses the information to identify the consumer, generates an Offer Response Block containing information about the withdrawal and sends to the Consumer.

5. Consumer checks Offer is OK, combines components from the TPO Block, the TPO Selection Block and the Offer Response Block to create a Pay Request Block and sends to the Value Acquirer.

Note that the above diagram:

- describes the general case where a Financial Institution can offer withdrawal of several different types of electronic cash. In practice usually only one form of electronic cash may be offered. However, there may be several different protocols which may be used for the same “brand” of electronic cash.

- the financial institution may use the results of the authentication to identify not only the consumer but also the account from which the withdrawal is to be made. If no single account can be identified, then it must be obtained by other means. For example:
  - the consumer could specify the account number in the initial dialogue (see step 1), or
  - the consumer could have been identified earlier, for example using a Baseline Authentication OTP Transaction, and an account selected from a list provided by the Financial Institution.
Chapter 5 Open Trading Protocol Transactions

The second diagram illustrates the case when an Authentication Exchange is not included.

The Baseline Withdrawal without Authentication might be used:

- when a previous OTP transaction, for example a Baseline Deposit or a Baseline Authentication, authenticated the consumer, and a secure channel has been maintained, therefore the authenticity of the consumer is known

- when authentication is achieved as part of a proprietary payment protocol and is therefore included in the Payment Exchange

- when authentication of the consumer has been achieved by some other means, for example, by using a pass phrase, or a proprietary banking software solution.

OTP aware applications supporting the Consumer Trading Role must check for the existence of an Authentication Request Block in the first OTP Message to determine whether the Baseline Withdrawal includes an Authentication Exchange or not.
5.5.3 Baseline Withdrawal Payment Messages

Once the Offer Response Trading Block has been received, the sequence of OTP Messages illustrated in Figure 20 occurs. These are the same whether or not an Authentication of the Consumer has occurred. Note that these continue where the previous diagrams (Figure 18 and Figure 19) finish.

The remainder of this sub-section on the Baseline Withdrawal OTP Transaction defines the contents of each Trading Block. For most Trading Blocks, the content does not alter with the variations described above. Where differences apply, these are stated.

5.5.4 TPO (Trading Protocol Options) Block

The TPO (Trading Protocol Options) Block (see section 5.3.2) must contain the following Trading Components:

- one Protocol Options Component which defines the options which apply to the whole OTP Transaction. See Section 3.1.
- one Brand List Component (see section 3.6) which contains the payment brand and protocols which may be selected for use in the Payment Exchange.
5.5.5 TPO Selection Block

The TPO Selection Block (see section 4.2) is only used by Baseline Withdrawal with Authentication. It contains:

- one Brand Selection Component (see section 3.7) for use in the Payment Exchange. It contains the results of the consumer selecting a Payment Brand and Payment Protocol from the list provided in the Brand List Component.

5.5.6 Authentication Request Block

The Authentication Request Block (see section 4.4) must contain the following Trading Component:

- one Authentication Data Component (see section 3.2)

5.5.7 Authentication Response Block

The Authentication Response Block (see section 4.5) must contain the following Trading Component:

- one Authentication Response Component (see section 3.3).

5.5.8 Offer Response Block

The Offer Response Block (see section 4.3) must contain the following components:

- zero or one Authentication Data Components (see section 3.2) An Authentication Data Component is required for each Payment Exchange, where its Pay Amount Component contains an AuthDataRef attribute.
- one Order Component (see section 3.4) which contains details about the withdrawal, for example the amount of value being withdrawn and any fees which might apply.
- one Pay Amount Component (see section 3.8) which contains information about the payment which is to be made.
- Organisation Components (see section 3.5) with the following roles:
  - the Merchant\(^\text{18}\) who is accepting the withdrawal
  - the Consumer who is making the withdrawal
  - the ValueAcquirer for the payment. The “ID” of the Value Acquirer Organisation Component is contained within the VaOrgRef attribute of the Pay Amount Component.
- one Delivery Component (see section 3.11) with the NoDelivExch attribute set to True.

\(^\text{18}\) A role of Merchant is used in that a service (i.e. a withdrawal of electronic cash) is being offered in return for a fee, for example bank charges of some kind. The term “Financial Institution” is used in the diagrams and in the text for clarity.
Chapter 5 Open Trading Protocol Transactions

If the Baseline Withdrawal Offer Response is being digitally signed then the Offer Response Block must contain a "Offer" Signature Component (see section 3.13). The Signature Component contains hashes of the following XML elements:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Offer Response Block within the OTP Transaction. It contains information that identifies the OTP Message and OTP Transaction
- the Transaction Id Component (see section 2.3.1) which globally uniquely identifies the OTP Transaction
- the following components of the Offer Response Block:
  - the Authentication Data Component if present
  - the Order Component
  - the Pay Amount Component
  - all the Organisation Component present, and
  - the Delivery Component,
- the following components of the TPO Block:
  - the Protocol Options Component, and
  - the Brand List Component

If the Baseline Withdrawal is a Baseline Withdrawal with Authentication then the Signature Component additionally contains a hash of the following:

- the Brand Selection Component contained in the TPO Selection Block.

5.5.9 Payment Request Block

The Payment Request Block (see section 4.6) contains:

- the following components copied from the Offer Response Block:
  - the Authentication Data Component if present
  - the Pay Amount Component
  - the Organisation Components with the roles of Merchant and ValueAcquirer
  - the "Offer" Signature Component if present
- the following component from the TPO Block:
  - the Brand List Component
- one Brand Selection Component either:
  - copied from the Offer Response Block if the withdrawal is a Baseline Withdrawal with Authentication, or
  - created by the Consumer, containing the payment brand and payment protocol selected, if the withdrawal is a Baseline Withdrawal without Authentication
- one Payment Scheme Component (see section 3.9) if required by the payment method used (see the Payment Method supplement to determine if this is needed).

If signatures are used then the Value Acquirer should check that the Signature Component is valid (see section 3.13.7 Checking Hashes and Signatures).
5.5.10 Payment Exchange Block

The Payment Exchange Block (see section 4.7) contains:

- one Payment Scheme Component (see section 3.9) which contains payment method specific data. See the Payment Method supplement for the payment method being used to determine what this should contain.

5.5.11 Payment Response Block

The Payment Response Block (see section 4.8) contains:

- one Pay Receipt Component (see section 3.10) which contains scheme specific data which can be used to verify the payment occurred
- one Payment Scheme Component (see section 3.9) if required which contains payment method specific data. See the Payment Method supplement for the payment method being used to determine what this should contain
- the "Offer" Signature Component (see section 3.13) from the Payment Request Block if present.

If a signed pay receipt is being provided, indicated by the SignedPayReceipt attribute of the Pay Amount Component of the Offer Response Block being set to True, then the Payment Response Block additionally contains one extra "Pay Receipt" Signature Component which contains hashes of the following:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Payment Response Block,
- the Transaction Id Component (see section 2.3.1) within the Trans Ref Block that globally uniquely identifies the OTP Transaction,
- the Pay Receipt Component from the Payment Response Block and
- the "Offer" Signature Component from the Payment Request Block if present.

5.6 Baseline Value Exchange OTP Transaction

The Baseline Value Exchange Transaction uses Payment Exchanges (see section 1.2.2) to support the exchange of value in one currency obtained using one payment method with value in the same or another currency using the same or another payment method. Examples of its use include:

- electronic cash advance on a credit card. For example the first payment could be a dollar SET Payment Exchange on a credit card with the second Payment Exchange being a download of DigiCash e-cash in dollars.
- foreign exchange using the same payment method. For example the payment could be an upload of Mondex value in French Francs and the second a download of Mondex value in British Pounds
foreign exchange using different payment methods. For example the first payment
could be a SET payment in Euros followed a download of Geld Karte in
Deutchmarks.

The Baseline Value Exchange uses three Trading Exchanges:

- an Offer Exchange (see section 1.2.1) which provides details of what values and
currencies will be exchanged
- two Payment Exchanges (see section 1.2.2) which carry out the two payments
involved

These Trading Exchanges are implemented by a set of predefined OTP Messages (see section
2.2) which are exchanged between the Trading Roles (see section 1.1). Each OTP Message
contains Trading Blocks (see section 3.15) which contain the Trading Components (see section
3) which are required by the Trading Exchanges.

The Trading Blocks used by the Baseline Value Exchange OTP Transaction are:

- TPO (Trading Protocol Options) Block
- TPO Selection Block
- Offer Response Block
- Payment Request Block
- Payment Exchange Block
- Payment Response Block

If the Baseline Value Exchange OTP Transaction fails, then in a future version of OTP there
will be a Restart OTP Transaction (see section 5.7 Restart Transaction (provisional)) which, if
possible, will allow the transaction to be restarted. This will be particularly useful if the
transaction fails between the first and second payments since it would permit the second
payment to occur.

5.6.1 Baseline Value Exchange Variations

The Baseline Value Exchange OTP Transaction occurs in two basic forms:

- Brand Dependent Value Exchange. Where the content of the offer, for example the
rate at which one form of value is exchanged for another, is dependent on the
payment brands and protocols selected by the consumer, and
- Brand Independent Value Exchange. Where the content of the offer is not
dependent on the payment brands and protocols selected.

In Brand Dependent Value Exchange the TPO Block and the Offer Response Block are sent
separately by the Merchant\(^\text{19}\) to the Consumer, i.e.:

\(^{19}\) Note that the role is a Merchant even though the organisation carrying out the Value Exchange may be a Bank or some
other Financial Institution. This is because the Bank is acting as a merchant in that they are making an offer which the
Consumer can either accept or decline.
• the Brand List Components for the two payments are sent to the Consumer in a TPO Block,

• the Consumer selects a Payment Brand and Payment Protocol from the Brand List Component for each of the payments in the Value Exchange

• the Consumer sends the selected brands and protocols back to the Merchant in a TPO Selection Block, and

• the Merchant Uses the information received to define the content of the Offer Response Block and then sends it to the Consumer.

In Brand Independent Value Exchange the TPO Block and the Offer Response Block are sent together by the Merchant to the Consumer in the same OTP Message at the start of the OTP Transaction.

These two alternatives are illustrated in the two diagrams below. The first diagram illustrates a Brand Dependent Value Exchange.

Figure 33 Brand Dependent Value Exchange
The second diagram illustrates the a Brand Independent Value Exchange.

1. Consumer decides to conduct a Value Exchange and sends information about the exchange to the Merchant, e.g. using HTML.

2. Merchant decides which payment brand and protocols to offer for each payment, places them in Pay Options Components in a TPO Block, creates an Offer Response Block containing details about the Value Exchange, e.g. exchange rates, commission, etc. and sends to Consumer.

3. OTP aware application started. Consumer selects the payment brand and payment protocol to use for each payment, records selections in two Pay Selection Components, checks Offer is OK, combines the Pay Selection Component for the first payment with information from the TPO Block and Offer Response Block to create a Pay Request Block for the first payment and sends it to Value Acquirer 1.

4. Value Acquirer 1 processes the Pay Request Block for the first payment and carries out payment protocol...

The TPO Block and Offer Response Block may only be combined into the same OTP Message if the content of the Offer Response Block does not change as a result of selecting the payment brands and payment protocols to be used in the Value Exchange.

Note that the TPO Block and the Offer Response Block may be sent in separate OTP messages even if the Offer Response Block does not change. However this increases the number of messages in the transaction and is therefore likely to increase transaction response times.

The content of the Offer Response Block may vary, for example, since:
- the exchange rate and commission to be paid may vary by payment brand.
- the value acquirer may differ depending on the payment brand selected.

OTP aware applications supporting the Consumer Trading Role must check for the existence of an Offer Response Block in the first OTP Message to determine whether the Baseline Value Exchange is brand dependent.

Whether or not the Value Exchange is brand dependent, the exchange of Trading Blocks between the Consumer and the Value Acquirers are the same. This is illustrated in the diagram below. Note that this diagram continues where the previous diagrams (Figure 33 and Figure 34) finish.
The remainder of this sub-section on the Baseline Value Exchange OTP Transaction defines the contents of each Trading Block. The content does not alter with the variations described above.

5.6.2 TPO (Trading Protocol Options) Block

The TPO (Trading Protocol Options) Block (see section 5.3.2) must contain the following Trading Components:

- one Protocol Options Component which defines the options which apply to the whole OTP Transaction. See Section 3.1.
- two Brand List Components (see section 3.6) one for each Payment Exchange where each Brand List Component contains one or more payment brands and protocols which may be selected for use in the Payment Exchange.
5.6.3 TPO Selection Block

The TPO Selection Block (see section 4.2) is only used by Brand Dependent Value Exchange. It contains:

- two Brand Selection Components (see section 3.7). One for each of the Payment Exchanges. Each Brand Selection Component contains the results of the consumer selecting a Payment Brand and Payment Protocol from the list provided in the Brand List Component.

5.6.4 Offer Response Block

The Offer Response Block (see section 4.3) contains the following components:

- zero, one or two Authentication Data Component (see section 3.2). An Authentication Data Component is required for each Payment Exchange, where its Pay Amount Component contains an AuthDataRef attribute.
- one Order Component (see section 3.4) which contains details about the value exchange, for example, exchange rates, commission, etc.
- two Pay Amount Components (see section 3.8) which contain information about each of the two payments which are to be made
- Organisation Components (see section 3.5) with the following roles:
  - Merchant who is providing the goods or services
  - Consumer who is making the purchase
  - one or two ValueAcquirers for the payments\(^\text{20}\). The “ID” of a Value Acquirer Organisation Component is contained within the VaOrgRef attribute of each of the Pay Amount Components

If the Baseline Value Exchange Offer Response is being digitally signed then the Offer Response Block must contain an “Offer” Signature Component (see section 3.13). The Signature Component contains hashes of the following XML elements:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Offer Response Block within the OTP Transaction. It contains information that identifies the OTP Message and OTP Transaction
- the Transaction Id Component (see section 2.3.1) which globally uniquely identifies the OTP Transaction
- the following components of the Offer Response Block:
  - the Authentication Data Component if present
  - the Order Component
  - the two Pay Amount Components
  - all the Organisation Components present, and
- the following components of the TPO Block:
  - the Protocol Options Component, and
  - the Brand List Component

\(^{20}\) one Value Acquirer may be used if the same Value Acquirer is being used for both payments
If the Baseline Value Exchange is a Brand Dependent Value Exchange then the Signature Component additionally contains a hash of the following:

- the two Brand Selection Components contained in the TPO Selection Block.

### 5.6.5 Payment Request Block (first payment)

The Payment Request Block (see section 4.6) for the first payment contains:

- the following components copied from the Offer Response Block:
  - the Authentication Data Component for the first payment if required
  - the Pay Amount Component for the first payment
  - the Organisation Components with the roles of *Merchant* and *ValueAcquirer* for the first payment
  - the Signature Component if present

- the following component copied from the TPO Block:
  - the Brand List Component for the first payment

- one Brand Selection Component for the first payment which is either:
  - copied from the Offer Response Block if the purchase is a Brand Dependent Value Exchange, or
  - created by the Consumer, containing the payment brand and payment protocol selected, if the purchase is a Brand Independent Value Exchange

- one Payment Scheme Component (see section 3.9) if required by the payment method used (see the Payment Method supplement to determine if this is needed).

Note that:

- the Pay Amount Component for the first payment is the one within the Offer Response Block that contains no `StartAfter` element (see section 3.8)

- the Authentication Data Component to include is identified by the `AuthDataRef` attribute of the Pay Amount Component for the first payment. If no `AuthDataRef` attribute is present then no Authentication Data Component is required

- the Value Acquirer to include is identified by the `VaOrgRef` attribute of the Pay Amount Component for the first payment

- the Brand List Component to include is the one identified by the `BrandListRef` attribute of the Pay Amount Component for the first payment

- the Brand Selection Component to include from the Offer Response Block is the one that contains an Element Reference (see section 2.5) which identifies the Brand List Component for the first payment

- if a Brand Selection Component is being created, then it must identify the Brand List Component for the first payment.
5.6.6 Payment Exchange Block (first payment)

The Payment Exchange Block (see section 4.7) for the first payment contains:

- one Payment Scheme Component (see section 3.9) which contains payment method specific data for the payment method being used by the first payment. See the Payment Method supplement for the payment method being used to determine what this should contain.

5.6.7 Payment Response Block (first payment)

The Payment Response Block for the first payment (see section 4.8) contains:

- one Pay Receipt Component (see section 3.10) which contains scheme specific data which can be used to verify the first payment occurred
- one Payment Scheme Component (see section 3.9) if required by the payment method used by the first payment which contains payment method specific data. See the Payment Method supplement for the payment method being used to determine what this should contain
- the Signature Component (see section 3.13) from the Payment Request Block for the first payment if present.

If a signed pay receipt is being provided, indicated by the SignedPayReceipt attribute of the Pay Amount Component for the first payment being set to True, then the Payment Response Block additionally contains a "Pay Receipt" Signature Component which contains hashes of the following:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Payment Response Block for the first payment,
- the Transaction Id Component (see section 2.3.1) within the Trans Ref Block that globally uniquely identifies the OTP Transaction,
- the Pay Receipt Component from the Payment Response Block for the first payment and
- the "Offer" Signature Component from the Payment Request Block for the first payment, if present.

5.6.8 Payment Request Block (second payment)

The Payment Request Block (see section 4.6) for the second payment contains:

- the following components copied from the Offer Response Block:
  - the Authentication Data Component for the second payment if required
  - the Pay Amount Component for the second payment
  - the Organisation Components with the roles of Merchant and ValueAcquirer for the second payment
  - the “Offer” Signature Component if present
- the following component copied from the TPO Block:
  - the Brand List Component for the second payment
Chapter 5 Open Trading Protocol Transactions

- one Brand Selection Component for the second payment which is either:
  - copied from the Offer Response Block if the purchase is a Brand Dependent Value Exchange, or
  - created by the Consumer, containing the payment brand and payment protocol selected, if the purchase is a Brand Independent Value Exchange
- one Payment Scheme Component (see section 3.9) if required by the payment method used (see the Payment Method supplement to determine if this is needed)
- the "Pay Receipt" Signature Component copied from the Payment Response Block for the first payment (see section 4.8)

Note that:
- the Pay Amount Component for the second payment is the one within the Offer Response Block that contains a \texttt{StartAfter} element (see section 3.8) that identifies the Pay Amount Component for the first payment
- the Authentication Data Component to include is identified by the \texttt{AuthDataRef} attribute of the Pay Amount Component for the second payment. If no \texttt{AuthDataRef} attribute is present then no Authentication Data Component is required
- the Value Acquirer to include is identified by the \texttt{VaOrgRef} attribute of the Pay Amount Component for the second payment
- the Brand List Component to include is the one identified by the \texttt{BrandListRef} attribute of the Pay Amount Component for the second payment
- the Brand Selection Component to include from the Offer Response Block is the one that contains an Element Reference (see section 2.5) which identifies the Brand List Component for the second payment
- if a Brand Selection Component is being created, then it must identify the Brand List Component for the second payment.

5.6.9 Payment Exchange Block (second payment)

The Payment Exchange Block (see section 4.7) for the second payment contains:

- one Payment Scheme Component (see section 3.9) which contains payment method specific data for the payment method being used by the second payment. See the Payment Method supplement for the payment method being used to determine what this should contain.

5.6.10 Payment Response Block (second payment)

The Payment Response Block for the second payment (see section 4.8) contains:

- one Pay Receipt Component (see section 3.10) which contains scheme specific data which can be used to verify the second payment occurred
- one Payment Scheme Component (see section 3.9) if required by the payment method used by the second payment which contains payment method specific data.
See the Payment Method supplement for the payment method being used to determine what this should contain

- all the Signature Components (see section 3.13) from the Payment Request Block for the second payment if present.

If a signed pay receipt is being provided, indicated by the SignedPayReceipt attribute of the Pay Amount Component for the second payment being set to True, then the Payment Response Block additionally contains a "Pay Receipt" Signature Component which contains hashes of the following:

- the Trans Ref Block (see section 2.3) for the OTP Message which contains the first usage of the Payment Response Block for the second payment,
- the Transaction Id Component (see section 2.3.1) within the Trans Ref Block that globally uniquely identifies the OTP Transaction,
- the Pay Receipt Component from the Payment Response Block for the second payment,
- the "Offer" Signature Component from the Payment Request Block for the second payment, if present, and
- the "Pay Receipt" Signature Component from the Payment Request Block for the second payment, if present.

5.6.11 Baseline Value Exchange Signatures

The use of signatures to ensure the integrity of a Baseline Value Exchange is illustrated by the diagram below.
5.7 Restart Transaction (provisional)

Although not included in this version of the OTP Specification. It is the intention of the authors of this specification that there will be an Restart Transaction which will facilitate the restarting of any OTP Transaction which has failed.

In outline it is anticipated that the Restart Transaction will:

- rely on the Consumer Trading Role initiating a restart
- consist of the Consumer sending information about the failed transaction to an appropriate Net Location

The information sent will consist of a Trading Transaction Record containing some or all parts of one or more OTP Messages previously sent between the Trading Roles.

The role receiving the Trading Transaction Record will identify whether and how the transaction might be restarted and then communicate this information back to the Consumer so that the consumer can continue with the transaction.

The above information is a guide only and must not be relied upon.
6. Terminating and Resending OTP Transactions

6.1 Terminating OTP Transactions

Transactions can either be terminated by:
• the Consumer, or by
• one of the other Trading Roles in the trade.

Transactions can also be terminated because:
• the OTP Message was in error in that it did not conform to this specification, or
• it was cancelled since one of the Trading Roles could not accept the transaction, for example the Consumer found the contents of the offer response unacceptable or the Value Acquirer found a signature invalid.

6.1.1 Consumer Termination of Transactions

If the Consumer is terminating a transaction then she:
• sends a Fail Trading Block (see section 4.11) to the sender of the OTP Message which caused the Fail Trading Block to be generated, then
• directs the browser to one of the Net Locations\(^\text{21}\) contained in the Protocol Options Component (see section 3.1).

When the Fail Trading Block is received by the non-consumer, then it may be recorded for later analysis.

6.1.2 Non-consumer Termination of Transactions

Two cases need to be considered:
• the first OTP Message containing a TPO Trading Block has not been sent. For example the Consumer has requested that an OTP transaction be started, for example by "clicking" on a button on a web page but the merchant has not yet responded, and
• after the first OTP Message containing a TPO Trading Block has been sent.

\(^{21}\) The content of a Net Location is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.
Chapter 6 Terminating and Resending OTP Transactions

In the first case, the consumer must be sent a message, for example in the format of a web page which explains the reason for the failure. The format of this message is not defined by OTP.

In the second case, the non-consumer sends an OTP Message to the Consumer containing the Fail Trading Block (see section 4.11). The consumer software should then display the message and terminate processing of the OTP Transaction and direct the browser to one of the Net Locations contained in the Protocol Options Component (see section 3.1).

6.2 Resending OTP Messages

Any role in an OTP Transaction which is sending an OTP Message to which a response is expected should run a timer which starts when the OTP Message is sent.

If no OTP Message in reply has been received after a period of time, then the OTP Message should be re-sent. In this case the Status attribute of the OTP Message (see section 2.1.1) should be changed from Original to Resend.

An OTP Message received which is incomprehensible or which fails security checks, including verification of its signature(s), may be logged but does not count as a correct response and should not reset the timer.

This retransmission behaviour needs to be repeated for repeated time outs up to a retransmission limit.

Appropriate defaults and policies for time out and retransmission limit depend on the transport being used for the OTP messages and is covered in Transport Method specific supplements.

In general, a Trading Role that receives an OTP Message which appears correct and meets security checks, including verification of its signature(s), needs to decide if the OTP Message contains a request for current status or contains some other type of information. Requests for current status are always answered based on the actual current status.

All other OTP Messages received must be checked to see if they are duplicates of previous OTP Messages. If they are, and the previous OTP Message is still being processed, the duplicate copy of that OTP Message should be ignored.

If the OTP Message is a duplicate that was previously responded to, then no action should be taken on the OTP Message other than that the previous OTP Message should be resent to the trading role who sent the message. This requires that some sort of local cache or database be maintained of OTP Messages received together with corresponding OTP Messages which were sent.

---

22 The content of a Net Location is dependent on the Transport Mechanism (such as HTTP) that is used. See the Transport Mechanism supplement.

23 This version of OTP does not include requests for current status. It may be included in later versions.
Chapter 6 Terminating and Resending OTP Transactions

The length of time OTP Messages received and sent should be kept in such a memory depends on the transport and payment protocols being used and is covered in the transport and payment protocol specific documents. In general, OTP Messages should be remembered for the longest of the lengths of time recommended in any of the transport and payment protocols in use by the OTP aware application.
Appendix A - XML Name Summary

In a later version of specification, this appendix will summarise each attribute name in the protocol including:

- the elements where it is used
- any limitations on attribute value size
- attribute format
- any validation rules which apply and fail codes to use if validation fails
Appendix B - Failure Codes

In a later version of this specification, this appendix will contain a list of the FailCodes, and English Language versions of FailTexts (see 3.15 Fail Reason Component).

**Note:** The authors of this specification will welcome translations of Fail Texts in other languages at the appropriate time.
Appendix C - XML Overview

This section contains an overview of [XML]. Its purpose is to provide sufficient explanation so that the XML examples in this document may be understood. This description is not complete. For more detail and the full specification see the reference contained in the Preface to this document.

XML is based on SGML has the goal of enabling "generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML."

XML is designed as a universal, open data format for the Internet and allows the structure of data in messages to be clearly and unambiguously defined.

In the following examples, underlined words are to be replaced by proper names; for example, document name could be OtpMessage, EDIMessage, and so on.

The structure of data in XML is defined using a number of key components. These are:

- document definitions
- element declarations and
- attribute declarations.

These are described below.

Document Definition

A document definition has the following form:

```xml
<!DOCTYPE DocumentName [DocumentStructureDefinition] >
```

<table>
<thead>
<tr>
<th><strong>DocumentName</strong></th>
<th>For OTP this is always OtpMessage.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DocumentStructure</strong></td>
<td>This contains the declarations of elements, attributes and entities.</td>
</tr>
</tbody>
</table>

For example:

```xml
<!DOCTYPE X [ 
  <!ELEMENT Y (Y1, Y2) >
  <!ATTLIST Y ('a' | 'b' | 'c') 'a' >
  <!ELEMENT Z (Z1 | Z2) >
  <!ATTLIST Z CDATA #REQUIRED>
]>
```
Element Declaration

An element declaration has the following form:

```xml
<!ELEMENT ElementName (ElementContents)>
```

This defines the relationships among the elements, the order of occurrences of the elements, and their number of occurrences.

Example 1

An element X consists of elements A, B, and C in that order. This would be declared as follows:

```xml
<!ELEMENT X (A, B, C) />
```

If the elements A, B, and C can appear in any order, "&" is used in place of ", ".

As XML this would be expressed as follows:

```xml
<X>
  <A> DataA </A>
  <B> DataB </B>
  <C> DataC </C>
</X>
```

In this example

- "<X>" is a start tag and "</X>" is an end tag
- "DataA", "DataB", and "DataC" is the content of the element and can consist of other elements, or character data or may even be empty.

Example 2

An element X consists of one of the elements A, B, or C. This would be declared as follows:

```xml
<!ELEMENT X ( A | B | C ) >
```

If element A is selected then this would be expressed as:

```xml
<X>
  <A> DataA </A>
</X>
```
Example 3

An element $X$ consists of element $A$ occurring zero or more times and element $B$ occurring one or more times in that order. This would be declared as follows:

```
<!ELEMENT X ( A*, B+ ) >
```

The "*" indicates zero or more, and the "+" indicates one or more.

If $A$ occurred zero times and $B$ occurred twice then this would be expressed as:

```
<X>
  B> DataB </B>
  B> DataB </B>
</X>
```

Data Types used in the element declarations

The previous examples described how one element can be defined as having "children" elements. In element declarations XML also supports data types. The data types used in this OTP specification are shown below.

<table>
<thead>
<tr>
<th>Data Types</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>#PCDATA</td>
<td>The element content contains data which the XML parser can search to look for tags or entity declarations.</td>
</tr>
<tr>
<td>ANY</td>
<td>The element content can contain any element defined in any order.</td>
</tr>
<tr>
<td>EMPTY</td>
<td>The element content contains no data.</td>
</tr>
</tbody>
</table>

Attribute declarations

Attribute declarations describe information about an element. More than one attribute can be defined for one element. Attributes are contained within the start tag of an element. They are defined as follows:

```
<!ATTLIST ElementName AttributeName1 DeclaredValue1 DefaultValue1
  AttributeName2 DeclaredValue2 DefaultValue2
  ...
  AttributeNameN DeclaredValueN DefaultValueN >
```

Declared value

When the permissible values of an attribute are known, those values are declared as a list in the declared value.
When the list of permissible values are not pre-defined, data types are specified instead. The data types which can be used for attribute declarations are listed below. Only the ones used in this OTP specification are shown.

<table>
<thead>
<tr>
<th>Attribute Types</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDATA</td>
<td>Character data. Characters other than attribute value delimiters such as (&quot;”) can be used. Characters of zero length are allowed.</td>
</tr>
<tr>
<td>NMTOKEN</td>
<td>An attribute which conforms with the rules for an XML name. In outline it must start with a letter and be followed by any combination of letters, digits, or a few special characters. No spaces are allowed.</td>
</tr>
<tr>
<td>NMTOKENS</td>
<td>One or more NMTOKEN separated by spaces.</td>
</tr>
<tr>
<td>ID</td>
<td>Identifier. This value of this attribute is unique for each element.</td>
</tr>
<tr>
<td>IDREF</td>
<td>This value of this attribute matches the value of some ID attribute of an element in the same XML document. It is used to point to that element.</td>
</tr>
<tr>
<td>IDREFS</td>
<td>One or more IDREFs separated by spaces.</td>
</tr>
</tbody>
</table>

**Default value**

Default values indicate whether or not the attribute must be present in an element.

For default values, the following default keywords as well as some concrete values can be specified. Only the default keywords used in this OTP specification are shown.

<table>
<thead>
<tr>
<th>Values</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>#REQUIRED</td>
<td>CANNOT abbreviate. Some value must be specified for this attribute.</td>
</tr>
<tr>
<td>#IMPLIED</td>
<td>When an attribute with this default value is not specified, the application gives the pre-determined attribute value.</td>
</tr>
<tr>
<td>'value'</td>
<td>The 'value specified is the default. Other values may be used.</td>
</tr>
<tr>
<td>#FIXED 'value'</td>
<td>The value must and can only be the value specified.</td>
</tr>
</tbody>
</table>

**Example**

An example of an attribute declaration follows:

```xml
<!ATTLIST X
  Att1 ( A, B ) #REQUIRED >
```

In this example a value for ATT1 must be present as it is "#REQUIRED" and it must be either "A" or "B" for the XML document to be valid. For example:

```xml
<X Att1='B'> DataX </X>
```
Appendix D - Open Trading Protocol Formal Definition

This section will contain a copy of the XML DTD for the Open Trading Protocols for information purposes.

The master copy of the DTD for OTP, which should be relied upon is available for download from the OTP web site (http://www.otp.org).

Note: As this document is a draft for public comment, the OTP DTD has not yet been completed. Therefore in the interim, the DTD definitions contained within the body of the OTP specification should be regarded as the master versions.