



AS4 Profile of ebMS V3 Version 1.0

Committee Specification 01

24 April 2010

Specification URIs:

This Version:

<http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/200707/AS4-profile-cs-01.pdf>

(Authoritative)

<http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/200707/AS4-profile-cs-01.html>

<http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/200707/AS4-profile-cs-01.odt>

Previous Version:

<http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/200707/AS4-profile-cd-02.pdf>

(Authoritative)

<http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/200707/AS4-profile-cd-02.html>

<http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/200707/AS4-profile-cd-02.odt>

Latest Version:

<http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/200707/AS4-profile.pdf>

<http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/200707/AS4-profile.html>

<http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/200707/AS4-profile.odt>

Technical Committee:

[OASIS ebXML Messaging Services TC](#)

Chair:

Ian Jones, British Telecommunications plc <ian.c.jones@bt.com>

Editor:

Jacques Durand, Fujitsu Computer Systems <jdurand@us.fujitsu.com>

Related Work:

This specification is related to:

- [OASIS ebXML Messaging Services Version 3.0: Part 1, Core Specification](#)

Declared XML Namespace:

<http://docs.oasis-open.org/ebxml-msg/ns/ebms/v3.0/profiles/200707>

Abstract:

While ebMS 3.0 represents a leap forward in reducing the complexity of Web Services B2B messaging, the specification still contains numerous options and comprehensive alternatives for

35 addressing a variety of scenarios for exchanging data over a Web Services platform. The AS4
36 profile of the ebMS 3.0 specification has been developed in order to bring continuity to the
37 principles and simplicity that made AS2 successful, while adding better compliance to Web
38 services standards, and features such as message pulling capability and a built-in Receipt
39 mechanism. Using ebMS 3.0 as a base, a subset of functionality is defined along with
40 implementation guidelines adopted based on the "just-enough" design principles and AS2
41 functional requirements to trim down ebMS 3.0 into a more simplified and AS2-like specification
42 for Web Services B2B messaging. This document defines the AS4 profile as a combination of a
43 conformance profile that concerns an implementation capability, and of a usage profile that
44 concerns how to use this implementation. A couple of variants are defined for the AS4
45 conformance profile - the AS4 ebHandler profile and the AS4 Light Client profile - that reflect
46 different endpoint capabilities.

47 **Status:**

48 This document was last revised or approved by the ebXML Messaging Services Committee on
49 the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest
50 Approved Version" location noted above for possible later revisions of this document.

51 Technical Committee members should send comments on this specification to the Technical
52 Committee's email list. Others should send comments to the Technical Committee by using the
53 "Send A Comment" button on the Technical Committee's web page at
54 <http://www.oasis-open.org/committees/ebxml-msg/>

55 For information on whether any patents have been disclosed that may be essential to
56 implementing this specification, and any offers of patent licensing terms, please refer to the
57 Intellectual Property Rights section of the Technical Committee web page at
58 <http://www.oasis-open.org/committees/ebxml-msg/ipr.php>

59 The non-normative errata page for this specification is located at
60 <http://www.oasis-open.org/committees/ebxml-msg/>

61 Notices

62 Copyright © OASIS® 2010. All Rights Reserved.

63 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
64 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

65 This document and translations of it may be copied and furnished to others, and derivative works that
66 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
67 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
68 and this section are included on all such copies and derivative works. However, this document itself may
69 not be modified in any way, including by removing the copyright notice or references to OASIS, except as
70 needed for the purpose of developing any document or deliverable produced by an OASIS Technical
71 Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be
72 followed) or as required to translate it into languages other than English.

73 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
74 or assigns.

75 This document and the information contained herein is provided on an "AS IS" basis and OASIS
76 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
77 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
78 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
79 PARTICULAR PURPOSE.

80 OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
81 necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard,
82 to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to
83 such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that
84 produced this specification.

85 OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of
86 any patent claims that would necessarily be infringed by implementations of this specification by a patent
87 holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR
88 Mode of the OASIS Technical Committee that produced this specification. OASIS may include such
89 claims on its website, but disclaims any obligation to do so.

90 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
91 might be claimed to pertain to the implementation or use of the technology described in this document or
92 the extent to which any license under such rights might or might not be available; neither does it
93 represent that it has made any effort to identify any such rights. Information on OASIS' procedures with
94 respect to rights in any document or deliverable produced by an OASIS Technical Committee can be
95 found on the OASIS website. Copies of claims of rights made available for publication and any
96 assurances of licenses to be made available, or the result of an attempt made to obtain a general license
97 or permission for the use of such proprietary rights by implementers or users of this OASIS Committee
98 Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no
99 representation that any information or list of intellectual property rights will at any time be complete, or
100 that any claims in such list are, in fact, Essential Claims.

101 The names "OASIS", ebXML, ebXML Messaging Services, ebMS are trademarks of [OASIS](http://www.oasis-open.org), the owner
102 and developer of this specification, and should be used only to refer to the organization and its official
103 outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving
104 the right to enforce its marks against misleading uses. Please see
105 <http://www.oasis-open.org/who/trademark.php> for above guidance.

106

Table of Contents

107	1 Introduction.....	5
108	1.1 Terminology.....	6
109	1.2 Normative References.....	6
110	1.3 Non-normative References.....	7
111	2 AS4 Conformance Profiles for ebMS V3.....	8
112	2.1 The AS4 ebHandler Conformance Profile.....	8
113	2.1.1 Features Set.....	8
114	2.1.2 WS-I Conformance Profiles.....	11
115	2.1.3 Processing Mode Parameters.....	11
116	2.2 The AS4 Light Client Conformance Profile.....	13
117	2.2.1 Feature Set.....	14
118	2.2.2 WS-I Conformance Requirements.....	15
119	2.3 Conformance Profiles Compatibility.....	16
120	3 AS4 Additional Features.....	17
121	3.1 Compression.....	17
122	3.2 Reception Awareness features and Duplicate Detection.....	18
123	3.3 Alternative Pull Authorization.....	19
124	3.4 Semantics of Receipt in AS4.....	20
125	4 AS4 Usage Profile of ebMS 3.0.....	21
126	4.1 AS4 Usage Rules.....	21
127	4.1.1 Core Components / Modules to be Used.....	21
128	4.1.2 Bundling rules.....	22
129	4.1.3 Security Element.....	23
130	4.1.4 Signing Messages.....	23
131	4.1.5 Signing SOAP with Attachments Messages.....	23
132	4.1.6 Encrypting Messages.....	24
133	4.1.7 Encrypting SOAP with Attachments Messages.....	24
134	4.1.8 Generating Receipts.....	25
135	4.1.9 MIME Header and Filename information.....	26
136	4.2 AS4 Usage Agreements.....	26
137	4.2.1 Controlling Content and Sending of Receipts.....	26
138	4.2.2 Error Handling Options.....	27
139	4.2.3 Securing the PullRequest.....	28
140	4.2.4 Reception Awareness Parameters.....	29
141	4.2.5 Default Values of Some PMode Parameters.....	30
142	4.2.6 HTTP Confidentiality and Security.....	31
143	4.2.7 Deployment and Processing requirements for CPAs.....	32
144	4.2.8 Message Payload and Flow Profile.....	32
145	4.2.9 Additional Deployment or Operational Requirements.....	33
146	5 Conformance Clauses.....	34
147	5.1 AS4 ebHandler Conformance Clause.....	34
148	5.2 AS4 Light Client Conformance Clause.....	34
149	Appendix B Acknowledgments.....	38
150	Appendix C Revision History.....	39
151		

1 Introduction

152

153 Historically, the platform for mission-critical business-to-business (B2B) transactions have steadily moved
154 from proprietary networks (VANs) to Internet-based protocols free from the data transfer fees imposed by
155 the VAN operators. This trend has been accelerated by lower costs and product ownership, a maturing of
156 technology, internationalization, widespread interoperability, and marketplace momentum. The exchange
157 of EDI business documents over the Internet has substantially increased along with a growing presence
158 of XML and other document types such as binary and text files.

159 The Internet messaging services standards that have emerged provide a variety of options for end users
160 to consider when deciding which standard to adopt. These include pre-Internet protocols, the EDIINT
161 series of AS1/AS2/AS3, simple XML over HTTP, government specific frameworks, ebMS 2.0, and Web
162 Services variants. As Internet messaging services standards have matured, new standards are emerging
163 that leverages prior B2B messaging services knowledge for applicability to Web Services messaging.

164 The emergence of the ebMS 3.0 specification represents a leap forward in Web Services B2B messaging
165 services by meeting the challenge of composing many Web Services standards into a single
166 comprehensive specification for defining the secure and reliable exchange of documents using Web
167 Services. ebMS 3.0 composes the fundamental Web Services standards like SOAP 1.1/1.2, SOAP with
168 Attachments and MTOM, WS-Security 1.0/1.1, and WS-Reliability 1.1/WS-ReliableMessaging 1.1
169 together with guidance for the packaging of messages and receipts along with definitions of messaging
170 choreographies for orchestrating document exchanges.

171 Like AS2, ebMS 3.0 brings together many existing standards that govern the packaging, security, and
172 transport of electronic data under the umbrella of a single specification document. While ebMS 3.0
173 represents a leap forward in reducing the complexity of Web Services B2B messaging, the specification
174 still contains numerous options and comprehensive alternatives for addressing a variety of scenarios for
175 exchanging data over a Web Services platform.

176 In order to fully take advantage of the AS2 success story, this profile of the ebMS 3.0 specification has
177 been developed. Using ebMS 3.0 as a base, a subset of functionality has been defined along with
178 implementation guidelines adopted based on the “just-enough” design principles and AS2 functional
179 requirements to trim down ebMS 3.0 into a more simplified and AS2-like specification for Web Services
180 B2B messaging. The main benefits of AS4 compared to its previous version are compatibility with Web
181 services standards, message pulling capability, and a built-in Receipt mechanism.

182 Profiling ebMS V3 means:

- 183 ● defining of a subset of ebMS V3 options to be supported by the AS4 handler,
- 184 ● deciding which types of message exchanges must be supported, and how these exchanges
185 should be conducted (level of security, binding to HTTP, etc.)
- 186 ● deciding of AS4-specific message contents and practices (how to make use of the ebMS
187 message header fields, in an AS4 context).
- 188 ● deciding of some operational best practices, for the end-user.

189 The overall goal of a profile for a standard is to ensure interoperability by:

- 190 ● Establishing particular usage and practices of the standard within a community of users,
- 191 ● Defining the subset of features in this standard that needs to be supported by an implementation.

192 Two kinds of profiles are usually to be considered when profiling an existing standard:

- 193 1. **Conformance Profiles.** These define the different ways a product can conform to a standard,
194 based on specific ways to use this standard. A conformance profile is usually associated with a

195 specific conformance clause. Conformance profiles are of prime interest for product managers
196 and developers: they define a precise subset of features to be supported.

197 2. **Usage Profiles** (also called Deployment Profiles). These define how a standard should be used
198 by a community of users, in order to ensure best compatibility with business practices and
199 interoperability. Usage profiles are of prime interest for IT end-users: they define how to configure
200 the use of a standard (and related product) as well as how to bind this standard to business
201 applications. A usage profile usually points at required or compatible conformance profile(s).

202 AS4 is defined as a combination of:

- 203 ● A couple of AS4 Conformance Profiles (see section 2), that define the subset of ebMS V3
204 features to be supported by an AS4 implementation.
- 205 ● An AS4 Usage Profile (section 4) that defines how to use an AS4-compliant implementation in
206 order to achieve similar functions as specified in AS2.

207 Two AS4 conformance profiles (CP) are defined below:

208 (1) **the AS4 ebHandler CP**. This conformance profile supports both Sending and Receiving roles,
209 and for each role both message pushing and message pulling.

210 (2) **the AS4 light Client CP**. This conformance profile supports both Sending and Receiving roles,
211 but only message pushing for Sending and message pulling for Receiving. In other words, it does not
212 support incoming HTTP requests, and may have no IP address.

213 Compatible existing conformance profiles for ebMS V3 are:

- 214 ● Gateway RM V3 or Gateway RX V3: an MSH product implementing any of these profiles will also
215 be conforming to the AS4 ebHandler CP (the reverse is not true).

216 NOTE: Full compliance to AS4 actually requires and/or authorizes a message handler to implement a few
217 additional features beyond the above CPs. These features are described in Section 3.

218 1.1 Terminology

219 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
220 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as
221 described in IETF RFC 2119.

222 1.2 Normative References

- 223 **[ebMS2]** OASIS Standard, *OASIS ebXML Message Service Specification Version 2.0*,
224 April 1, 2002. [http://www.oasis-open.org/committees/ebxml-
msg/documents/ebMS_v2_0.pdf](http://www.oasis-open.org/committees/ebxml-
225 msg/documents/ebMS_v2_0.pdf)
- 226 **[ebMS3]** OASIS Standard, *OASIS ebXML Messaging Services, Version 3.0: Part 1, Core
227 Features*, 2007. [http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/core/ebms_core-3.0-spec.pdf](http://docs.oasis-open.org/ebxml-
228 msg/ebms/v3.0/core/ebms_core-3.0-spec.pdf)
- 229 **[ebMS3-CP]** OASIS Committee Draft 03*OASIS ebXML Messaging Services, Version 3.0:
230 Conformance Profiles*, 2008. [http://www.oasis-
open.org/committees/document.php?document_id=29854](http://www.oasis-
231 open.org/committees/document.php?document_id=29854)
- 232 **[GZIP]** *GNU Gzip Manual, Free Software Foundation*, 2006.
233 <http://www.gnu.org/software/gzip/manual/index.html>
- 234 **[RFC2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
235 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>

236 **[RFC2045]** *N Freed, et al, Multipurpose Internet Mail Extensions (MIME) Part One: Format*
237 *of Internet Message Bodies, 1996.* <http://www.ietf.org/rfc/rfc2119.txt>

238 **[SOAPATTACH]** J. Barton, et al, *SOAP Messages with Attachments*, 2000
239 <http://www.w3.org/TR/SOAP-attachments>

240 **[WSIAP10]** *WS-I Attachment Profile V1.0*, Web-Services Interoperability Consortium, 2007.
241 <http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicprofile>

242 **[WSIBP20]** *WS-I Basic Profile V2.0 (draft)*, Web-Services Interoperability Consortium, 2009.
243 <http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicprofile>

244 **[WSIBSP11]** Abbie Barbir, et al, eds, *Basic Security Profile Version 1.1*, Web-Services
245 Interoperability Consortium, 2006.
246 <http://www.wsi.org/Profiles/BasicSecurityProfile-1.1.html>

247 **[ebBP-SIG]** OASIS ebXML Business Process TC, *ebXML Business Signals Schema*,
248 2006.<<http://docs.oasis-open.org/ebxml-bp/ebbp-signals-2.0>>

249 **[WSS11]** Anthony Nadalin, et al, eds., *Web Services Security: SOAP Message Security*
250 *1.1*, 2005.<http://docs.oasis-open.org/wss/v1.1/>

251 **1.3 Non-normative References**

252

253 **[IIC-DP]** OASIS Committee Draft 01, *ebXML Deployment Profiles Templates*, 2006.
254 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebcore

255 **[ebCPPA]** OASIS, *Collaboration-Protocol Profile and Agreement Specification Version 2.0*,
256 http://www.oasis-open.org/committees/ebxml-cppa/documents/ebCPP-2_0.pdf,
257 September 23, 2002.

258 **[ebDGT]** OASIS, *ebXML Deployment Guide Template Specification Version 1.0* (ebXML
259 IIC) [http://www.oasis-](http://www.oasis-open.org/committees/download.php/1713/ebMS_Deployment_Guide_Template_10.doc)
260 [open.org/committees/download.php/1713/ebMS_Deployment_Guide_Template_](http://www.oasis-open.org/committees/download.php/1713/ebMS_Deployment_Guide_Template_10.doc)
261 [10.doc](http://www.oasis-open.org/committees/download.php/1713/ebMS_Deployment_Guide_Template_10.doc), April 7, 2003.

262 **[BPSS]** ebXML, *ebXML Business Process Specification Schema Version 1.0.1*,
263 <http://www.ebxml.org/specs/ebBPSS.pdf>, May 11, 2001.

2 AS4 Conformance Profiles for ebMS V3

264

265

266 **NOTE:** AS4 is more than a Conformance Profile, in the sense given in [ebMS3-CP]. It is a combination of
 267 a Conformance Profile and of an Usage Profile, as explained in the introduction section. Consequently,
 268 only this section (section 2) is conforming to the format recommended in [ebMS3-CP] for describing
 269 conformance profiles. The usage profile part (section 4) is following a format based on tables similar to
 270 those found in [IIC-DP].

2.1 The AS4 ebHandler Conformance Profile

272 The AS4 ebHandler is identified by the URI:

273 <http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/cprofiles/200809/as4ebhandler>

2.1.1 Features Set

275 AS4 CP is defined as follows, using the table template and terminology provided in Appendix F
 276 (“Conformance”) of the core ebXML Messaging Services V3.0 specification [ebMS3].

277

Conformance Profile: AS4 ebHandler	Profile summary: <“Sending+Receiving” / “AS4 eb Handler” / Level 1 / HTTP1.1 + SOAP 1.2 + WSS1.1 >
Functional Aspects	Profile Feature Set
ebMS MEP	<p>Both Sender and Receiver MUST support all ebMS simple MEPs :</p> <ul style="list-style-type: none"> ● One-way / Push, ● One-way / Pull, <p>Regardless of which MEP is used, the sending of an eb:Receipt message MUST be supported:</p> <ul style="list-style-type: none"> ● For the One-way / Push, both “response” and “callback” reply patterns MUST be supported. ● For the One-way / Pull, the “callback” pattern is the only viable option, and the User message sender MUST be ready to accept an eb:Receipt either piggybacked on (or bundled with) a PullRequest, or piggybacked on another User Message, or sent separately. In all MEPs, the User message receiver MUST be able to send an eb:Receipt as a separate message (i.e. not piggybacked on a PullRequest message or on another User message). An MSH conforming to this profile is therefore NOT required to bundle an eb:Receipt with any other ebMS header or message body. <p>Use of the ebbpsig:NonRepudiationInformation element (as defined in [ebBP-SIG]) MUST be supported as content for the eb:Receipt message, i.e. when conforming to this profile a Sending MSH must be able to create a Receipt with such a content, and a Receiving MSH must be able to process it.</p>

Reliability	<p>Reception Awareness, defined as the ability for a Sending ebHandler to notify its application (message Producer) of lack of reception of an eb:Receipt related to a sent message, MUST be supported. This implies support for: (a) correlating eb:Receipts with previously sent User messages, based on the ebMS message ID, (b) detection of a missing eb:Receipt for a sent message, (c) ability to report an error to the message Producer in case no eb:Receipt has been received for a sent message.</p> <p>The semantics of sending back an eb:Receipt message is: a well-formed ebMS user message has been received and the MSH is taking responsibility for its processing, (no additional application-level delivery semantics, and no payload validation semantics).</p> <p>No support for a WS reliable messaging specification is required although that is an option.</p>
Security	<p>The following security features MUST be supported:</p> <ul style="list-style-type: none"> ● Support for username / password token, digital signatures and encryption. ● Support for content-only transforms. ● Support for security of attachments. ● Support for message authorization at P-Mode level (see 7.10 in [ebMS3]) Authorization of the Pull signal - for a particular MPC - must be supported at minimum. <p>Two authorization options MUST be supported by an MSH in the Receiving role, and at least one of them in the Sending role:</p> <ul style="list-style-type: none"> ● Authorization Option 1: Use of the WSS security header targeted to the “ebms” actor, as specified in section 7.10 of ebMS V3, with the wsse:UsernameToken profile. This header may either come in addition to the regular wsse security header (XMLDsig for authentication), or may be the sole wsse header, if a transport-level secure protocol such as SSL or TLS is used. An example of message is given in Appendix ... ● Authorization Option 2: Use of a regular wsse security header (XMLDsig for authentication, use of X509), and no additional wsse security header targeted to “ebms”, In that case, the MSH must be able to use the credential present in this security header for Pull authorization, i.e. to associate these with a specific MPC. <p>NOTE on XMLDsig: XMLDsig allows arbitrary XSLT Transformations when constructing the plaintext over which a signature or reference is created. Conforming applications that allow use of XSLT transformations when verifying either signatures or references are encouraged to maintain lists of “safe” transformations for a given partner, service, action and role combination. Static analysis of XSLT expressions with a human user audit is encouraged for trusting a given expression as “safe” .</p>
Error generation and reporting	<p>The following error processing capabilities MUST be supported:</p>

	<ul style="list-style-type: none"> ● Capability of the Receiving MSH to report errors from message processing, either as ebMS error messages or as Faults to the Sending MSH. The following modes of reporting to Sending MSH are supported: (a) sending error as a separate request (ErrorHandling.Report.ReceiverErrorsTo=<URL of Sending MSH>), (b) sending error on the back channel of underlying protocol (ErrorHandling.Report.AsResponse="true"). ● Capability to report to a third-party address (ErrorHandling.Report.ReceiverErrorsTo=<other address>). ● Capability of Sending MSH to report generated errors as notifications to the message producer (support for Report.ProcessErrorNotifyProducer="true")(e.g. delivery failure). ● Generated errors: All specified errors in [ebMS3] are to be generated when applicable, except for EBMS:0010: On Receiving MSH, no requirement to generate error EBMS:0010 for discrepancies between message header and the following P-Mode features: P-Mode.reliability and P-Mode.security, but requirement to generate such error for other discrepancies
Message Partition Channels	<p>Message partition channels (MPC) MUST be supported in addition to the default channel, so that selective pulling by a partner MSH is possible. This means AS4 handlers MUST be able to use the @mpc attribute and to process it as expected.</p>
Message packaging	<p>The following features MUST be supported both on sending and receiving sides:</p> <ul style="list-style-type: none"> ● Support for attachments. ● Support for MessageProperties. ● Support for processing messages that contain both a signal message unit (eb:SignalMessage) and a user message unit (eb:UserMessage).
Interoperability Parameters	<p>The following interoperability parameters values MUST be supported for this conformance profile:</p> <p>Transport: HTTP 1.1</p> <p>SOAP version: 1.2</p> <p>Reliability Specification: none.</p> <p>Security Specification: WSS 1.1. When using the One-way / Pull MEP, the response message must use by default the same WSS version as the request message. Otherwise, the version to be applied to a message is specified in the P-Mode.security</p>

278

279

280 2.1.2 WS-I Conformance Profiles

281 The Web-Services Interoperability consortium has defined guidelines for interoperability of
282 SOAP messaging implementations. In order to ensure maximal interoperability across
283 different SOAP stacks, MIME and HTTP implementations, **compliance with the following WS-I**
284 **profiles is REQUIRED whenever related features are used:**

- 285 ● Basic Security Profile (BSP) 1.1 [WSIBSP11]
- 286 ● Attachment Profile (AP) 1.0, [WSIAP10] with regard to the use of MIME and SwA.

287 Notes:

- 288 ● Compliance with AP1.0 would normally require compliance with BP1.1, which in turn requires the
289 absence of SOAP Envelope in the HTTP response of a One-Way (R2714). However, recent BP
290 versions such as BP1.2 [WSIBP12] override this requirement. Consequently, the AS4 ebHandler
291 conformance profile does not require conformance to these deprecated requirements inherited
292 from BP1.1 (R2714, R1143) regarding the use of HTTP.
- 293 ● The above WS-I profiles must be complied with within the scope of features exhibited by the AS4
294 ebHandler conformance profile. For example, since only SOAP 1.2 is required by AS4 ebHandler,
295 the requirements from BSP 1.1 that depend on SOAP 1.1 would not apply. Similarly, none of the
296 requirements for DESCRIPTION (WSDL) or REGDATA (UDDI) apply here, as these are not used.

297 This conformance profile may be refined in a future version to require conformance to the following WS-I
298 profiles, once approved and published by WS-I:

- 299 ● Basic Profile 2.0 (BP2.0)

300

301 2.1.3 Processing Mode Parameters

302 **This section contains a summary of P-Mode parameters relevant to AS4 features for this conformance**
303 **profile. An AS4 handler MUST support and understand those that are mentioned as "required". For each**
304 **parameter, either:**

- 305 – full support is required: an implementation is supposed to support the possible options for this
306 parameter.
- 307 – Support for a subset of values is required.
- 308 – No support is required: an implementation is not required to support the features controlled by this
309 parameter, and therefore not required to understand this parameter.

310

311 0. General PMode parameters:

- 312 ● **(PMode.ID:** support not required)
- 313 ● **(PMode.Agreement:** support not required)
- 314 ● **PMode.MEP:** support **required** for: [http://www.oasis-open.org/committees/ebxml-](http://www.oasis-open.org/committees/ebxml-msg/one-way)
315 [msg/one-way](http://www.oasis-open.org/committees/ebxml-msg/one-way)
- 316 ● **PMode.MEPbinding:** support **required** for: [http://www.oasis-](http://www.oasis-open.org/committees/ebxml-msg/{push,pull})
317 [open.org/committees/ebxml-msg/{ push, pull}](http://www.oasis-open.org/committees/ebxml-msg/{push,pull})

- 318 ● **PMode.Initiator.Party**: support required.
- 319 ● **PMode.Initiator.Role**: support required.
- 320 ● **PMode.Initiator.Authorization.username** and
- 321 **PMode.Initiator.Authorization.password**: support **required** for:
- 322 **wsse:UsernameToken**.
- 323 ● **PMode.Responder.Party**: support required.
- 324 ● **PMode.Responder.Role**: support required.
- 325 ● **PMode.Responder.Authorization.username** and
- 326 **PMode.Responder.Authorization.password**: **support required for:**
- 327 **wsse:UsernameToken**.
- 328

329 **1. PMode[1].Protocol:**

- 330 ● **PMode[1].Protocol.Address**: support **required** for "http" scheme.
- 331 ● **PMode[1].Protocol.SOAPVersion**: support **required** for SOAP 1.2.
- 332

333 **2.PMode[1].BusinessInfo:**

- 334 ● **PMode[1].BusinessInfo.Service**: support required.
- 335 ● **PMode[1].BusinessInfo.Action**: support required.
- 336 ● **PMode[1].BusinessInfo.Properties[]**: **support required**.
- 337 ● **(PMode[1].BusinessInfo.PayloadProfile[]: support not required)**
- 338 ● **(PMode[1].BusinessInfo.PayloadProfile.maxSize: support not required)**
- 339

340 **3. PMode[1].ErrorHandling:**

- 341 ● **(PMode[1].ErrorHandling.Report.SenderErrorsTo**: support not required)
- 342 ● **PMode[1].ErrorHandling.Report.ReceiverErrorsTo**: support required (for address of
- 343 the MSH sending the message in error or for third-party).
- 344 ● **PMode[1].ErrorHandling.Report.AsResponse**: support required (true/false).
- 345 ● **(PMode[1].ErrorHandling.Report.ProcessErrorNotifyConsumer** support not required)
- 346 ● **PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer**: support required
- 347 (true/false)
- 348 ● **PMode[1].ErrorHandling.Report.DeliveryFailuresNotifyProducer**: support required
- 349 (true/false)
- 350

351 **4. PMode[1].Reliability:**

352 none.

353

354 **5. PMode[1].Security:**

- 355 ● **PMode[1].Security.WSSVersion:** support required for: {1.1 }
- 356 ● **PMode[1].Security.X509.Sign:** support required.
- 357 ● **PMode[1].Security.X509.Signature.Certificate:** support required.
- 358 ● **PMode[1].Security.X509.Signature.HashFunction:** support required.
- 359 ● **PMode[1].Security.X509.Signature.Algorithm:** support required.
- 360 ● **PMode[1].Security.X509.Encryption.Encrypt:** support required.
- 361 ● **PMode[1].Security.X509.Encryption.Certificate:** support required.
- 362 ● **PMode[1].Security.X509.Encryption.Algorithm:** support required.
- 363 ● **(PMode[1].Security.X509.Encryption.MinimumStrength:** support not required)
- 364 ● **PMode[1].Security.UsernameToken.username:** support required.
- 365 ● **PMode[1].Security.UsernameToken.password:** support required.
- 366 ● **PMode[1].Security.UsernameToken.Digest:** support required (true/false)
- 367 ● **(PMode[1].Security.UsernameToken.Nonce:** support not required)
- 368 ● **PMode[1].Security.UsernameToken.Created:** support required.
- 369 ● **PMode[1].Security.PModeAuthorize:** support required (true/false)
- 370 ● **PMode[1].Security.SendReceipt:** support required (true/false)
- 371 ● **Pmode[1].Security.SendReceipt.ReplyPattern:** support required (both “response” and “callback”))

372 **2.2 The AS4 Light Client Conformance Profile**

373 The AS4 light Client is identified by the URI:

374 <http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/cprofiles/200809/as4lightclient>

375 **2.2.1 Feature Set**

376

<p>Conformance Profile:</p> <p>AS4-LightClient</p>	<p>Profile summary: <“Sending+Receiving” / “ lighthandler-rm” / Level 1 / HTTP1.1 + SOAP 1.1></p>
<p>Functional Aspects</p>	<p>Profile Feature Set</p>
<p>ebMS MEP</p>	<p>The following MEPs MUST be supported: One-way / Push (as initiator), and One-way / Pull (as initiator).</p> <p>Regardless of which MEP is used, the sending of an eb:Receipt message MUST be supported:</p> <ul style="list-style-type: none"> ● For the One-way / Push, the “response” reply pattern MUST be supported. ● For the One-way / Pull, the “callback” pattern is the only viable option, and the User message sender MUST be ready to accept an eb:Receipt either piggybacked on a PullRequest, or sent separately. The User message receiver MUST be able to send an eb:Receipt separately from the PullRequest. <p>In all MEPs, the User message receiver MUST be able to send an eb:Receipt as a separate message (i.e. not piggybacked on a PullRequest message or on another User message). An MSH conforming to this profile is therefore NOT REQUIRED to bundle an eb:Receipt with any other ebMS header or message body. However, when receiving a Receipt, an MSH conforming to this profile MUST be able to process an eb:Receipt bundled with an other ebMS message header or body.</p> <p>Use of the ebbpsig:NonRepudiationInformation element (as defined in [ebBP-SIG]) MUST be supported as content for the eb:Receipt message, i.e. when conforming to this profile a Sending MSH must be able to create a Receipt with such a content, and a Receiving MSH must be able to process it. .</p>
<p>Reliability</p>	<p>Reception Awareness, defined as the ability for a Sending light Client to notify its application (message Producer) of lack of reception of an eb:Receipt related to a sent message, MUST be supported. This implies support for:</p> <p>(a) correlating eb:Receipts with previously sent User messages, based on the ebMS message ID,</p> <p>(b) detection of a missing eb:Receipt for a sent message,</p> <p>(c) ability to report an error to the message Producer in case no eb:Receipt has been received for a sent message.</p> <p>The semantics of sending back an eb:Receipt message is: a well-formed ebMS user message has been received and the MSH is taking responsibility for its processing, (no additional application-level delivery semantics, and no payload validation semantics).</p> <p>Support for a WS reliable messaging specification is NOT REQUIRED although that is an option.</p>

Security	<p>Both authorization options for message pulling (authorizing PullRequest for a particular MPC) described in the ebHandler conformance profile MUST be supported:</p> <ol style="list-style-type: none"> 1. Support for username / password token: minimal support for wss:UsernameToken profile in the Pull signal - for authorizing a particular MPC. Support for adding a WSS security header targeted to the “ebms” actor, as specified in section 7.10 of ebMS V3, with the wsse:UsernameToken profile. The use of transport-level secure protocol such as SSL or TLS is recommended. 2. Support for a regular wsse security header (XMLDsig for authentication, use of X509), and no additional wsse security header targeted to “ebms”,
Error generation and reporting	<p>Error notification to the local message producer MUST be supported (e.g. reported failure to deliver pushed messages).</p> <p>The reporting of message processing errors for pulled messages to the remote party MUST be supported via Error messages (such an error may be bundled with another pushed message or a Pull signal.).</p>
Message Partition Channels	<p>Sending on the default message partition channel is sufficient (support for additional message partitions is NOT REQUIRED.)</p>
Message packaging	<p>Support for attachments is NOT REQUIRED – i.e. the message payload will use the SOAP body -, Support for MessageProperties is NOT REQUIRED.</p>
Interoperability Parameters	<p>The following interoperability parameters values MUST be supported for this conformance profile:</p> <p>Transport: HTTP 1.1</p> <p>SOAP version: 1.2</p> <p>Reliability Specification: none.</p> <p>Security Specification: WSS 1.1.</p>

377

378 2.2.2 WS-I Conformance Requirements

379

380 This conformance profile will require compliance with the following WS-I profile, once formally approved
381 by WS-I (currently in Board approval draft status):

- 382 • Basic Profile 2.0 [WSIBP20]

383 Note: the above WS-I profile **MUST** be complied with within the scope of features exhibited by the AS4
384 Light Client ebMS conformance profile.

385 **2.3 Conformance Profiles Compatibility**

386 The AS4 profile is compatible with the following ebMS V3 conformance profiles, defined in [ebMS3-CP]:

- 387 • Gateway RM V2/3
- 388 • Gateway RM V3
- 389 • Gateway RX V2/3
- 390 • Gateway RX V3

391 AS4 may be deployed on any MSH that conforms to one of the above conformance profiles.

392

3 AS4 Additional Features

393

394

395 This section defines features that were not specified in ebMS V3 and therefore out of scope for the
396 previous conformance profiles (ebHandler CP and Light Client CP). These features should be considered
397 as additional capabilities that are either required by or made optional to AS4 implementations **as indicated**
398 **below**.

399 The profiling tables below can be used for adding user-defined profiling requirements to be adopted within
400 a business community. Whenever the feature – or its profiling - is mandatory, the right-side column
401 (Profile Requirement) will specify it.

402

3.1 Compression

403

404

405 Application payloads that are built in conformance with the SOAP Messages with Attachments
406 [SOAPATTACH] specification may be compressed. Support for compression MUST then be provided by
407 AS4 implementations. Compression of the SOAP envelope and/or payload containers within the SOAP
408 Body of an ebMS Message is not supported.

409

410 To compress the payload(s) of a message build in conformance with the SOAP Messages with
411 Attachments [SOAPATTACH] specification the GZIP [GZIP] compression algorithm MUST be used.
412 Compression MUST be applied before payloads are attached to the SOAP Message.

413 The eb:PartInfo element in the message header that relates to the compressed message part, MUST
414 have an eb:Property element with @name =“Compressed”:

415 `<eb:Property name="Compressed"/>`

416 The content type of the compressed attachment MUST be "application/gzip".

417 These are indicators to the receiver that compression has been used on this part.

418

419 When compression, signature and encryption are required of the MSH, the message MUST be
420 compressed prior to being signed and/or encrypted.

421

422 Packaging requirements:

423 ● A eb:PartInfo/eb:PartProperties/eb:Property/@name="MimeType" value is RECOMMENDED to
424 identify the mimetype of the payload before compression was applied.

425 ● A eb:PartInfo/eb:PartProperties/eb:Property/@name="CharacterSet" value is RECOMMENDED
426 to identify the character set of the payload before compression was applied.

428 Example:

429 `<eb:PartInfo href="cid=f00@example.com" <mailto:cid=f00@example.com >>`

430 `<eb:PartProperties>`

431 `<eb:Property name="MimeType">application/xml</eb:Property>`

432 `<eb:Property name="CharacterSet">utf-8</eb:Property>`

433 <eb:Property name="Compressed"/>
 434 </eb:PartProperties>

435 <eb:PartInfo>

436

437 An additional PMode parameter is defined, **that MUST be supported**:

- 438 ● **PMode[1].PayloadService.Compression:** {true / false}

439 **True:** some attached payload(s) may be compressed over this MEP segment.

440 **False** (default): no compression is used over this MEP segment.

441

442 NOTE: the requirement for Compression feature applies to both conformance profiles (AS4 ebHandler
 443 and AS4 light Client)

444

445 3.2 Reception Awareness features and Duplicate Detection

446

447 These capabilities are making use of the eb:Receipt **as the sole type of acknowledgement**. Duplicate
 448 detection only relies on the eb:MessageInfo/eb:MessageId.

449

Features	Profile requirements
Reception awareness error handling (REQUIRED support)	Ability for the MSH expecting an eb:Receipt to generate an error in case no eb:Receipt has been received for a sent message. It is RECOMMENDED that this error be a new error: Code = EBMS:0301, Short Description = MissingReceipt, Severity = Failure, Category = Communication. Ability for the MSH expecting an eb:Receipt to report a MissingReceipt error to the message Producer
Message Retry (OPTIONAL support)	Ability for a User message sender that has not received an expected eb:Receipt to resend the User message. If doing so, the eb:MessageInfo/eb:MessageId element of the resend message and of the original User message MUST be same. [removed: However, the eb:MessageInfo/eb:Timestamp MUST be different.] When resending a message for which non-repudiation of receipt is required, the sender MUST ensure that the hash values for the digests to be included in the Receipt (i.e. the content of MessagePartNRInformation elements), do not vary from the original message to the retry(ies), so that non-repudiation of receipt can be asserted based on the original message and the receipt of any of its retries.

Duplicate Detection (REQUIRED support)	Ability for the MSH receiving a User message to detect and/or eliminate duplicates based on eb:MessageInfo/eb:MessageId. If duplicates are just detected (not eliminated) then at the very least it is REQUIRED that the Receiving MSH notifies its application (message Consumer) of the duplicates. For examples, these could be logged. Related quantitative parameters (time window for the detection, or maximum message log size) are left for implementors to decide.
Others	

450

451 NOTE: these requirements apply to both conformance profiles (AS4 ebHandler and AS4 light Client)

452 **The following additional PMode parameters are defined and MUST be supported:**

453

- 454 • **PMode[1].ReceptionAwareness:** (true / false)
- 455 • **PMode[1].ReceptionAwareness.Replay:** (true / false)
- 456 • **PMode[1].ReceptionAwareness.Replay.Parameters:** (contains a composite
457 string specifying: (a) maximum number of retries or some timeout, (b) frequency of
458 retries or some retry rule). The string contains a sequence of parameters of the
459 form: name=value, separated by either comas or `;`. Example:
460 "maxretries=10,period=3000", in case the retry period is 3000 ms.
- 461 • **PMode[1].ReceptionAwareness.DuplicateDetection:** (true / false)
- 462 • **PMode[1].ReceptionAwareness.DetectDuplicates.Parameters:** (contains a
463 composite string specifying either (a) maximum size of message log over which
464 duplicate detection is supported, (b) maximum time window over which duplicate
465 detection is supported). The string contains a sequence of parameters of the form:
466 name=value, separated by either comas or `;`. Example:
467 "maxsize=10Mb,checkwindow=7D", in case the duplicate check window is
468 guaranteed of 7 days minimum.

469

470 **3.3 Alternative Pull Authorization**

471 In addition to the two authorization options described in the AS4 Conformance Profile (section 2.1.1), an
472 implementation MAY optionally decide to support a third authorization technique, based on transient
473 security (SSL or TLS).

474 SSL/TLS can provide certificate-based client authentication. Once the identity of the Pulling client is
475 established, the Security module may pass this identity to the ebms module, which can then associate it
476 with the right authorization entry, e.g. the set of MPCs this client is allowed to pull from.

477 This third authorization option – compatible with AS4 although not specified in ebMS Core V3 - relies on
478 the ability of the ebms module to obtain the client credentials. This capability represents an (optional) new
479 feature.

480 Pull request authentication service, there may be no need for any WS-Security headers in the Pull
481 request at all.

482

483 **3.4 Semantics of Receipt in AS4**

484

485 The notion of Receipt in ebMS V3 is not associated with any particular semantics. However, when
486 combined with security (signing), it is intended to support Non Repudiation of Receipt (NRR).

487 In AS4, the eb:Receipt message serves both as a business receipt (its content is profiled in Section 2),
488 and as a reception indicator, being a key element of the reception awareness feature. No particular
489 delivery semantics can be assumed however: the sending of an eb:Receipt only means the following,
490 from a message processing viewpoint:

491 (a) The related ebMS user message has been received and is well-formed.

492 (b) The Receiving MSH is taking responsibility for processing this user message, However, no
493 guarantee can be made that this user message will be ultimately delivered to its Consumer
494 application (this responsibility lays however now on the Receiver side).

495 The meaning of NOT getting an expected Receipt, for the sender of a related user message, is one of the
496 following:

- 497 1. The user message was lost and never received by the Receiving MSH.
- 498 2. The user message was received, but the eb:Receipt was never generated, e.g. due to a faulty
499 configuration (PMode).
- 500 3. The user message was received, the eb:Receipt was sent back but was lost on the way.

501 See section 4.1.8 for AS4 usage rules about Receipts.

502

503

4 AS4 Usage Profile of ebMS 3.0

504

505

506 While the previous sections were describing messaging handler requirements for AS4 compliance (i.e.
507 mostly intended for product developers), this section is about configuration and usage options.

508 This section is split in two major subsections:

509

- 510 • **The AS4 Usage Rules:** this section is stating the rules for using messaging features in an AS4-
511 compliant way.
- 512 • **The AS4 Usage Agreements:** this section is reminding the users of what are the main options
513 left open by the AS4 profiles, that they have to agree on in order to interoperate.

514

515 Both sections are about features that are under responsibility of the user when using an AS4-compliant
516 product.

517

4.1 AS4 Usage Rules

518

4.1.1 Core Components / Modules to be Used

521 This table summarizes which functional modules in the ebMS V3 specification are required to be
522 implemented by the AS4 profile, and whether or not these modules are actually profiled for AS4.

523

ebMS V3 Component Name and Reference	Profiling status
Messaging Model (section 2)	Usage: Required Profiled: Yes Notes: This Profile only supports the One-Way/Push MEP (Sync and Async) and the One-Way/Pull MEP
Message Pulling and Partitioning (section 3)	Usage: Required Profiled: No Notes: The profiling of QoS associated with Pulling is defined in another module. The MPC and pulling feature itself are not profiled.
Processing Modes (section 4)	Usage: Required Profiled: Yes
Message Packaging (section 5)	Usage: Required

	<p>Profiled: Yes</p> <p>Notes: Default business process defines acceptable defaults for Role, Service, and Action. Bundling options for message headers (piggybacking) are restricted.</p>
Error Handling (section 6)	<p>Usage: Required</p> <p>Profiled: Yes</p> <p>Notes: Addition of some new Error Codes regarding Reception Awareness</p>
Security Module (section 7)	<p>Usage: Required</p> <p>Profiled: Yes</p> <p>Notes: Guidance regarding which part(s) of the message may be encrypted and included in the signature. Further guidance on how to secure the PullRequest Signal and the preventing of replay attacks..</p>
Reliable Messaging Module (section 8)	<p>Usage: Not Required</p> <p>Profiled: No</p> <p>Notes: This profile does not require the use of the Reliable Messaging Module using either WS-ReliableMessaging or WS-Reliability. It relies instead on eb:Receipts for supporting a light reliability feature called "Reception Awareness".</p>

524

525 **4.1.2 Bundling rules**

526

Scope of the Profile Feature	Defines bundling (or "piggybacking") rules of ebMS MEPs, including Receipts.
Specification Feature	
Specification Reference	ebMS v3.0, Section 2.2
Profiling Rule (a)	<p>This profile supports the One-Way/Push MEP.</p> <p>Both synchronous and asynchronous transport channels for the response (eb:Receipt) are allowed by this profile.</p> <p>When sending a Receipt for this MEP, a Receiving MSH conforming to this profile SHOULD NOT bundle the Receipt with any other ebMS message header or body.</p>
Profiling Rule (b)	<p>This profile supports the One-Way/Pull MEP. When sending a Receipt for this MEP, a Receiving MSH conforming to this profile SHOULD NOT bundle the Receipt with any other ebMS message header (including a PullRequest signal) or message body,</p>
Test References	

527

528

529 **4.1.3 Security Element**

530

Specification Feature	Use of WSS features
Specification Reference	ebMS v3.0, Section 7.1
Profiling Rule (a)	When using digital signatures or encryption, an AS4 MSH implementation is REQUIRED to use the Web Services Security X.509 Certificate Token Profile [WSS11-X509].
Alignment	[WSS11] Anthony Nadalin, et al, eds., <i>Web Services Security: SOAP Message Security 1.1</i> , 2005. < http://docs.oasis-open.org/wss/v1.1/ > [WSS11-X509] A. Nadalin, et al, eds., <i>Web Services Security X.509 Certificate Token Profile 1.1</i> , 2006.
Test References	
Notes	

531

532 **4.1.4 Signing Messages**

533

Specification Feature	Digital Signatures for SOAP message headers and body
Specification Reference	ebMS v3.0, Section 7.2
Profiling Rule (a)	AS4 MSH implementations are REQUIRED to use Detached Signatures as defined by the XML Signature Specification [XMLDSIG] when signing AS4 user or signal messages. Enveloped Signatures as defined by [XMLDSIG] are not supported by or authorized in this profile.
Profiling Rule (b)	AS4 MSH implementations are REQUIRED to include the entire eb:Messaging SOAP header block and the SOAP Body in the signature.
Alignment	
Test References	

534

535 **4.1.5 Signing SOAP with Attachments Messages**

536

Specification Feature	Signing attachments
-----------------------	---------------------

Specification Reference	ebMS v3.0, Section 7.3
Profiling Rule (a)	AS4 MSH implementations are REQUIRED to use the Attachment-Content-Only transform when building application payloads using SOAP with Attachments [SOAPATTACH]. The Attachment-Complete transform is not supported by this profile.
Profiling Rule (b)	AS4 MSH implementations are REQUIRED to include the entire eb:Messaging header block and all MIME body parts of included payloads in the signature.
Alignment	
Test References	

537

538 4.1.6 Encrypting Messages

539

Specification Feature	
Specification Reference	ebMS v3.0, Section 7.4
Profiling Rule (a)	AS4 MSH implementations are SHALL NOT encrypt the eb:PartyInfo section of the eb:Messaging header. Other child elements of the eb:Messaging header MAY be encrypted or left unencrypted as defined by trading partner agreements or collaboration profiles.
Profiling Rule (b)	If an AS4 user message is to be encrypted and the user-specified payload data is to be packaged in the SOAP Body, AS4 MSH implementations are REQUIRED to encrypt the SOAP Body.
Alignment	
Test References	

540

541 4.1.7 Encrypting SOAP with Attachments Messages

542

Specification Feature	Encryption of message attachments.
Specification Reference	ebMS v3.0, Section 7.5
Profiling Rule (a)	If an AS4 user message is to be encrypted and the user-specified payload data is to be packaged in conformance with the [SOAPATTACH] specification, AS4 MSH implementations are REQUIRED to encrypt the MIME Body parts of included payloads.
Alignment	

Test References	
Notes	

543

544 **4.1.8 Generating Receipts**

545

Specification Feature	eb:Receipt signal messages
Specification Reference	ebMS v3.0, Section 7.12..2 (Persistent Signed Receipt) ebMS v3.0, Section 5.2.3.3, eb:Messaging/eb:SignalMessage/eb:Receipt
Profiling Rule (a): Receipts for reception awareness	When a Receipt is to be used solely as a reception indicator (for reception awareness), the sender of the Receipt MAY decide to not insert the <code>ebbsig:NonRepudiationInformation</code> child element. No other element than <code>ebbsig:NonRepudiationInformation</code> is allowed as child of <code>eb:Receipt</code> . If this element is not used, then <code>eb:Receipt</code> MUST be empty.
Profiling Rule (b): Receipts for Non Repudiation of Receipt (NRR)	<p>Non Repudiation of Receipt (NRR) requires <code>eb:Receipt</code> signals to be signed, and to contain digests of the original message parts for which NRR is required.</p> <p>When signed receipts as requested in AS4 that make use of default conventions, the Sending message handler (i.e. sending messages for which signed receipts are expected) MUST identify message parts using Content-Id values in the MIME headers, and MUST sign the SOAP body and all attachments using the http://docs.oasis-open.org/wss/oasis-wss-SwAProfile-1.1#Attachment-Content-Signature-Transform within the SignedInfo References list.</p> <p>As a reminder, the Sending message handler MUST not encrypt any signed content before signing (Section 7.6 in ebMS V3). If using compression in an attachment, the Sending message handler MUST sign the data after compression (see section 3.1). Variations from default conventions can be agreed to bilaterally, but conforming implementations are only required to provide receipts using the default conventions described in this section.</p>
Profiling Rule (c)	<p>An AS4 message that has been digitally signed MUST be acknowledged with a message containing an <code>eb:Receipt</code> signal that itself is digitally signed. The <code>eb:Receipt</code> MUST contain the information necessary to provide nonrepudiation of receipt of the original message, as described in profiling rule (b).</p> <p>NOTE: the digest(s) to be inserted in the <code>ebbp:MessagePartNRInformation</code> element(s) or the Receipt, related to the original message parts for which a receipt is required, may be obtained from the signature information of the original message (<code>ds:SignedInfo</code> element), as only those parts that have been signed are subject to NRR. This means a Receiving message handler may not have to compute digests outside its security module.</p>

Alignment	
Test References	

546

547

548 4.1.9 MIME Header and Filename information

549

Specification Feature	Optional presence of a “filename” value in “Content-disposition” header on MIME body parts:
Specification Reference	MIME specification (IETF) [RFC2045]
Profiling Rule (a)	The “Content-disposition” header on MIME body parts, when used, MUST carry filename information. Implementations MUST support the setting (when sending) and reading (when receiving) of “Content-disposition” header,
Profiling Rule (b)	When end users wish to supply filenames and have that information confidential, they SHOULD use TLS/SSL based encryption.
Alignment	
Test References	

550

551

552 4.2 AS4 Usage Agreements

553

554 This section defines the operational aspect of the profile: configuration aspects that users have to agree
 555 on, mode of operation, etc. **This section is not normative and is provided here only as guidance for users.**

556 All the user agreement options related to a specific type of message exchange instance (e.g. related to a
 557 specific type of business transaction) are controlled by the Processing Mode (PMode) parameters defined
 558 in the ebMS Core V3 specification. This section only lists the parameters that are particularly relevant to
 559 AS4.

560

561 4.2.1 Controlling Content and Sending of Receipts

562

Scope of the Profile Feature	Choose among options in sending Receipts.
Specification Feature	

Specification Reference	ebMS v3.0, Section 2.2
Usage Profiling (a)	<p>Must eb:Receipts be used for non-repudiation of receipt (NRR), or just act as reception awareness feature? For non-repudiation, the eb:Receipt element must contain a well-formed ebbp:NonRepudiationInformation element. This is indicated by the new PMode parameter:</p> <p>Pmode[1].Security.SendReceipt.NonRepudiation : value = 'true' (to be used for non-repudiation of receipt), value = 'false' (to be used simply for reception awareness).</p>
Usage Profiling (b)	<p>Receipts for One-Way/Push MEP:</p> <p>Both synchronous and asynchronous transport channels for the response (eb:Receipt) are allowed by this profile. and Callback)</p> <p>This option is controlled by PMode parameter: ,</p> <ul style="list-style-type: none"> • Pmode[1].Security.SendReceipt.ReplyPattern: value = 'Response' (sending receipts on the HTTP response or back-channel). • Pmode[1].Security.SendReceipt.ReplyPattern: value = 'Callback' (sending receipts using a separate connection.)
Usage Profiling (c)	<p>Receipts for the One-Way/Pull MEP: ,</p> <p>Pmode[1].Security.SendReceipt.ReplyPattern: value = 'Callback' (sending receipts using a separate connection, and not bundled with PullRequest.)</p>
Test References	
Notes	

563

564 4.2.2 Error Handling Options

565

Specification Feature	Error Handling options
Specification Reference	
Usage Profiling (a): Receiver-side error	<p>All Receiver-side error reporting options are left for users to agree on, including the choice to not report at all: (reformatting of font below)</p> <p>PMode[1].ErrorHandling.Report.ReceiverErrorsTo: recommendation is to report such Receiver-side errors to the Sender. Otherwise: reporting URI that is different from sender URI?</p> <p>PMode[1].ErrorHandling.Report.AsResponse : recommendation for one-way messages (except when pulling is in use) is value="true": report errors on the back-channel of erroneous messages. Errors for pulled messages can only be reported on a separate connection.</p> <p>PMode[1].ErrorHandling.Report.ProcessErrorNotifyConsumer : (true / false) for controlling escalating theerror to the application layer.</p>

Usage Profiling (b): Reception Awareness errors	<p>What is the behavior of a Sender that failed to receive a Receipt (even after message retries)?</p> <p>(a) No error reporting (in case no reception awareness required).</p> <p>(b) Error reporting from the Sender MSH to its message Producer (application-level notification). Error type: EBMS:0301: MissingReceipt (see Section 3.2 in Additional Features.)</p> <p>PMode parameter: (reformatting of font below)</p> <p>PMode[1].ErrorHandling.Report.MissingReceiptNotifyProducer: (new) true if (b), false if (a)</p> <p>PMode[1].ErrorHandling.Report.SenderErrorsTo: (in case an error should be sent about such failures – e.g. to a third party if not to the original Receiver of the non-acknowledged user message.)</p>
Usage Profiling (c): Error about Receipts	<p>How are errors about Receipt messages reported? (reformatting of font below)</p> <p>PMode[1].ErrorHandling.Report.SenderErrorsTo: reporting URI that is different from Receiver URI?</p> <p>PMode[1].ErrorHandling.Report.AsResponse : (true / false) NOTE: In case of Receipts already sent over the HTTP back-channel, can only be “false” meaning such errors will be sent over separate connection.</p> <p>PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer : (true / false) for controlling escalating the error to the application layer.</p>
Alignment	
Test References	
Notes	

566

567 4.2.3 Securing the PullRequest

568

Specification Feature	Pulling authorization options
Specification Reference	<p>ebMS v3.0, Section 7.11.x</p> <p>AS4 Conformance Profile authorization options (section 2.1.1)</p>
Usage Profiling (a)	<p>An AS4 Sending MSH may authenticate a Receiving MSH that sends a PullRequest in two ways:</p> <p>(a) (Option 1 in 2.1.1) Use of the WSS security header targeted to the “ebms” actor, as specified in section 7.10 of ebMS V3, with the wsse:UsernameToken profile.</p>

	<p>(b) (Option 2 in 2.1.1) by using [WSS11-X509] coupled with the Message Partition Channel that a Pull signal is accessing for pulling messages.</p> <p>PMode parameters: (reformatting of font below)</p> <p>PMode.Initiator.Authorization: must be set to true (the initiator of a Pull request must be authorized).</p> <p>PMode.Initiator.Authorization.username: (for option (a))</p> <p>PMode.Initiator.Authorization.password: (for option (a))</p> <p>PMode[1].Security.PModeAuthorize: must be set to true in the PMode leg describing the transfer of a pulled message.</p> <p>PMode[1].Security.X509.sign: (for option (b))</p> <p>PMode[1].Security.X509.SignatureCertificate: (for option (b))</p> <p>NOTE: in (b), PMode parameters about X509 are controlling both the authentication of PullRequest signals and authentication of other User Messages.</p>
Usage Profiling (b)	<p>PullRequest signals: are they sent using the HTTPS transport protocol with optional Client-side Authentication?</p> <p>PMode[1].Protocol.Address: The URL scheme will indicate whether HTTPS is used or not.</p>
Alignment	
Test References	
Notes	

569

570 4.2.4 Reception Awareness Parameters

571

Specification Feature	Message Replay and Duplicate Detection options
Specification Reference	N/A AS4 Profile: additional features (section 3)
Usage Profiling (a): Sender options	<p>In case Reception Awareness is used: what is the behavior of a Sender that did not receive a Receipt?</p> <p>(c) No message replay.</p> <p>(d) Resend the message. Replay parameters: to agree on: (1) retry number, (2) retry frequency.</p>

	<p>PMode parameters (additional to those defined in ebMS Core V3): (reformatting of font below)</p> <p>PMode[1].ReceptionAwareness: (true / false)</p> <p>PMode[1].ReceptionAwareness.Replay: (true / false)</p> <p>PMode[1].ReceptionAwareness.Replay.Parameters: (contains a composite string specifying: (a) maximum number of retries or some timeout, (b) frequency of retries or some retry rule.</p>
Usage Profiling (b): Receiver options	<p>Is duplicate detection enabled?</p> <p>(a) No. duplicates are not detected.</p> <p>(b) In addition to (a), a receiver detects and eliminates duplicates based on eb:MessageInfo/eb:MessageId.</p> <p>PMode parameters (additional to those defined in ebMS Core V3): (reformatting of font below)</p> <p>PMode[1].ReceptionAwareness.DuplicateDetection: (true / false)</p> <p>PMode[1].ReceptionAwareness.DuplicateDetection.Parameters</p>
Others	
Notes	

572

573

574 4.2.5 Default Values of Some PMode Parameters

575

Specification Feature	Default values and authorized values for main PMode parameters.
Specification Reference	ebMS 3.0, Appendix D.3
Usage Profiling (a)	<p>PMode.MEP parameter will be constrained to the following value:</p> <p>http://docs..oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay</p>
Usage Profiling (b)	<p>PMode.MEPbinding parameter will be constrained to the following values:</p> <p>http://docs..oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push</p> <p>http://docs..oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/pull</p>
Usage Profiling (c)	<p>PMode.Initiator.Role parameter will have the following default value:</p> <p>http://docs..oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator</p>
Usage Profiling (d)	PMode.Responder.Role parameter will have the following default value:

	http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder
Usage Profiling (e)	<p>PMODE[1].BusinessInfo.Service parameter will have the following default value:</p> <p>http://docs.oasis-open.org/ebxml-msg/as4/200902/service</p> <p><i>NOTE: this default is to be considered a PMode content default: absence of the PMode itself will cause the default value defined in the ebMS V3 specification (section 4.3) to apply. This value is usually enforced by the MSH implementation itself.</i></p>
Usage Profiling (f)	<p>PMODE[1].BusinessInfo.Action parameter will have the following default value:</p> <p>http://docs.oasis-open.org/ebxml-msg/as4/200902/action</p> <p><i>NOTE: this default is to be considered a PMode content default: absence of the PMode itself will cause the default value defined in the ebMS V3 specification (section 4.3) to apply. This value is usually enforced by the MSH implementation itself.</i></p>
Usage Profiling (g)	PMODE[1].Reliability parameters are not supported by this profile
Alignment	
Test References	
Notes	

576

577 4.2.6 HTTP Confidentiality and Security

578

Specification Feature	<p>HTTP Security Management and Options</p> <p>This table is intended as a guide for users, to specify their own agreements on HTTP confidentiality and security.</p>
Specification Reference	ebMS 3, Section 7, Appendix D.3.6.
Usage Profiling (a)	<p>Is HTTP transport-layer encryption required?</p> <p>What protocol version(s)?</p>
Usage Profiling (b)	What encryption algorithm(s) and minimum key lengths are required?
Usage Profiling (c)	What Certificate Authorities are acceptable for server certificate authentication?
Usage Profiling (d)	Are direct-trust (self-signed) server certificates allowed?
Usage Profiling (e)	Is client-side certificate-based authentication allowed or required?
Usage Profiling (f)	What client Certificate Authorities are acceptable?
Usage Profiling (g)	What certificate verification policies and procedures must be followed?

Alignment	
Test References	
Notes	

579

580 **4.2.7 Deployment and Processing requirements for CPAs**

581

Usage Profile Feature	CPA Access
Usage Profiling (a)	Is a specific registry for storing CPAs required? If so, provide details.
Usage Profiling (b)	Is there a set of predefined CPA templates that can be used to create given Parties' CPAs?
Usage Profiling (c)	Is there a particular format for file names of CPAs, in case that file name is different from CPAId value?
Others	

582

583 **4.2.8 Message Payload and Flow Profile**

584

Usage Profile Feature	Message Quantitative Aspects
Usage Profiling (a)	What are typical and maximum message payload sizes that must be handled? (maximum, average)
Usage Profiling (b)	What are typical communication bandwidth and processing capabilities of an MSH for these Services?
Usage Profiling (c)	Expected Volume of Message flow (throughput): maximum (peak), average?
Usage Profiling (d)	(Section 2.1.4) How many Payload Containers must be present?
Usage Profiling (e)	What is the structure and content of each container? [List MIME Content-Types and other process-specific requirements.] Are there restrictions on the MIME types allowed for attachments?
Usage Profiling (f)	How is each container distinguished from the others? [By a fixed ordering of containers, a fixed Manifest ordering, or specific Content-ID values.]. Any expected relative order of attachments of various types?
Usage Profiling (g)	Is there an agreement that message part filenames must be present in MIME Content-Disposition parameter ?
Others	

585

586 **4.2.9 Additional Deployment or Operational Requirements**

587

Usage Profile Feature	Operational or Deployment Conditions
Usage Profiling (a)	Operational or deployment aspects that are object to further requirements or recommendations.
Others	

588

589 5 Conformance Clauses

590

591 5.1 AS4 ebHandler Conformance Clause

592 In order to conform to the AS4 ebHandler Profile, an implementation must comply with all normative
593 statements and requirements in Section 2.1.

594 In particular, it must:

595 - observe all requirements stated as such in the Feature Set table of Section 2.1.1.

596 - comply with WS-I requirements listed in Section 2.1.2.

597 - support the PMode parameters as required in Section 2.1.3.

598 In addition, the implementation must implement the additional features as indicated in Section 3.

599 Finally, the implementation must support the Usage Rules defined in Section 4.1.

600 The Usage Agreements in Section 4.2 are not prescriptive, and implementations are free to support any
601 subset of the features described, that are not already mandated in sections 2.1, 3 or 4.1.

602

603 5.2 AS4 Light Client Conformance Clause

604 In order to conform to the AS4 Light Client Profile, an implementation must comply with all normative
605 statements and requirements in Section 2.2.

606 In particular, it must:

607 - observe all requirements stated as such in the Feature Set table of Section 2.2.1.

608 - comply with WS-I requirements listed in Section 2.2.2.

609 - support the PMode parameters as required in Section 2.2.3.

610 In addition, the implementation must implement the additional features as indicated in Section 3.

611 Finally, the implementation must support the Usage Rules defined in Section 4.1.

612 The Usage Agreements in Section 4.2 are not prescriptive, and implementations are free to support any
613 subset of the features described, that are not already mandated in sections 2.2, 3 or 4.1.

614

615 Appendix A Sample Messages

616 Receipts Samples

617

618 When the NonRepudiationInformation element is used in a Receipt, it contains a sequence of Message-
619 PartNRInformation items for each message part for which evidence of non repudiation of receipt is being
620 provided. In the normal default usage, these message parts are those that have been signed in the origin-
621 al message. Each message part is described with information defined by an XML Digital Signature Refer-
622 ence information item. The following example illustrates the ebMS V3 Signal Message header.
623

```
624 <eb3:Messaging Soap12:mustUnderstand="true" xmlns:wsu="http://docs.oasis-
625 open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="ValueOfMes-
626 sagingHeader">
627   <eb3:SignalMessage>
628     <eb3:MessageInfo>
629       <eb3:Timestamp>2009-11-06T08:00:09Z</eb3:Timestamp>
630       <eb3:MessageId>orderreceipt@seller.com</eb3:MessageId>
631       <eb3:RefToMessageId>orders123@buyer.com</eb3:RefToMessageId>
632     </eb3:MessageInfo>
633     <eb3:Receipt>
634       <ebbp:NonRepudiationInformation>
635         <ebbp:MessagePartNRInformation>
636           <dsig:Reference URI="#5cb44655-5720-4cf4-a772-19cd480b0ad4">
637             <dsig:Transforms>
638               <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-ex-
639 c-c14n#" />
640             </dsig:Transforms>
641             <dsig:DigestMethod Al-
642 gorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
643             <dsig:DigestValue>o9QDCwWSiGVQACEsJH5nqkVE2s0=</dsig:Di-
644 gestValue>
645           </dsig:Reference>
646         </ebbp:MessagePartNRInformation>
647         <ebbp:MessagePartNRInformation>
648           <dsig:Reference URI="cid:a1d7fdf5-d67e-403a-ad92-3b9deff25d43@buyer.-
649 com">
650             <dsig:Transforms>
651               <dsig:Transform Algorithm="http://docs.oasis-open.org/wss/oas-
652 is-wss-SwAProfile-1.1#Attachment-Content-Signature-Transform" />
653             </dsig:Transforms>
654             <dsig:DigestMethod Al-
655 gorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
656             <dsig:DigestValue>iWNSv2W6SxbOYZliPzZDcXAxrWI=</dsig:Digest-
```

```

657 Value>
658         </dsig:Reference>
659         </ebbp:MessagePartNRInformation>
660         </ebbp:NonRepudiationInformation>
661     </eb3:Receipt>
662 </eb3:SignalMessage>
663 </eb3:Messaging>
664

```

665 For a signed receipt, a Web Services Security header signing over (at least) the signal header is required.
666 An example WS-Security header is as follows :

667

```

668 <wsse:Security s:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
669 open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
670 xmlns:s="http://www.w3.org/2003/05/soap-envelope">
671     <wsu:Timestamp wsu:Id="_1" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
672 200401-wss-wssecurity-utility-1.0.xsd">
673         <wsu:Created>2009-11-06T08:00:10Z</wsu:Created>
674         <wsu:Expires>2009-11-06T08:50:00Z</wsu:Expires>
675     </wsu:Timestamp>
676     <wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
677 200401-wss-soap-message-security-1.0#Base64Binary"
678     ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
679 1.0#X509v3" wsu:Id="_2"
680     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
681 1.0.xsd">MIIFADCCBGmgAwIBAgIEOmitted</wsse:BinarySecurityToken>
682     <ds:Signature Id="_3" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
683         <ds:SignedInfo>
684             <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
685 c14n#" />
686             <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
687             <ds:Reference URI="#ValueOfMessagingHeader">
688                 <ds:Transforms>
689                     <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
690                         <InclusiveNamespaces PrefixList="xsd"
691     xmlns="http://www.w3.org/2001/10/xml-exc-c14n#" />
692                     </ds:Transform>
693                 </ds:Transforms>
694                 <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
695                 <ds:DigestValue>ZXnOmitted=</ds:DigestValue>
696             </ds:Reference>
697         </ds:SignedInfo>
698         <ds:SignatureValue>rxAP4of8JCpUkOmitted=</ds:SignatureValue>
699         <ds:KeyInfo>
700             <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-

```

```
701 open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
702     <wsse:Reference URI="#_2" ValueType="http://docs.oasis-
703 open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" />
704     </wsse:SecurityTokenReference>
705     </ds:KeyInfo>
706 </ds:Signature>
707 </wsse:Security>
708
709
```

710 **Appendix B Acknowledgments**

711 The following individuals were members of the committee during the development of this specification or
712 of a previous version of it:

713

714 Timothy Bennett, Drummond Group Inc. <timothy@drummondgroup.com>

715 Ian Jones, British Telecommunications plc <ian.c.jones@bt.com>

716 Jacques Durand, Fujitsu <jdurand@us.fujitsu.com>

717 Dale Moberg, Axway <dmoberg@axway.com>

718 Richard Emery, Axway <remery@us.axway.com>

719 John Voss, CISCO <jovoss@cisco.com>

720

Appendix C Revision History

721

Rev	Date	By Whom	What
	25 Jul 2008	J. Durand / Tim Bennett	Initial draft
Rev 02	28 Oct 2008	J. Durand	candidate CD draft
Rev 03	15 Feb 2009	J. Durand	Various edits, updates on Receipts, Message samples.
CD 2	10/03/09	J. Durand	CD 2 draft for PR

722