

4371 ReliableMessaging). In V3 the P-Mode parameter
4372 Reliability.AtLeastOnce.AckOnDelivery specifies this semantics which in general
4373 depends on the implementation: when "false", it is similar to V2 (on receipt).

4374 **F.2.3.2. Rule CM3-b: Reliability Contracts**

4375 The reliability contracts At-Least-Once delivery, At-Most-Once delivery, In-Order delivery, that in V3 are
4376 specified in the P-Mode, and also in the message header in case WS-Reliability is used, respectively
4377 map to V2 header elements: eb2:AckRequested, eb2:DuplicateElimination, eb2:MessageOrder.

4378 Any of the above reliability contracts requires the use of a reliable messaging module in V3, e.g. an
4379 implementation of WS-Reliability or of WS-ReliableMessaging.

4380 The delivery failure notification in V2 (always required for non-acknowledged messages) is supported by
4381 WS-Reliability and therefore by V3 using WS-Reliability. Such failure notification is not explicitly
4382 mandated by WS-ReliableMessaging, or could take place on either side. In order to achieve the same
4383 notification policy as in V2, when used in V3 an implementation of WS-ReliableMessaging must be
4384 extended with the same notification capability.

4385 **Note:**

4386 The conditions under which delivery failure is notified to the From Party (in eb2) or
4387 message Producer (in eb3) may be different.

4388 **F.2.3.3. Rule CM3-c: Duplicate Elimination**

4389 eb2:MessageHeader/eb2:DuplicateElimination maps to wsrn:Request/wsrn:DuplicateElimination in WS-
4390 Reliability. It maps to the AtMostOnce delivery assurance definition in WS-ReliableMessaging, assuming
4391 an implementation of WS-ReliableMessaging that supports this delivery assurance.

4392 **F.2.3.4. Rule CM3-d: Use of Sequences and Sequence Numbers**

4393 An eb2 message that contains either AckRequested or DuplicateElimination or both, and no
4394 eb2:MessageOrder, may map to a V3 message (when using WS-Reliability) with no wsrn:SequenceNum
4395 – only a wsrn:MessageId/@groupId value, which is unique for every such message.

4396 **Note:**

4397 The elements that identify a message sent reliably in V3 (wsrn:SequenceNum,
4398 wsrn:MessageId/@groupId in WS-Reliability, or
4399 /wsrn:Sequence/wsrn:MessageNumber in WS-ReliableMessaging) do NOT map to the
4400 ebMS message ID element (i.e. eb2:MessageData/eb2:MessageId in V2, and
4401 eb3:MessageInfo/eb3:MessageId in V3).

4402 **F.2.3.5. Rule CM3-e: Message Ordering**

4403 In case message ordering is required:

4404 eb2:MessageOrder maps to wsrn:Request/wsrn:MessageOrder.

4405 eb2:SequenceNumber maps to wsrn:Request/wsrn:SequenceNum (with WS-Reliability).

4406 The scope of a message sequence (and of the message ordering contract) is determined by
4407 eb2:ConversationId in V2, and by MessageId/@groupId in V3; i.e. sequence numbers must be unique
4408 within this scope.

4409 The feature maps to the InOrder delivery assurance definition in WS-ReliableMessaging, assuming an
4410 implementation of WS-ReliableMessaging that supports this delivery assurance.

4411 **F.2.3.6. Rule CM3-f: Expiration Timeout**

4412 In case eb2:MessageHeader/eb2:MessageData/eb2:TimeToLive is used for expressing the maximum
4413 time during which reliability mechanisms are required to handle the message, it maps to
4414 wsrn:Request/wsrn:ExpiryTime.

4415 **F.2.4. (CM4) MEP Mapping Rules**

4416 Defines how V2 header elements that control the MEP in use and its mapping to the underlying protocol,
4417 map into V3 and vice versa. Also defines how CPA elements that control ebMS V2 MEPs map to P-Mode
4418 parameter and vice-versa.

4419 **F.2.4.1. Rule CM4-a: One-Way/Push With No Signals**

4420 In V3, this MEP, with no ebMS signal and no reliability acknowledgments on the response (back-
4421 channel), will map to a V2 message with no SyncReply element in eb2 header. RefToMessageId must
4422 not be used in the V3 message (it has a strict MEP semantics). The agreements map as follows:

4423 V2 (CPA): syncReplyMode=none.

4424 V3 (P-Mode): PMode.MEP="One-way", PMode.MEPbinding="push",
4425 PMode.ErrorHandling.Report.AsResponse="false". PMode.Reliability.ReplyPattern must NOT be
4426 "Response".

4427 **F.2.4.2. Rule CM4-b: One-Way/Push With Signals**

4428 One-Way / Push in V3, with ebMS signal and reliability acknowledgments on the response (back-
4429 channel), will map to a V2 message with SyncReply element in eb2 header. RefToMessageId must not
4430 be used in the V3 message (it has a strict MEP semantics). The agreements map as follows:

4431 V2 (CPA): syncReplyMode= mshSignalsOnly.

4432 V3 (P-Mode): PMode.MEP="One-way", PMode.MEPbinding="push",
4433 PMode.ErrorHandling.Report.AsResponse="true", PMode.Reliability.ReplyPattern="Response".

4434 **F.2.4.3. Rule CM4-c: Two-Way/Sync With No Signals**

4435 In V3, this MEP, with no ebMS signal and no reliability acknowledgments on the response (back-
4436 channel), will map to a V2 message (1st leg) with SyncReply element in eb2 header. In both versions,
4437 the response message refers to the request (leg 1) using RefToMessageId. The agreements map as
4438 follows:

4439 V2 (CPA): (leg 1) syncReplyMode= responseOnly.

4440 V3 (P-Mode): PMode.MEP="Two-way", PMode.MEPbinding="sync",
4441 PMode.ErrorHandling.Report.AsResponse="false". PMode.Reliability.ReplyPattern may NOT be
4442 "Response".

4443 **F.2.4.4. Rule CM4-d: Two-Way/Sync With Signals**

4444 In V3, this MEP, with ebMS signal and reliability acknowledgments on the response (back-channel), will
4445 map to a V2 message (1st leg) with SyncReply element in eb2 header. In both versions, the response
4446 message refers to the request (leg 1) using RefToMessageId. The agreements map as follows:

4447 V2 (CPA): (leg 1) syncReplyMode= signalsAndResponse

4448 V3 (P-Mode): PMode.MEP="Two-way", PMode.MEPbinding="sync",
4449 PMode.ErrorHandling.Report.AsResponse="true". PMode.Reliability.ReplyPattern ="Response".

4450 **F.2.4.5. Rule CM4-e: Two-Way/Push-and-Push**

4451 In V3, this MEP will map to an exchange of two messages in V2, where the second message refers to the
4452 first one using RefToMessageId (as in V3). The agreements map as follows:

4453 Option 1: (signals may be sent back on underlying response)

4454 V2 (CPA): (leg 1 and leg 2) syncReplyMode= mshSignalsOnly.

4455 V3 (P-Mode): PMode.MEP="Two-way", PMode.MEPbinding="Push-and-Push".

4456 PMode.ErrorHandling.Report.AsResponse="true". PMode.Reliability.ReplyPattern="Response".

4457 Option 2: (signals may NOT be sent back on underlying response)

4458 V2 (CPA): (leg 1 and leg 2) syncReplyMode= none.
4459 V3 (P-Mode): PMode.MEP="Two-way", PMode.MEPbinding="Push-and-Push".
4460 PMode.ErrorHandling.Report.AsResponse="false". PMode.Reliability.ReplyPattern different from
4461 "Response".

4462 **F.2.5. (CM5) Signal Mapping Rules**

4463 **F.2.5.1. Rule CM5-a: Error Metadata Mapping**

4464 The metadata mapping of the Error elements in V2 and V3 is as follows. In some cases the semantics is
4465 close though not exactly same.

- 4466 (a) Cases where a straight mapping exist from V2 to V3:
- 4467 1. V2: Error/@severity (warning, error) maps to V3: eb:Error/@severity (respectively:
4468 warning, failure)
 - 4469 2. V2: Error/@codeContext maps to V3: eb:Error/@origin
 - 4470 3. V2: Error/@errorCode maps to V3: eb:Error/shortDescription
 - 4471 4. V2: Error/@location maps to V3: eb:Error/ErrorDetail
 - 4472 5. V2: Error/Description maps to V3: eb:Error/Description
 - 4473 6. V2: MessageData/RefToMessageId maps to V3: eb:Error/@refToMessageInError
- 4474 (b) Cases where error element in V2 has no specified counterpart in V3:
- 4475 1. V2: Error/@id. In V3 would map to: XML Id attribute.
- 4476 (c) Cases where error element in V3 has no specified counterpart in V2:
- 4477 1. V3: eb:Error/@errorCode
 - 4478 2. V3: eb:Error/@category

4479 **F.2.5.2. Rule CM5-b: Error Value Mapping**

4480 The value-equivalence between Errors in V2 and V3 is as follows, based on the semantics of these
4481 errors:

4482 Note: the severity levels may not map in some cases, meaning that processing may continue in V3 while
4483 aborting in V2.

- 4484 (a) Cases where a straight mapping exist from V2 to V3:
- 4485 1. V2: ValueNotRecognized maps to V3: ValueNotRecognized
 - 4486 2. V2: NotSupported maps to V3: FeatureNotSupported
 - 4487 3. V2: DeliveryFailure maps to V3: DeliveryFailure
 - 4488 4. V2: MimeProblem maps to V3: MimeInconsistency
- 4489 (b) Cases where a case by case mapping exist from V2 to V3:
- 4490 1. V2: Inconsistent may map to V3: ValueInconsistent, in some cases InvalidHeader
 - 4491 2. V2: SecurityFailure maps to V3: FailedAuthentication or FailedDecryption
 - 4492 3. V2: OtherXML may map to V3: Other
 - 4493 4. V2: Unknown maps to (in most cases) V3: Other
- 4494 (c) Cases where error value in V2 has no counterpart in V3:
- 4495 1. V2: TimeToLiveExpired: no counterpart (not relevant).
- 4496 (d) Cases where error value in V3 has no counterpart in V2:
- 4497 1. V3: ConnectionFailure,
 - 4498 2. V3: EmptyMessagePartitionChannel

- 4499 3. V3: ProcessingModeMismatch
4500 4. V3: DysfunctionalReliability

4501 **F.2.5.3. Rule CM5-c: Ping and Pong Services**

- 4502 (a) Ping Service:
- 4503 1. V2: Service element: urn:oasis:names:tc:ebxml-msg:service, and Action element
4504 containing: Ping.
- 4505 2. V3: Service element: http://docs.oasis-open.org/ebxml-
4506 msg/ebms/v3.0/ns/core/200704/service, and Action element: http://docs.oasis-
4507 open.org/ebxml-msg/ebms/v3.0/ns/core/200704/test
- 4508 (b) Pong Service:
- 4509 No corresponding Pong service in V3 Core specification. This feature may be defined in
4510 a forthcoming Part 2 (Advanced Features).

4511 **F.2.6. (CM6) Processing Mode Mapping Rules**

4512 These mapping rules, to be specified in a separate white paper, will define how the messaging subset of
4513 an existing CPA instance in V2 maps to a V3 P-Mode. They also provide guidance on how to represent
4514 a P-Mode with a CPA and related extensions.
4515

4516 APPENDIX G. Conformance

4517 This section introduces the notion of conformance profiles for MSH implementations. The expression
4518 "conformance profile" is to be understood in the sense of [QAFW]. A conformance profile in ebMS will
4519 define a class of implementations that may implement only a subset of this specification, and/or a
4520 particular set of options (e.g. transport protocol binding, SOAP version). This specification does not
4521 define nor recommend any specific conformance profile. Such conformance profiles will be defined
4522 separately from the ebMS standard, in an adjunct document. A particular conformance profile will be
4523 distinguished as the baseline for achieving interoperability between most implementations dedicated to
4524 e-Business or e-Government.

4525 The section defines a common structure and syntax for defining conformance profiles.

4526 Note: "Conformance profile" should not be confused with "usage profile":

- 4527 • *Conformance profile*: defines a set of capabilities that an MSH implementation must have. This is
4528 determined at development time regardless of the way the MSH is being used later.
- 4529 • *Usage profile*: defines a way of using an MSH implementation, that a community of users has
4530 agreed upon. This may in turn require a particular conformance profile.

4531 For example, a conformance profile may require that an MSH support the optional MessageProperties
4532 header element, meaning it is able to extract it from a received message or to add it to a message to be
4533 sent. In contrast, a usage profile will additionally require that some specific property name be present in
4534 the MessageProperty element of each message.

4535 The interpretation of normative material follows the general rule below, as a complement to RFC2119:

- 4536 • When the keywords OPTIONAL, SHOULD and MAY apply to the behavior of the implementation,
4537 the implementation is free to support these behaviors or not, as meant in [RFC2119].
- 4538 • When the keywords OPTIONAL, SHOULD and MAY apply to message contents that relate to a
4539 more general feature, an implementation that conforms to a profile requiring support for this
4540 feature MUST be capable of processing these optional message contents according to the
4541 described ebXML semantics.
- 4542 • The keywords REQUIRED, SHALL or MUST indicate features that an MSH must support or
4543 implement, but only within the context of a conformance profile requiring support for this feature
4544 or module containing this feature.
- 4545 • When an MSH receives a message that exhibits some content feature that is either
4546 recommended or required by the specification, and if this MSH implements a conformance profile
4547 that does not require support for that content feature, then it MUST generate a
4548 FeatureNotSupported error (see Section 6).

4549

4550

APPENDIX H. Acknowledgments

4551

The OASIS ebXML Messaging Services Technical Committee would like to acknowledge the contributions of its committee members, who at the time of publication were:

4552

4553

Hamid Ben Malek, Fujitsu Software <hbenmalek@us.fujitsu.com>

4554

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

4555

Ric Emery, Axway Inc. <remery@us.axway.com>

4556

Kazunori Iwasa, Fujitsu Limited <kiwasa@jp.fujitsu.com>

4557

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

4558

Rajashekar Kailar, Centers for Disease Control and Prevention <kailar@bnetal.com>

4559

Dale Moberg, Axway Inc. <dmoberg@us.axway.com>

4560

Sacha Schlegel, Individual <sacha@schlegel.li>

4561

Pete Wenzel, Sun Microsystems <pete.wenzel@sun.com>

4562

4563

In addition, the following former Technical Committee members contributed to this specification:

4564

Doug Bunting, Sun Microsystems <doug.bunting@sun.com>

4565

Matthew MacKenzie, Adobe Systems Incorporated <mattm@adobe.com>

4566

Jeff Turpin, Axway Inc. <jturpin@us.axway.com>