



中华人民共和国国家标准

GB/T 19256.3—2006

基于 XML 的电子商务 第 3 部分:消息服务规范

Electronic business eXtensible Markup Language(ebXML)—
Part 3: Message Service specification(ebMS)

(ISO/TS 15000-2:2004, MOD)



2006-09-18 发布

2007-03-01 实施



中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会

发布

前 言

GB/T 19256《基于 XML 的电子商务》目前分为下列 9 个部分：

- 第 1 部分：技术体系结构；
- 第 2 部分：协同规程轮廓与协议规范；
- 第 3 部分：消息服务规范；
- 第 4 部分：注册系统信息模型规范；
- 第 5 部分：注册服务规范；
- 第 6 部分：业务过程规范模式；
- 第 7 部分：业务过程构件设计规则；
- 第 8 部分：报文设计规则；
- 第 9 部分：核心构件与业务信息实体规范。

将来还可能增加新的部分。

本部分为 GB/T 19256 的第 3 部分。

本部分修改采用 ISO/TS 15000-2:2004《基于 XML 的电子商务 第 2 部分：消息服务规范 (ebMS)》。

本部分与 ISO/TS 15000-2:2004 的主要差异如下：

- 按照国家标准的编写格式要求对原文的一些章节及章节编号错误做了适当的调整；
- 引言对应于原文的“Introduction”的部分内容；
- 第 1 章对应于原文的 1.2.1 的部分内容；
- 第 2 章对应于原文的 1.1.4；
- 增加第 3 章“术语与定义”；
- 第 4 章对应于原文的“Introduction”的部分内容；
- 第 5 章对应于原文的第 I 部分；
- 第 6 章对应于原文的第 II 部分；
- 附录 A～附录 C 对应于原文的附录 A～附录 C；
- 参考文献对应于原文的参考文献的部分内容；
- 对文中一些错误的地方进行了改正。

本部分的附录 A、附录 B 和附录 C 为规范性附录。

本部分由中国标准化研究院提出。

本部分由全国电子商务标准化技术委员会归口。

本部分起草单位：中国标准化研究院。

本部分主要起草人：任冠华，胡涵景，魏宏，刘碧松，孙文峰，陈煌，刘颖，章建方。

ISO 前言

国际标准化组织(ISO)是一个世界性的各国标准机构(ISO 成员团体)的联盟。国际标准的制定工作一般通过 ISO 技术委员会完成。每个成员团体如果对某个技术委员会建立的某个项目感兴趣,则有权向该技术委员会表述意见。任何与 ISO 有联络关系的官方和非官方国际组织都可以直接参与该项工作。ISO 与国际电工委员会(IEC)在电工技术标准的所有领域密切合作。

国际标准是根据 ISO/IEC 标准化工作导则第 3 部分规定的规则进行起草的。

技术委员会的主要任务是起草国际标准。由技术委员会正式通过的国际标准草案须经各成员团体投票表决。一项国际标准草案至少需要 75% 的成员团体投票赞成,才能作为正式的国际标准发布。

在某些情况下,尤其是当市场有迫切需求时,技术委员会也可以决定发布其他类型的国际标准:

- ISO 公共适用规范(ISO/PAS):它代表 ISO 标准工作组中的技术专家间达成的协议,且该工作组的上一级委员会中有 50% 以上的成员投票赞成对其进行发布。
- ISO 技术规范(ISO/TS):它代表技术委员会成员间达成的协议,且该委员会成员中有 2/3 以上的成员投票赞成对其进行发布。

ISO/PAS 或 ISO/TS 发布 3 年之后要进行复审,以确定其是否在下一个 3 年内继续有效,或通过修订将其转化成国际标准,或将其废止。如果被确认为继续有效,那么在随后的另一个 3 年之后,仍需对其再进行一次复审,但这时的复审结论只能在将其转化成国际标准或者废止两者中做出选择。

需要注意的是,本部分的某些内容可能会涉及一系列专利权。ISO 不负责标出这些专利权。

ISO/TS 15000-2 是由 OASIS 下设的 ebXML 消息服务技术委员会起草(原文件名为《OASIS/ebXML 消息服务规范(2.0 版)》),由 ISO/TC 154(行政、商业和工业中的过程、数据元和文档格式)采纳。ISO/TS 15000-2 与 OASIS/ebXML 消息服务规范 2.0 版在内容上完全相同。

ISO/TS 15000 系列标准在《基于 XML 的电子商务》的总标题下由下列部分组成:

- 第 1 部分:协同规程轮廓与协议规范(ebCPP);
- 第 2 部分:消息服务规范(ebMS);
- 第 3 部分:注册系统信息模型规范(ebRIM);
- 第 4 部分:注册服务规范(ebRS)。

引 言

本部分是基于 XML 的电子商务系列标准之一。该系列标准的目的是实现统一的全球电子市场。在这个市场中,任何规模、任意地理位置的企业都可以通过交换基于 XML 的消息进行业务往来。该系列标准给出了一种构建完整的、模块化的电子商务框架的方法。

本部分着重定义了用于交换电子商务消息的独立于通信协议的方法。它定义了特定的封装结构以支持业务信息可靠、安全地传输。此外,本部分还定义了一种灵活的封装技术,允许消息中包含任何格式类型的负载。这种特点保证了使用传统语法(例如 UN/EDIFACT、ASC X12 或 HL7)的原有电子商务系统的使用者能够和新兴技术的使用者一起利用 ebXML 基础设施。

本部分定义了保证两个参与方间进行安全、可靠消息交换的 ebMS 协议。它描述了:

- a) 用于对参与双方间传输的负载数据进行打包的 ebXML 消息结构;
- b) 通过数据通信协议进行发送和接收消息的 MSH 的行为。

本部分独立于负载和所使用的通信协议。附录描述了如何将本部分与 HTTP 协议[RFC2616]和 SMTP 协议[RFC2821]组合起来使用。

0.1 功能概述

本部分由下列主题组成:

0.1.1 核心功能

- a) 打包规范——规定了如何将 ebXML 消息及其相关部分打包成一种可以使用诸如 HTTP 或 SMTP 的通信协议进行发送的格式(见 5.1.1)。
- b) ebXML SOAP Envelope 扩展——规定了消息服务中生成或处理 ebXML 消息所必须的信息结构和成分(见 5.1.3)。
- c) 错误处理——规定了 ebXML 消息服务如何向另一个 ebXML MSH 报告其所检查到的错误(见 5.3.2)。
- d) 安全性——规定了 ebXML 消息安全语义(见 5.3.1)。
- e) SyncReply(同步应答)——向下一个 MSH 指出应答是否以同步的形式返回(见 5.3.3)。

0.1.2 附加特征

- a) 可靠消息传输——可靠消息传输功能定义了一种互操作协议。按照该协议,任意两个消息服务实施能够传且只传一次语义来可靠地交换消息(见 6.1)。
- b) 消息状态服务——描述了使一个服务能够发现另一个 MSH 或一个单独消息的状态的服务(见 6.2 和 6.3)。
- c) 消息顺序——保证接收方 MSH 接收消息的顺序(见 6.4)。
- d) 逐跳——通过中间的 MSH 节点发送消息(见 6.5)。

0.1.3 附录

- a) 附录 A 模式——本规范性附录包括用于 ebXML SOAP Header(头)和 Body(体)扩展的 XML 模式定义[XMLSchema]。
- b) 附录 B 通信协议封装映射——本规范性附录描述了如何通过 HTTP 和 SMTP 来传输符合 ebXML 消息服务的消息。
- c) 附录 C 安全轮廓——讨论了安全服务轮廓。

0.2 文件约定

属性名称以小写字母开头,元素名称以大写字母开头。

本部分凡涉及密码相关内容,按国家有关法规实施。本部分不涉及任何具体的密码运算,所有密码运算均在符合国家有关法规的密码设备中进行。本部分中所引用的 MD2、MD5 和 RSA 等密码算法均为举例说明。

0.3 说明和假设

本部分假设读者对通信协议、MIME、XML、SOAP、带有附件的 SOAP 消息以及安全技术有一定的了解。

所有的例子都为资料性的。如果本部分与例子之间存在不一致,以本部分为准。

建议实施者在使用本部分之前一定要阅读和理解 GB/T 19256.2 及其涵义。

基于 XML 的电子商务

第 3 部分:消息服务规范

1 范围

本部分定义了通过诸如 HTTP 或 SMTP 的通信协议传输 ebXML 消息的消息封装和头文档模式,以及发送和接收 ebXML 消息的软件行为。同时本部分给出了用于 ebXML 基础设施的消息封装、路由和传输设备。此外,本部分还给出了必要的安全性和可靠性特征,以支持国际电子商务。

本部分适用于将要实施 ebXML 消息服务的软件开发群体。

2 规范性引用文件

下列文件中的条款通过 GB/T 19256 的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 18793—2002 信息技术 可扩展置标语言(XML)1.0(W3C RFC-xml:1998, NEQ)

GB/T 19256.2 基于 XML 的电子商务 第 2 部分:协同规程轮廓与协议规范(GB/T 19256.2—2006, ISO/TS 15000-1: 2004, MOD)

GB/T 19256.6 基于 XML 的电子商务 第 6 部分:业务过程规范模式(GB/T 19256.6—2006, UN/CEFACT:ebXML business process specification schema, MOD)

ebXML Glossary(version 0.99), UN/CEFACT

3 术语、定义和缩略语

3.1 术语和定义

下列术语和定义适用于本部分。

3.1.1

通信协议信封 communication protocol envelope

ebXML 消息的最外层信封,如 HTTP 或 SMTP。

3.1.2

消息信封 message envelope

包含 ebXML 消息的两个主要部分(头和负载容器)的、独立于通信协议的信封,特指 MIME 多部分/相关的信封。

3.1.3

消息头 message header

关于成功生成或处理 ebXML 消息传输服务和 ebXML 消息等的必要信息的结构和构成的规定。

3.1.4

消息传输能力 messaging capabilities

支持参与方间文档交换的能力集合。如通信协议及其参数、安全定义以及结束和接收消息的通用特性。

3.1.5

消息传输服务 messaging service

能够使贸易伙伴间进行互操作的、安全可靠的消息交换的框架。

3.1.6

负载 payload

非 ebXML 信封部分的数据/信息。

3.1.7

负载容器 payload container

用于封装 ebXML 消息的真实负载的可选容器。如果存在一个负载,则负载容器必须由 MIME 头部分(ebXML 负载信封)和内容部分(负载本身)组成。

3.1.8

负载信封 payload envelope

与 MIME 部分相关的特定 MIME 头。

3.2 缩略语

下列缩略语适用于本部分。

CPA	Collaboration-Protocol Agreement	协同规程协议
CPP	Collaboration-Protocol Profile	协同规程轮廓
ebMS	ebXML Messaging Service	ebXML 消息服务
ebXML	electronic business eXtensible Markup Language	电子商务可扩展置标语言
EDI	Electronic Data Interchange	电子数据交换
FTP	File Transfer Protocol	文件传输协议
HTTP	HyperText Transfer Protocol	超文本传输协议
IETF	Internet Engineering Task Force	互联网工程任务组
JMS	Java Message Service	Java 消息服务
MIME	Multipurpose Internet Mail Extensions	多功能因特网邮件扩展
MSH	Message Service Handler	消息服务处理器
OASIS	Organization for the Advancement of Structured Information Standards	结构化信息标准推进组织
SAML	Security Assertion Markup Language	安全声明标记语言
SMTP	Simple Message Transfer Protocol	简单邮件传输协议
SOAP	Simple Object Access Protocol	简单对象访问协议
SSL	Security Socket Layer	安全套接字层
TCP/IP	Transmission Control Protocol/Internet Protocol	传输控制与网际协议
TLS	Transport Layer Security	传输层安全性
URN	Uniform Resource Name	统一资源名称
W3C	World Wide Web Consortium	互联网联合组织
XML	eXtensible Markup Language	可扩展置标语言

4 操作概念

4.1 背景和目的

传统的业务信息交换必须符合各种基于标准的语法。这些交换在很大程度上是以由主机和批处理衍生出的 EDI 标准为基础的。其中一些标准定义了与特定通信协议的绑定。这些 EDI 技术应用效果很好,然而它们的实施难度很大,而且成本很高。因此,这些系统的应用通常仅局限于具有成熟信息技

术能力的大型企业。

基于 XML 的业务交换的迅速增加成为了定义一种新的全球电子交换方式的催化剂,利用这种新的方式可以使得所有的业务活动,不论规模大小,都能够进行电子商务活动。本部分的主要目的是促进 XML 框架内的电子商务消息的交换。标识为 ebXML 消息“负载”的业务消息不一定必须用 XML 来表示。传统的 EDI 格式与基于 XML 的消息都可以通过 ebMS 进行传输。实际上,ebMS 的负载可采用任何一种数字形式来表示,如 XML、ASC X12、HL7、AIAG E5、数据库表、二进制图像文件等等。

ebXML 体系结构要求 ebMS 协议可以加载于任何一种可用的通信协议上。因此,本部分并不要求只使用一种特定的通信协议。虽然本版本只给出了与 HTTP 和 SMTP 协议的绑定,但也可以使用其他协议。

ebXML 需求规范[ebREQ]要求通信必须安全可靠。ebXML 的任务主要集中于调整现有的和正在形成的技术,不鼓励制定新的协议。因此,本部分定义了现有安全标准和协议的语境中的安全性,规定了现有标准可以满足的需求,其他需求则必须推迟到新技术或标准可用的时候再再进行规定,例如特殊消息头元素的加密。

在[ebREQ]中定义的可靠性需求与通过通信信道进行的 ebXML 消息传输有关。本部分给出了满足[ebREQ]的机制。本部分的可靠消息传输元素为通信层提供了可靠性,但并不将其作为对本部分支持的应用程序的业务层确认。这是个非常重要的区别。业务过程通常期望对其生成的消息进行响应。该响应可以对应应用程序所接收的消息采用简单的确认形式,其中应用程序用于接收消息或接收反映源消息功能的伴随消息。这些消息不在本部分范围之内。在本部分中定义的确认不能指出 ebXML 消息的负载在语法上是否正确,也不能确认负载消息的正确性。它无法指出这些信息的业务可接受性或负载内容的一致性。本部分的目的是使发送方相信接收方 MSH 能够安全、完整地接收到消息。

ebMS 基本体系结构假设消息是在两个符合本部分的 MSH 节点之间进行交换。这两个 MSH 节点给出了根据要求进行扩展后可支持逐跳环境的跳-跳模型。逐跳环境允许消息的下一个目的地作为中间 MSH,而不是由最初的发送 MSH 标识的“接收 MSH”。ebMS 体系结构假设消息发送方可以不知道传输消息的特定路径。然而,必须假定最初的发送方知道消息的最终接收方以及一个或多个中间跳中的第一个跳。

ebMS 支持“服务质量”的概念。服务质量的水平是根据消息交换直接涉及的两个参与方之间的协议进行控制的。实际上,两个参与方间可以存在多个协议。为了满足业务交换的特殊需要,可以对协议进行修改。例如,业务参与方可以在一个合同中对从国内厂家购买产品的消息交换进行说明,在另一个合同中对从国外厂家购买产品的消息交换进行说明。通过这样的选择,参与方可以同意遵守在贸易合作中达成的协议。多个协议也可以存在于从最初的发送方到最终的接收方等处理消息的各方之间。这些协议包括:

- a) 消息源站点的 MSH 与最终目的地的 MSH 之间的协议;
- b) 消息源站点的 MSH 与作为中间站点的 MSH 之间的协议;
- c) 消息最终目的地的 MSH 与作为中间站点的 MSH 之间的协议。

当然,中间点之间也存在协议。但是,源站点的 MSH 与最终目的地的 MSH 可以不知道这些协议。

符合本部分的 MSH 应遵守其本身与其他符合本部分的任一 MSH 之间进行通信的有效协议。从广义上说,这些协议被表示为 CPA。本部分标识了必须商定的信息,但没有规定用于制定和维护这些协议的方法或形式。在实际中应假设合同的实际内容已包含在初始化文件/配置文件、数据库或符合 GB/T 19256.2 的 XML 文档中。

4.2 运作策略和限制

从逻辑上讲,ebMS 是一个或多个业务应用与一个通信服务之间的一个服务。这就要求对业务应用与 MSH 之间抽象的服务接口进行定义。本部分认可该接口,但没有对其进行定义。本部分以后的版本将对该服务接口结构进行规定。

本部分定义了两个通信协议的绑定,且规定 MSH 独立于任一通信协议。由于早期工作集中在用

于传输的 HTTP 协议上,因此本部分没有优先考虑该协议。当然也可以使用其他协议,并且本部分以后的版本中可能给出与这些协议相关的详细说明。

ebMS 依赖于外部的配置信息。这些信息可通过所定义的业务过程或贸易伙伴协议来确定。获取这些数据的目的是为了在 CPP 或 CPA 中使用。GB/T 19256.2 对构成这些协议的信息进行了定义。ebXML 体系结构定义了基础设施的构件和 ebMS 之间的关系。就 MSH 而言,组成 CPP/CPA 的信息可用于支持正常的运作。然而,由特定的 MSH 实施系统所使用的方法并不要求存在单独的 CPA 实例。CPA 被表示为一个 XML 文档。某些实施系统可以选择增加并使用一个含有 CPA 信息的数据库。本部分没有规定如何获得、存储或使用 CPA 信息,只是说明必须提供用于 MSH 的特定信息项以获得成功的运作。

4.3 运作方式

本部分没有规定如何在 ebXML 总框架中安装 MSH。假定某些 MSH 实施系统并不实现本部分中已定义的所有功能。例如,一些贸易伙伴并不要求可靠的消息传输服务,因此在它们的 MSH 中不存在可靠消息传输能力。但是,所有的 MSH 实施系统应符合关于特定实施系统所支持功能的规范,并可以给出对所请求但不支持的功能的错误提示。关于 MSH 实施系统的文档应在实施系统中标识出没有满足要求的所有 ebMS 特征。

从概念上说,ebMS 可以分解为如下三个部分:

- a) 抽象的服务接口;
- b) MSH 提供的功能;
- c) 对底层传输服务的映射。

图 1 描述了存在于 ebMS 体系结构的可能的实施系统中的功能模块的逻辑排列。这些模块按照一种指明它们之间的内在关系和从属性的方式进行排列。

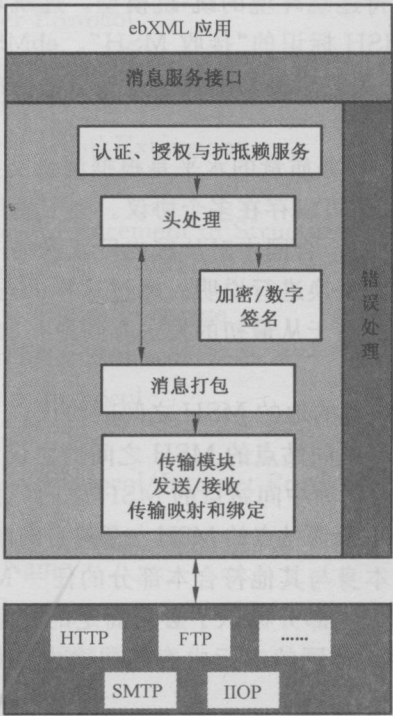


图 1 ebXML MSH 构件之间的典型关系

头处理(Header Processing)——使用下列应用程序生成一个 ebXML 消息的消息头元素。这些应用程序包括:经过消息服务接口,管理消息的 CPA 信息和生成如数字签名、时戳以及唯一标识符的信息。

头解析(Header Parsing)——把接收到的 ebXML Header 元素中的信息提取或转化为适用于 MSH 实施系统处理的一种形式。

安全服务(Security Services)——包括数字签名的生成与确认、加密、认证和认可。这些服务可由包括头处理和头解析构件的其他 MSH 构件使用。

可靠消息传输服务(Reliable Messaging Services)——处理 ebXML 消息的传输和确认。该服务包括处理可靠传输消息的持续时间、重试、错误通知和确认。

消息打包(Message Packaging)——将 ebXML 消息(ebXML Header 元素与负载)封装在符合[SOAPAttach]的容器中。

错误处理(Error Handling)——对 MSH 或应用程序处理消息期间遇到的错误的报告进行处理。

消息服务接口(Message Service Interface)——一个抽象的服务接口。应用程序使用该接口与发送和接收消息的 MSH 相互作用,MSH 使用该接口与处理接收消息(传输模块)的应用程序相连接。

5 核心功能

5.1 带有 SOAP 的 ebXML

本部分定义了了在 SOAP Envelope(信封)内的修饰命名空间的 SOAP Header 和 Body 元素扩展集合。这些元素打包在 MIME 多部分中,以使负载或附件与 SOAP 扩展元素包含在一起。通常,主要在以下情况中使用单个 ebXML SOAP 扩展元素:

- 可以使用不同的软件构件来生成 ebXML SOAP 扩展元素;
- ebXML SOAP 扩展元素不总是存在;
- ebXML SOAP 扩展元素中包含的数据可以与其他 ebXML SOAP 扩展元素分别进行数字签名。

5.1.1 打包规范

ebXML 消息是一个独立于通信协议的 MIME/Multipart 消息信封,在结构上符合[SOAPAttach]规范,也称其为消息包。

在消息包中包含两个逻辑 MIME 部分:

- 第一个 MIME 部分称为头容器,它包含符合 SOAP1.1 的消息,该 XML 文档在本部分的其余章节被称为 SOAP 消息;
- 零个或多个称为负载容器的附加 MIME 部分,包括应用层负载。

图 2 描述了 ebXML 消息的一般结构和组成。

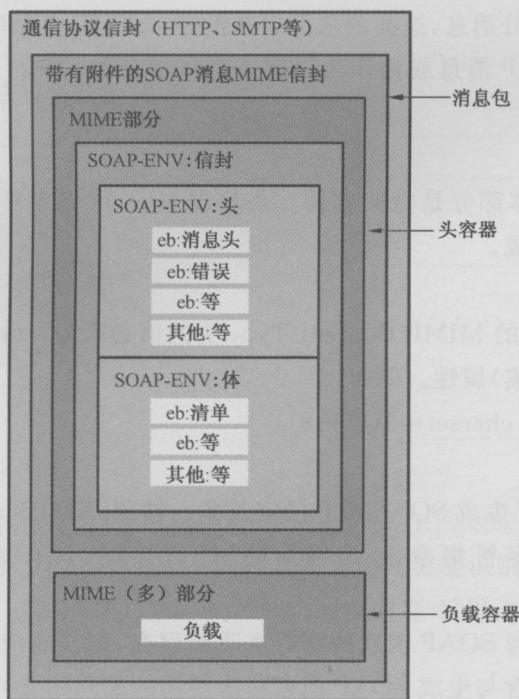


图 2 ebXML 消息结构

SOAP 消息是一个包含 SOAP Envelope 元素的 XML 文档。SOAP Envelope 元素是表示 SOAP 消息的 XML 文档的根元素。SOAP Envelope 元素由以下元素组成：

- a) 一个 SOAP Header 元素：用于增加 SOAP 消息特征的一种通用机制，包括特定的 ebXML 头元素。
- b) 一个 SOAP Body 元素：与消息负载部分相关的、由 MSH 控制的数据和信息的一个容器。

5.1.1.1 SOAP 结构一致性

ebXML 消息打包符合如下规范：

- a) 简单对象访问协议(SOAP)规范 1.1[SOAP]；
- b) 带有附件的 SOAP 消息规范[SOAPAttach]。

在 SOAP 消息中带有 ebXML 头并不意味着 ebXML 重载现有的 SOAP 语义，而是将基于 SOAP 的 ebXML 语义直接映射到 SOAP 语义中。

5.1.1.2 消息包

消息包的所有 MIME 头元素都符合[SOAPAttach]。另外，在消息包中的 Content-Type(<内容-类型>)MIME 头还包含一个 type(类型)属性，该属性与包含 SOAP 消息文档的 MIME 体部分的 MIME 媒体类型相匹配。按照[SOAP]规范，SOAP 消息的 MIME 媒体类型的值为“text/xml”。

建议初始头包括一个按照 MIME[RFC2045]构建的 Content-ID(<内容-标识>)MIME 头。对于 Multipart/Related(<多部分/相关>)的媒体类型，除了所要求的参数之外，还应给出 start(<开始>)参数(在[RFC2387]的 MIME Multipart/Related 中是可选的)。这样就可以进行更健壮的错误检查。下例是一个用于 Multipart/Related 消息包的 MIME 头：

```
Content-Type: multipart/related; type="text/xml"; boundary="boundaryValue";
start=messagepackage-123@example.com
```

```
--boundaryValue
```

```
Content-ID: messagepackage-123@example.com
```

实施系统必须支持非多部分消息，这类消息可以在没有 ebXML 负载时存在。没有负载的 ebXML 消息可以作为一个普通的 SOAP 消息或是作为只有一个体部分的、符合[SOAPAttach]的多部分消息进行发送。

5.1.1.3 头容器

在本部分中，消息包的根体部分是指头容器。头容器是一个 MIME 体部分，由一个在[SOAPAttach]中定义的 SOAP 消息组成。

5.1.1.3.1 Content-Type

根据[SOAP]规范，头容器的 MIME Content-Type 头的值必须为“text/xml”。在 Content-Type 头中可以包含一个 charset(字符集)属性。例如：

```
Content-Type: text/xml; charset="UTF-8"
```

5.1.1.3.2 charset 属性

MIME charset 属性标识了生成 SOAP 消息的字符集。按照[XMLMedia]中关于 XML 的规定，该属性的语义以 text/xml 的“字符集参数/编码策略”进行描述，其有效值列表参见 <http://www.iana.org/>。

如果 MIME charset 属性与 SOAP 消息的编码声明都存在，则二者应是等同的。如果给出 MIME charset 属性，则它一定不能包含与生成 SOAP 消息时所使用的编码相互冲突的值。

为了获得最大的交互能力，建议在对本文档编码时使用 UTF-8 规范[UTF-8]。由于对媒体类型定

义的处理规则源于[XMLMedia]的 text/xml,因此 MIME 属性没有缺省值。

5.1.1.3.3 头容器范例

下列片断给出了头容器的一个例子:

```
Content-ID: <messagepackage-123@example.com>      ---| Header
Content-Type: text/xml; charset="UTF-8"             |
                                                    |
<SOAP:Envelope                                     --| SOAP Message|
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"> |
  <SOAP:Header>                                     |
  ...                                               |
  </SOAP:Header>                                    |
  <SOAP:Body>                                       |
  ...                                               |
  </SOAP:Body>                                     |
  </SOAP:Envelope>                                --|
                                                    |
--boundaryValue                                     ---|
```

5.1.1.4 负载容器

在符合[SOAPAttach]的消息包中,可以存在零个或多个负载容器。

如果消息包中包含一个应用负载,则应用负载应包含在一个负载容器中。

如果消息包中没有应用负载,则负载容器一定不存在。

每一个负载容器的内容必须在 SOAP Body 中的 ebXML Manifest(清单)元素中进行标识(见 5.2.2)。

本部分对应用负载的结构或内容没有任何规定,也没有任何限制。负载可以是简单的无格式文本对象或是复杂的嵌套的多部分对象。规定负载对象的结构与组成,是定义业务过程或定义使用 ebMS 的信息交换机构的权利。

5.1.1.4.1 负载容器范例

下列部分是一个负载容器和负载的例子:

```
Content-ID: <domainname.example.com>      -----| ebXML MIME|
Content-Type: application/xml              -----|
                                                    |
<Invoice>                                  -----|
  <Invoicedata>                             | Payload   |
  ...                                       |          |
  </Invoicedata>                           |          |
</Invoice>                                -----| Container
```

5.1.1.5 其他 MIME 参数

本部分中描述的任何 MIME 部分都可以包含符合 MIME[RFC2045]规范的其他 MIME 头。实施系统可以忽略本部分中没有定义的 MIME 头,同时必须忽略其无法识别的 MIME 头。

例如,一个实施系统可能包含消息的 content-length(<内容-长度>)。然而带有 content-length 的消息的接收方可以将其忽略。

5.1.1.6 报告 MIME 错误

如果在消息包中检测到一个 MIME 错误,则必须按照[SOAPAttach]中的规定进行报告。

5.1.2 XML 序言

如果 SOAP 消息的 XML 序言存在,则它可以包括一个 XML 声明。对于 XML 序言,本部分没有增加新的注释或处理指令。例如:

```
Content-Type: text/xml; charset="UTF-8"
<? xml version="1.0" encoding="UTF-8"? >
```

5.1.2.1 XML 声明

在 SOAP 消息中可以存在 XML 声明。如果存在,则 XML 声明必须包括 GB/T 18793—2002 所要求的版本说明,并可以包括一个编码声明。下面描述的语义必须由 ebMS 来实施。

5.1.2.2 编码声明

如果编码声明与头容器的 MIME 字符集同时存在,则 SOAP 消息的 XML 序言应包括编码声明,且该声明应等同于头容器的 MIME Content-Type 的 charset 属性(见 5.1.1.3)。

如果给出了编码声明,则编码声明一定不包括与生成 SOAP 消息时所使用的编码相互冲突的值。对 SOAP 消息进行编码时建议使用[UTF-8]。

如果使用 GB/T 18793—2002 中规定的规则 XML 处理器不能确定字符编码,则应在 ebXML SOAP 头文档中给出 XML 声明及其包括的编码声明。

注:按照 GB/T 18793—2002,XML 文档中不要求包括编码声明。

5.1.3 ebXML SOAP Envelope 扩展

按照[SOAP]规范,所有的扩展元素的内容需要命名空间来修饰。在本部分中定义的所有 ebXML SOAP 扩展元素内容是由 5.1.2.2 中定义的 ebXML SOAP Envelope 扩展命名空间修饰的命名空间。

用于 ebXML SOAP 扩展的命名空间声明(xmlns 伪属性)可能包含在 SOAP Envelope、Header 或 Body 元素中,或直接包含在每一个 ebXML SOAP 扩展元素中。

5.1.3.1 命名空间伪属性

用于 ebXML SOAP Envelope 扩展的命名空间声明(xmlns 伪属性)(见[XMLNS])有一个必需值:
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd

5.1.3.2 xsi:schemaLocation 属性

下面的 SOAP 命名空间 <http://schemas.xmlsoap.org/soap/envelope/> 指的是 W3C XML 模式规范。OASIS 的 ebXML 消息传输技术委员会提供了一个 SOAP 模式版本。该 SOAP 模式版本与如下的 XML 模式规范[XMLSchema]的 W3C 推荐版本等效:

<http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd>

建议所有 ebXML MSH 实施系统应在 SOAP Envelope 元素中包括修饰 XMLSchema(XML 模式)实例命名空间的 schemaLocation(模式位置)属性,以便指出用来确认文档的、验证分析程序的模式文档位置。如果不包括 schemaLocation 属性,则可能阻碍所接收消息的 XML 模式确认。

例如:

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
  http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd">
```

另外,ebXML SOAP Header 和 Body 扩展元素内容可以受相似的限定,以便对确认分析程序找到模式文档位置进行标识,该模式文档包括修饰 ebXML 命名空间的 SOAP 扩展元素的定义。在[XMLSchema]的 W3C 推荐版本中,对 ebXML SOAP 扩展元素模式进行了定义(见附录 A)。修饰 XMLSchema 实例的命名空间的 schemaLocation 属性应包括一个从 ebXML SOAP Envelope 扩展命名空间到同一个元素中的模式文档的映射。该元素对 ebXML Envelope 扩展命名空间进行了声明。

在上面 5.1.3.1 中所描述的命名空间的 schemaLocation 为:

http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd

建议使用独立的 schemaLocation 属性,从而使不能正确使用 schemaLocation 属性来解决多个命名空间的模式的工具能够验证 ebXML SOAP 消息。例如:

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd">
  <SOAP:Header
    xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
    xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
      http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
    <eb:MessageHeader...>
    ...
  </eb:MessageHeader>
</SOAP:Header>
  <SOAP:Body xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
    header-2_0.xsd"
    xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2
      _0.xsd
        http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2 _ 0.
        xsd">
    <eb:Manifest eb:version="2.0">
    ...
  </eb:Manifest>
</SOAP:Body>
</SOAP:Envelope>
```

5.1.3.3 SOAP Header 元素

SOAP Header 元素是 SOAP Envelope 元素的第一个子元素。该元素必须有一个命名空间修饰符,该修饰符与命名空间“<http://schemas.xmlsoap.org/soap/envelope/>”的 SOAP Envelope 命名空间声明相匹配。

5.1.3.4 SOAP Body 元素

SOAP Body 元素是 SOAP Envelope 元素的第二个子元素。它必须有一个命名空间修饰符,该修饰符与命名空间“<http://schemas.xmlsoap.org/soap/envelope/>”的 SOAP Envelope 命名空间声明相匹配。

5.1.3.5 ebXML SOAP 扩展

ebXML 消息可扩展为含有下列主要扩展元素的 SOAP 消息:

5.1.3.5.1 SOAP Header 扩展元素

- a) MessageHeader——一个包括消息的路由信息(To/From 等)以及关于该消息的其他语境信息的必备型元素。
- b) SyncReply——指出下一个 SOAP 节点所要求的传输状态的元素。

5.1.3.5.2 SOAP Body 扩展元素

Manifest——一个指向负载容器或其他地方(如 web)中的数据的元素。该元素可被忽略。

5.1.3.5.3 ebXML 核心模块

a) 错误处理模块

ErrorList(错误列表)——包含报告上一个消息的错误列表的 SOAP 头元素。ErrorList 元素仅用于报告上一个消息的错误或警告,可以被忽略。

b) 安全模块

Signature(签名)——一个包含数字签名的元素。该数字签名符合用于对与消息相关的数据进行签名的[XMLDSIG]。该元素可以被忽略。

5.1.3.6 #通配符元素内容

按照模式中所指的那样,某些 ebXML SOAP 扩展元素允许通过扩展增加外来修饰命名空间的元素内容。按照[XMLNS],扩展元素内容必须由命名空间来修饰,同时也必须属于外来命名空间。外来命名空间是一个非 http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd 的命名空间。当扩展用于专用扩展或将来对协议进行扩充时,可以使用通配符元素。

MSH 实施系统可以忽略修饰命名空间的元素及其内容。

5.1.3.7 id 属性

本部分定义的每一个 ebXML SOAP 扩展元素都有一个 id(标识)属性,该属性是一个 XML ID。通过增加 XML ID 可以提供唯一标识 SOAP 消息中元素的能力。因为可以通过在 Reference(参考)元素中规定一个“#<idvalue>”的 URI 来确定是否包含或排除单个 ebXML SOAP 扩展元素,所以当将数字签名应用到 ebXML SOAP 消息中时可以使用该属性。

5.1.3.8 version 属性

必备型属性 version(版本)指出了 ebXML SOAP Header 扩展符合的本部分的版本。该属性的目的是提供将来更改版本的能力。为保持一致,本部分定义的任何 SOAP 扩展元素的所有版本属性值必须为“2.0”。ebXML 消息可以包括值不为“2.0”的 SOAP 头扩展元素。符合本部分的实施系统接收到非“2.0”版本修饰的 ebXML SOAP 扩展消息,如果能够识别已标识的版本并有能力处理,则该实施系统就可以处理该消息;如果不能识别出已标识的版本,则它必须作出错误应答。version 属性必须是由上面定义的 ebXML SOAP Envelope 扩展命名空间修饰的。

只有当必须从语义上改变元素且不能等待本部分的下一个版本发布的极端情况下,才支持在同一个 ebXML SOAP 文档中使用多个版本的 ebXML SOAP 扩展元素。

5.1.3.9 SOAP mustUnderstand 属性

关于 SOAP Header 扩展的必备型属性 SOAP mustUnderstand,即适用于 SOAP 命名空间(<http://schemas.xmlsoap.org/soap/envelope/>)的命名空间,指出按照[SOAP]元素内容必须被接收过程所理解或消息必须被拒绝。该属性值为“1(真)”表示元素必须被理解或拒绝;该属性值为“0(假)”,即缺省值,表示如果不理解该元素就可将其忽略。

5.1.3.10 ebXML "Next MSH" actor URI

当 URI(<urn:oasis:names:tc:ebxml-msg:actor:nextMSH>)在 SOAP actor(执行者)属性值的语境中被使用时,应被解释为扮演符合本部分 ebXML MSH 实例的角色中的一个实体。

已建立的 actor 的 URI 使得非 ebXML MSH 节点的 SOAP 节点加入到 ebXML 消息的消息路径成为可能。例如,经过数字签名或对消息加密的 SOAP 节点。

所有的 ebXML MSH 节点必须扮演该角色。

5.1.3.11 ebXML "To Party MSH" actor URI

当 URI(<urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH>)在 SOAP actor 属性值的语境中被使用时,应被解释为符合本部分 ebXML MSH 节点的一个实例。该实例扮演在同一个消息的 Message-Header/ To/PartyId 元素中已被标识的参与方角色。可以安排 ebXML MSH 扮演该角色。但是如何安排不在本部分范围之内。

ebXML 消息的最终目的地的 MSH 除了扮演根据 SOAP 定义的缺省角色外,还必须扮演接收方 MSH 的 actor 的 URI 角色。

5.2 核心扩展元素

5.2.1 MessageHeader 元素

在所有 ebXML 消息中,MessageHeader(消息头)元素都是必需的。它必须作为 SOAP Header 元素的一个子元素存在。

MessageHeader 元素是一个复合元素,它包括以下从属元素:

- a) 一个 id 属性(详见 5.1.3.7);
- b) 一个 version 属性(详见 5.1.3.8);
- c) 一个值为“1”的 SOAP mustUnderstand 属性(详见 5.1.3.9);
- d) From 元素;
- e) To 元素;
- f) CPAId(协同规程协议标识)元素;
- g) ConversationId(会话标识)元素;
- h) Service(服务)元素;
- i) Action(行为)元素;
- j) MessageData(消息数据)元素;
- k) DuplicateElimination(删除重复)元素;
- l) Description(描述)元素。

5.2.1.1 From 元素和 To 元素

必备型元素 From 标识发起消息的参与方,而必备型元素 To 标识预定接收消息的参与方。From 和 To 元素都包括诸如 DUNS 码的逻辑标识符,或包括暗含诸如 eMail 地址的物理地址标识符。

每一个 From 元素与 To 元素都包括:

- a) PartyId(身份标识)元素——出现一次或多次;
- b) Role(角色)元素——出现零次或一次。

如果 From 或 To 元素包括多个 PartyId 元素,则所有列表成员都必须标识同一机构。除非 type 属性值参考了多种标识机制,否则在 From 或 To 元素包含的 PartyId 元素列表中,任何给定的 type 属性值都是唯一的。

注:当参与方之间的消息传输涉及多个中间点时,该机制特别有用。通常,发送方应给出支持中间点和目的地的所有已知领域内的标识,以确定具体标识机制的优先权。

From 元素与 To 元素包括零个或一个 Role 子元素。如果该子元素存在,则应在 PartyId 子元素之后。

5.2.1.1.1 PartyId 元素

PartyId 元素只有一个 type 属性,其内容为一个字符串值。type 属性指出了 PartyId 元素内容中所属字符串的名称领域。所有参与方必须相互商定并理解 type 属性值。建议 type 值为一个 URI。进一步建议从 EDIRA(ISO 6523)、GB/T 14805 或 ANSI ASC X12105 注册系统中获取这些值。

如果 PartyId 的 type 属性不存在,则 PartyId 元素的内容必须是一个 URI[RFC2396]。否则接收 MSH 应报告一个错误(见 5.3.1.5),在报告中应将 errorCode(错误代码)设置为 Inconsistent,将 severity(严重性)设置为 Error。强烈建议 PartyId 元素的内容为一个 URI。

5.2.1.1.2 Role 元素

Role 元素标识了发送(当作为 From 元素的子元素存在时)和/或接收(当作为 To 元素的子元素存在时)消息的参与方的授权角色(fromAuthorizedRole 或 toAuthorizedRole)。Role 元素的值是 CPA 中规定的一个非空字符串。

注:Role 应尽可能定义为一个 URI。例如:<http://rosettanet.org/roles/buyer>。

以下部分说明了 From 元素和 To 元素的用法。

```
<eb:From>
  <eb:PartyId eb:type="urn:duns">123456789</eb:PartyId>
  <eb:PartyId eb:type="SCAC">RDWY</eb:PartyId>
  <eb:Role>http://rosettanet.org/roles/Buyer</eb:Role>
</eb:From>
<eb:To>
  <eb:PartyId>mailto:joe@example.com</eb:PartyId>
  <eb:Role>http://rosettanet.org/roles/Seller</eb:Role>
</eb:To>
```

5.2.1.2 CPAId 元素

必备型元素 CPAId 是对参与方的消息交换进行管理的参数进行标识的一个字符串。在考虑消息发送方的情况下,消息接收方必须能够将 CPAId 解析成一套单独的参数。

在参与方商定的命名空间内,CPAId 元素值必须是唯一的。它可能是 From 元素与 To 元素的 PartyId 值的连接,以某一方的 Internet 域名作为前缀的 URI 或是由其他的命名或注册服务提供和管理的命名空间。建议 CPAId 值为一个 URI。

CPAId 可参考在 GB/T 19256.2 中定义的 CPA 实例。下面为 CPAId 元素的一个例子:

```
<eb:CPAId>http://example.com/cpas/ourcpawithyou.xml</eb:CPAId>
```

当用 CPAId 元素标识消息传输参数时,消息传输参数由 CPA 中的适当元素来确定。

如果接收方确定消息同 CPA 相互冲突,但是由于本部分没有对该冲突的相应处理进行规定,因此,发送方只有在提前知道接收方具有处理这种冲突能力的情况下,才可生成这样的消息。

如果接收 MSH 检查到不一致,则它必须报告一个 errorCode 为 Inconsistent 和 severity 为 Error 的消息。如果不能识别 CPAId,则必须报告 errorCode 为 NotRecognized 和 severity 为 Error 的消息。

5.2.1.3 ConversationId 元素

必备型元素 ConversationId 是一个对组成两个参与方会话的相关消息集合进行标识的字符串。它在规定的 CPAId 语境内必须是唯一的。发起会话的参与方确定 ConversationId 元素值,该值在所有与会话有关的消息中都能得到反映。

ConversationId 元素使消息接收方能够识别出生成会话中早期消息或对其进行处理的应用程序或过程的实例。该元素对于会话中的所有消息保持不变。

用于 ConversationId 的值与实施系统相关。ConversationId 元素的例子如下:

```
<eb:ConversationId>20001209-133003-28572</eb:ConversationId>
```

注:实施系统可以自由选择如何标识和存储一个与特定会话相关的会话状态,并且应为其标识模式和另一个实施系统产生的 ConversationId 元素之间的映射提供便利。

5.2.1.4 Service 元素

必备型元素 Service 标识了处理消息的服务,并由服务的设计者对其进行规定。服务的设计者可以是:

- a) 一个标准机构;
- b) 个人或公司。

注:在 ebXML 业务过程模型的语境中,行为就是基于 GB/T 19256.6 中最低可能的角色(请求或应答角色),服务就是参与方中被授权角色相关行为的集合。

下面为 Service 元素的一个例子:

```
<eb:Service>urn:services:SupplierOrderProcessing</eb:Service>
```

注:本部分保留使用 Service 元素中以命名空间 urn:oasis:names:tc:ebxml-msg:service 开始的 URI。

Service 元素只有一个 type 属性。