ebXML　解説書

第 6 部

# ebXML 通信仕様

平成 14 年 3 月

(財)日本情報処理開発協会 電子商取引推進センター

この解説書は、経済産業省委託「平成 13 年度 精算・調達・運用支援統合情報システムに関する調査研究」事業の成果です。

はじめに

1999 年 11 月に、国際標準 EDI（UN／EDIFACT）の利用グループの支援を受けた国連 CEFACT（Center for Trade Facilitation and eBusiness）と、先進的 IT ベンダーのコンソーシアムである OASIS (Organisation for Advanced Structured Information Standards)の協業で始められた ebXML イニシャチブは、2001 年 5 月、今後の電子ビジネスコラボレーション実現のフレームワークとなる一連の仕様（ebXML 仕様）の第 1 版を完成し公表した。

ebXML 仕様は、従来のレガシーEDI や WEB－EDI を XML 化するに留まらず、取引企業同士のそれぞれのアプリケーションが、情報交換により合意されたビジネスプロセスを遂行してビジネス目標を達成する、すなわち電子ビジネスコラボレーションを実現させるために必要な標準仕様を定めている。

今後、当該標準仕様は、IT ベンダーの戦略的製品やサービスに取り入れられるとともに、ユーザー業界においてはビジネスプロセス改善の仕組みに採用されて行くことが期待されている。

(財)日本情報処理開発協会では経済産業省の委託を受けて、2001 年 5 月に公表された ebXML 仕様を中心に、電子商取引推進協議会の平成 13 年度 XML／EDI 標準化専門委員会の討議結果を反映し、次の 6 部からなる解説書を作成した。

第1部　ebXML 概要
第2部　ebXML ビジネスプロセス
第3部　ebXML 情報構成要素
第4部　ebXML レジストリ・リポジトリ
第5部　ebXML 交換協定
第6部　ebXML 通信仕様

なお、ebXML 仕様は、2001 年 5 月以降、第 2 章－第 3 章関連は UN／CEFACT、第 4 章－第 6 章関連は OASIS が仕様の改訂・保守を継続しており、ebXML 仕様の実装においては該当組織より発表されている最新版の仕様を参照されることを推奨する。

平成 14 年 3 月

財団法人日本情報処理開発協会

電子商取引推進センター

第 6 部 ebXML 通信仕様　序文

「ebXML 通信仕様」は、ebXML イニシャチブにより作成されたメッセージ取り扱い仕様を掲載する。

<参照 ebXML 仕様書>

Message Service Specification V1.0　　2001 年 5 月 11 日

# メッセージ取扱サービス仕様

**ebXML トランスポート・ルーティングおよびパッケージング**

**第1.0版**

2001 年 5 月 11 日

技術検証 ECOM XML/EDI 標準化専門委員会

# 1 本書の位置付け

本書は、電子ビジネスコミュニティのための ebXML 草案を定める。本書は何の制限もなく自由に配布することができる。

本書の形式は、Microsoft Word 2000 形式に変換したインターネット・ソサイアティーの標準 RFC 形式に基づいている。

メモ: 本書の実装者は、ebXML のウェブサイト(http://www.ebxml.org)を参照し、本書の現在の位置付け並びに改訂状況を参照のこと。

*仕様*

本技術仕様書は、ebXML 総会で承認されている。

本書は、ebXML 要件文書の要件を満たしている。

本バージョン

http://www.ebxml.org/specs/ebMS.pdf

最新バージョン

http://www.ebxml.org/specs/ebMS.pdf

# 2 ebXML 参加者

プロジェクトチームの電子メールリスト、電話による協議、および実際の会議を通じ、本書にアイデアを提供してくれたトランスポート・ルーティングおよびパッケージングプロジェクトチームのメンバの氏名を以下に記し、謝意を表す。

Ralph Berwanger – bTrade.com
Jonathan Borden – XMTP作成者
Jon Bosak – Sun Microsystems
Marc Breissinger – webMethods
Dick Brooks – Group 8760
Doug Bunting – Ariba
David Burdett – Commerce One
David Craft – VerticalNet
Philippe De Smedt – Viquity
Lawrence Ding – WorldSpan
Rik Drummond – Drummond Group
Andrew Eisenberg – Progress Software
Colleen Evans –Progress / Sonic Software
David Fischer – Drummond Group
Christopher Ferris – Sun Microsystems
Robert Fox – Softshare
Brian Gibb – Sterling Commerce
Maryann Hondo – IBM
Jim Hughes – Fujitsu
John Ibbotson – IBM
Ian Jones – British Telecommunications
Ravi Kacker – Kraft Foods
Henry Lowe – OMG
Jim McCarthy – webXI
Bob Miller – GXS
Dale Moberg – Sterling Commerce
Joel Munter – Intel
中垣 俊平 – NEC Corporation
Farrukh Najmi – Sun Microsystems
越智 昭 (Ochi Akira) – Fujitsu
Martin Sachs, IBM
Saikat Saha – Commerce One
島村正義 – Fujitsu
Prakash Sinha – Netfish Technologies
Rich Salz – Zolera Systems
Tae Joon Song – eSum Technologies, Inc.
Kathy Spector – Extricity
Nikola Stojanovic – Encoda Systems, Inc.
David Turner - Microsoft
Gordon Van Huizen – Progress Software
Martha Warfelt – DaimlerChrysler Corporation
Prasad Yendluri – Web Methods

# 3 目次

# 4 はじめに

本書は、XML に基づくメッセージを通して、企業がその規模や地域に関係なく取引を行うことが可能な、単一の世界的な電子商取引市場の構築というビジョンを実現する、仕様の一つである。一連の仕様により、モジュールで、しかも完全な電子ビジネスのフレームワークを実現できる。

本書では、通信プロトコルに依存せず、電子ビジネスメッセージを交換する方法を定義することに焦点を当てており、信頼度の高い、安全なビジネス情報の配信をサポートする、具体的なエンベロープ構造が定められている。さらに、本書では、ebXML 準拠のメッセージにあらゆる形式種類の搬送内容を含められるようにする、柔軟性の高いエンベロープ手法も定めている。この汎用性により、従来の構文(UN/EDIFACT、ASC X12、または HL7)を利用する電子ビジネスシステムであっても、新技術のユーザとともに、ebXML 基盤の利点を利用することが可能となる。

## 4.1 本書の総括

本書は、安全で信頼性の高い二者間のメッセージ交換を保証する ebXML メッセージ取扱サービスプロトコルを定めるもので、これには以下の説明が含まれている。

- 搬送内容データをパッケージして二者間で転送するのに使用される ebXML メッセージ構造
- データ通信プロトコルでメッセージを送受信するメッセージ取扱サービスハンドラの動作

本書の付録では、HTTP および SMTP で使用する方法を説明しているが、本書は、使用される搬送内容および通信プロトコルに依存しない。

本書は、以下の項目に沿って構成されている。

- **パッケージング仕様** – HTTP や SMTP などの通信プロトコルを使用して送信することが可能な形式で、ebXML および関連部品をパッケージする方法を説明する。(第7節)
- **ebXML SOAP 拡張** – ebXML メッセージの生成または処理のために ebXML メッセージ取扱サービスに必要な情報の構造および組成の仕様について説明する。(第8節)
- **メッセージ取扱サービスハンドラのサービス** – ある 1 つのサービスが別のメッセージ取扱サービスハンドラ(MSH)や個々のメッセージの状態を明らかにできるようにする 2 つのサービスについて説明する。(第9節)
- **高信頼性メッセージング** – 高信頼性メッセージング機能は、2 つのメッセージ取扱サービスが、「高信頼性メッセージング」の 1 回のみの配信意味情報を使用して送信したメッセージを「高い信頼度」で交換することができるようにする、相互運用可能なプロトコルを定義する。(第10節)
- **エラー処理** – この節では、ebXML メッセージ取扱サービスが、検出したエラーを別の ebXML メッセージ取扱サービスハンドラに報告する方法について説明する。(第11節)
- **セキュリティ** – ebXML メッセージのセキュリティ意味情報の仕様を説明する。(第 12
- 節)

本書の付録には以下のものを含む。

- **付録 A スキーマ** – ebXML SOAP ***ヘッダ***および***ボディ***のための[XML スキーマ]が含まれている。
- **付録 B 通信プロトコルエンベロープマッピング** – ebXML メッセージ取扱サービス準拠のメッセージを HTTP および SMTP で転送する方法について説明する。

## 4.2 文書規約

*斜体*の用語は、ebXML 用語集[ebGLOSS]で定義している。***太字の斜体の***用語は要素または属性の内容を表している。`Courier` フォントの用語は、MIME コンポーネントに関連している。メモは Times New Roman フォントで示し、情報提供を目的とした非規範的事項である。

本書で「～しなければならない」、「～してはならない」、「～する必要がある」、「～すべきである」、「～すべきでない」、「～した方がよい」、「～しない方がよい」、「～を推奨する」、「～することができる」、「～を選択できる」などの語が使用されているときは、以下に引用されている通り、RFC 2119 の説明に従って解釈することとする。

*メモ: これらの語の強制力は、語が使用されている文書の必要度によって異なる。*

- *～しなければならない: 「～する必要がある」、「～すべきである」と同じく、本書に必要不可欠の事項であることを意味する。*
- *～してはならない: 「～すべきでない」と同じく、本書で禁止されている事項であることを意味する。*
- *～した方がよい: 「～を推奨する」と同じく、ある特定の状況下で特定の事項を無視できる正当な理由があるものの、異なる方法を選ぶ前に、その結果を完全に理解し、慎重に考慮しておかなければならないことを意味する。*
- *～しない方がよい: 「～は推奨されない」と同じく、ある特定の状況下では特定の動作が可能であったり、さらには有効であったりするものの、説明されている動作を実施する前に、その結果を完全に理解し、慎重に考慮しておかなければならないことを意味する。*
- *～することができる: 「～を選択できる」と同じく、説明されている事項が完全にオプションであることを意味する。ベンダは、特定の市場で必要とされている、または他のベンダは省いているものの、自社製品の強化につながると思われる、などの理由で、説明されている事項を含めることができる。特定のオプションが含まれていない実装は、機能性は低下しても、そのオプションが含まれている別の実装と相互運用できるようになっていなければならない。同様に、特定のオプションが含まれている実装は、そのオプションが含まれていない別の実装と相互運用できるようになっていなければならない(当然、オプションで提供される機能は除く)。*

## 4.3 対象

本書は、*ebXML メッセージ取扱サービス*を実装するソフトウェア開発者を対象としている。

## 4.4 警告および前提要件

本書は、読者が転送プロトコル、MIME、XML、SOAP、添付事項付き SOAP メッセージ、セキュリティ技術を理解していることを前提としている。

本書の例は、すべて非規範的なものである。仕様と例の間に矛盾が生じた場合は、仕様が優先される。

## 4.5 関連文書

ebXML イニシアチブの一部として、本書とは別に以下の関連仕様一式が作成されている。

- **ebXML メッセージ取扱サービス要件仕様**–メッセージ取扱サービスの要件を定めている。
- **ebXML テクニカルアーキテクチャ**– ebXML の全体的なテクニカルアーキテクチャを定めている。
- **ebXML テクニカルアーキテクチャセキュリティ仕様**– 予想されるリスクの一部を除去するのに必要なセキュリティの仕組みを定めている。
- **ebXML コラボレーションプロトコルプロフィールおよびコラボレーションプロトコル合意書仕様**- この仕様に準拠したメッセージを送信する前に、相手方当事者について知る必要のある情報を発見したり、合意したりできる方法を定めている。
- **ebXML レジストリ/リポジトリサービス仕様**– ebXML 環境のレジストリサービスを定めている。

# 5 設計目標

本書の設計目標は、メッセージ取扱サービスのワイヤ形式およびプロトコルを定め、XML に基づく中小企業および大企業の間の電子ビジネスをサポートすることにある。仕様は主に XML ベースの電子ビジネスをサポートするよう設計されているが、仕様作成者は XML 以外のビジネス情報の交換も完全にサポートするよう努めた。本書は、堅固さと優れた性能を加えることにより独自の価値を付加できる能力をベンダに残しつつ、低費用の解法を可能にする。トランスポート・ルーティングおよびパッケージングプロジェクトチームの意図は、この仕様を可能な限り容易で簡潔なものにすることにある。

本書で説明されている必須機能が ebXML 概念検証チームによって規範化され、本書の明瞭さ、正確さ、効率が保証されるように、あらゆる努力を払っている。

# 6 システム概要

本書では、ebXML 基盤の *ebXML メッセージ取扱サービス*コンポーネントを定義する。*ebXML メッセージ取扱サービス*は、HTTP や SMTP などの通信プロトコルによる ebXML メッセージの転送に使用するメッセージエンベロープおよびヘッダ文書のスキーマを定める。本書には、ebXML メッセージのパッケージング、交換、および処理を行うソフトウェアの開発に十分な詳細を含む。

*ebXML メッセージ取扱サービス*は、業界で広範に受け入れられ、W3C XML プロトコルコアワーキンググループによる作業の土台としての役割も果たす、基本的なシンプルオブジェクトアクセスプロトコル[SOAP]および添付事項付き SOAP メッセージ[SOAPATTACH]仕様へのレイヤ拡張として定義される。*ebXML メッセージ取扱サービス*には、SOAP や添付事項付き SOAP メッセージ仕様で提供されていない国際電子ビジネスをサポートするのに必要なセキュリティおよび信頼性に関する機能を用意する。

## 6.1 メッセージ取扱サービスの目的

*ebXML メッセージ取扱サービス*は、既存のさまざまなプロトコルを使用して取引当事者間でメッセージを転送するための、強固な基本機能を定義する。*ebXML メッセージ取扱サービス*は、メッセージの信頼性、永続性、セキュリティ、拡張性を可能にするよう体系化されている。

*ebXML メッセージ取扱サービス*は、電子ビジネスを可能にするために、強固でしかも低費用の解法が必要となるような環境に提供されるもので、ebXML の 4 つの「基盤」コンポーネントの 1 つである。その他の 3 つのコンポーネントとは、レジストリ/リポジトリ、コラボレーションプロトコルプロフィールおよびコラボレーションプロトコル合意書、ebXML テクニカルアーキテクチャ[ebTA]である。

## 6.2 メッセージ取扱サービスの概要

*ebXML メッセージ取扱サービス*は、概念的に(1)抽象サービスインタフェース、(2)メッセージ取扱サービスヘッダ(MSH)が提供する機能、(3)基本的な転送サービスへのマッピングという 3 つの部分に分けることができる。

図 6-1 は、実装可能な *ebXML メッセージ取扱サービスアーキテクチャ*の 1 つに含まれる機能モジュールの論理的な配置を示す。これらのモジュールは、モジュール間の相互関係や依存状態を示す方法で配置している。

- **ヘッダ処理** - *ebXML メッセージ*の SOAP ***ヘッダ***要素の作成には、メッセージ取扱サービスインタフェースを通じて引き渡されたアプリケーションからの入力、メッセージを制御する *コラボレーションプロトコル合意書*の情報([ebCPP]で定める CPA)、デジタル署名、タイムスタンプ、一意の識別子などの生成情報を使用する。
- **ヘッダパーシング** - 受け取った SOAP ***ヘッダ***または***ボディ***要素の情報を抽出し、MSH での処理に適した形式に変換する。
- **セキュリティサービス** - デジタル署名の作成と確認、証明、および許可を行う。これらのサービスは、ヘッダ処理やヘッダパーシングを含む、MSH の他のコンポーネントで使用される場合がある。
- **高信頼性メッセージングサービス** - ***OnceAndOnlyOnce*** の ***deliverySemantics*** で送信される ebXML メッセージの配信と受領通知を処理する。このサービスには、信頼性の高い配信に必要なメッセージの永続化、再試行、エラー通知、および受領通知を含む。
- **メッセージパッケージング** - 最終的に *ebXML メッセージ*(SOAP ***ヘッダ***または***ボディ***要素と搬送内容)を添付事項付き SOAP メッセージ[SOAPATTACH]コンテナにエンベロープ化する。
- **エラー処理** - このコンポーネントは、MSH やアプリケーションでメッセージを処理する際に生じたエラー報告を取り扱う。

- **メッセージ取扱サービスインタフェース** - メッセージを送受信するためにアプリケーションが MSH との相互運用に使用したり、受領したメッセージを処理するアプリケーションとのインタフェースに MSH が使用したりする、抽象サービスインタフェースである。

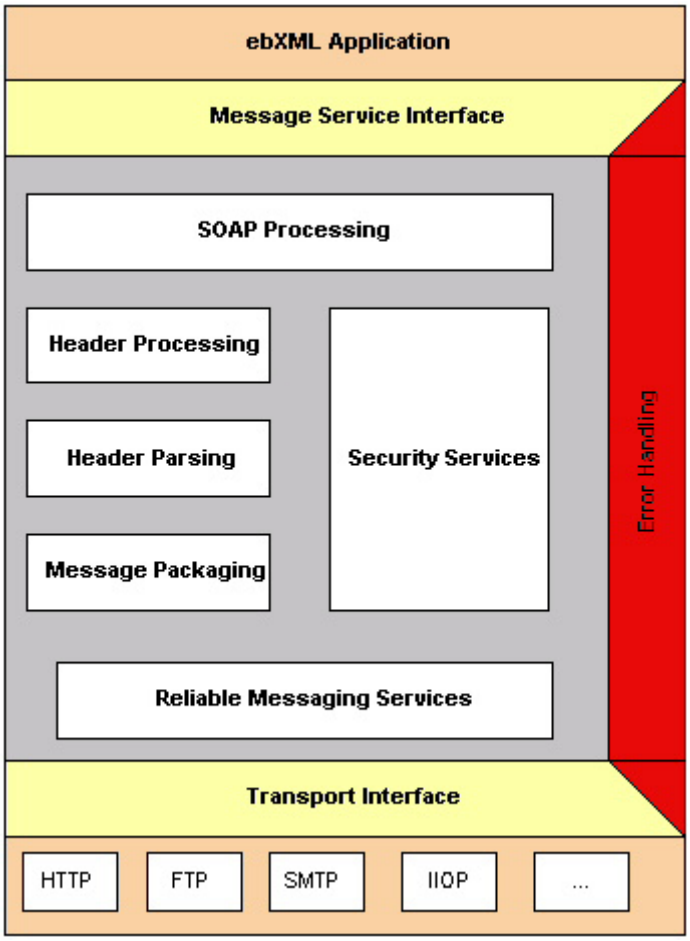



**図6-1 ebXML メッセージ取扱サービスハンドラコンポーネントの一般的な相互関係**

## 6.3 version 属性の使用

ebXML SOAP 拡張要素には、それぞれ ebXML メッセージ取扱サービス仕様のバージョンレベルに一致する version 属性があり、仕様全体を変更することなく、要素ごとに意味情報を変更することができる。

同じ ebXML SOAP 文書内で複数の ebXML SOAP 拡張要素のバージョンを使用することは可能である。しかし、ebXML メッセージ取扱サービス仕様の次のバージョンがリリースされるまで、要素の意味的な変更が必要となるような特別な場合を除いては ebXML SOAP 拡張要素のバージョンを使用すべきではない。

# 7 パッケージング仕様

## 7.1 はじめに

ebXML メッセージは、添付事項付き SOAP メッセージ[SOAPATTACH]仕様に従って並べられた、MIME/マルチパートメッセージプロトコルに依存しない通信プロトコルで、*メッセージパッケージ*と呼ぶ。

*メッセージパッケージ*には、以下の 2 つの論理的な MIME 部がある。

- SOAP1.1 準拠のメッセージ 1 つを含む、*ヘッダコンテナ*と呼ばれる MIME 部が 1 つ。本書では、以後この XML 文書を *SOAP メッセージ*と呼ぶ。
- アプリケーションレベルの搬送内容を含む、*搬送コンテナ*と呼ばれる MIME 部(パーツ数は任意)。

*SOAP メッセージ*は、SOAP ***エンベロープ***要素から成る XML 文書で、*SOAP メッセージ*であることを示す XML 文書の基本要素である。SOAP ***エンベロープ***の要素は以下のものから構成される。

- 1 つの SOAP ***ヘッダ***要素. これは、ebXML 固有のヘッダ要素を含む、*SOAP メッセージ*に機能を追加するための一般的な仕組みである。
- 1 つの SOAP ***ボディ***要素. これは、メッセージ取扱サービスハンドラ制御データと、メッセージの搬送内容部に関する情報のコンテナである。

ebXML メッセージの一般的な構造および構成を図 7-1 に示す。


**図 7-1 ebXML メッセージ構造**

### 7.1.1 SOAP 構造の適合性

*ebXML メッセージ*のパッケージングは、以下の仕様に従がわなければならない。

- シンプルオブジェクトアクセスプロトコル(SOAP) 1.1[SOAP]
- 添付事項付き SOAP メッセージ[SOAPATTACH]

*SOAP メッセージ*内で ebXML ヘッダを搬送することにより、ebXML が既存の SOAP 意味情報を取り消すわけではなく、SOAP 上の ebXML 意味情報は SOAP の意味情報に直接マッピングされる。

## 7.2 メッセージパッケージ

*メッセージパッケージ*の MIME ヘッダ要素は、すべて添付事項付き SOAP メッセージ[SOAPATTACH]の仕様に準拠していなければならない。また、*メッセージパッケージ*の MIME ヘッダ、`Content-Type` には、*SOAP メッセージ*文書を含む MIME ボディ部の MIME メディア種類と一致する `type` 属性が含まれていなければならない。*SOAP メッセージ*の MIME メディア種類の値は、[SOAP]仕様に従い、「`text/xml`」でなければならない。

基本部分には、[RFC2045]に準じた構造で MIME ヘッダ、`Content-ID` を含めること、そしてマルチパート/関連メディア種類に必須のパラメータに加え、常に `start` パラメータ([RFC2387]のオプション)を含めることを強く推奨する。これにより、エラーの検出がより確実になる。以下に例を示す。

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

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

## 7.3 ヘッダコンテナ

本書では、*メッセージパッケージ*の基本ボディ部を*ヘッダコンテナ*と呼びます。*ヘッダコンテナ*は、添付事項付き SOAP メッセージ[SOAPATTACH]仕様で定めている通り、1 つの *SOAP メッセージ*で構成しなければならない MIME ボディ部である。

### 7.3.1 Content-Type

*ヘッダコンテナ*の MIME `Content-Type` ヘッダには、[SOAP]仕様に従い「`text/xml`」の値がなければならない。`Content-Type` ヘッダには、「`charset`」属性を含めることができる。以下に例を示す。

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

#### 7.3.1.1 Charset 属性

MIME の `charset` 属性は、*SOAP メッセージ*の作成に使用する文字セットを識別する。この属性の意味情報は、[XMLMedia]で定める通り、`text/xml` の「文字セットのパラメータ/符号化について」で説明している。有効な値はhttp://www.iana.org/に示されている。

両方が示されている場合、MIME の `charset` 属性は、*SOAP メッセージ*の符号化宣言文と同値でなければならない。MIME の `charset` 属性を含める場合は、*SOAP メッセージ*作成の際に使用する符号化と一致した値でなければならない。

相互運用性を最大にするため、この文書を符号化する際は[UTF-8]を使用することを推奨する。
`text/xml` から派生したメディア種類に定義された処理規則[XMLMedia]により、この MIME 属性には既定値を設定していない。以下に例を示す。

```
    charset="UTF-8"
```

### 7.3.2 ヘッダコンテナ例

以下の記述は*ヘッダコンテナ*の例である。。

```
Content-ID: messagepackage-123@example.com                                    ---| ヘッダ
```

```
Content-Type: text/xml;                                                          |
              charset="UTF-8"                                                    |
                                                                                 |
<SOAP-ENV:Envelope                                                  --|SOAP メッセージ      |
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">     |                    |
  <SOAP-ENV:Header>                                                 |                    |
    …                                                               |                    |
  </SOAP-ENV:Header>                                                |                    |
  <SOAP-ENV:Body>                                                   |                    |
    …                                                               |                    |
  </SOAP-ENV:Body>                                                  |                    |
</SOAP-ENV:Envelope>                                                --|                    |
---boundaryValue                                                                  ---|
```

## 7.4 搬送コンテナ

添付事項付き SOAP メッセージ[SOAPATTACH]仕様に従い、*メッセージパッケージ*に任意で 1 つまたは複数の*搬送コンテナ*を含めることができる。

*メッセージパッケージ*にアプリケーション搬送内容を含める場合は、*搬送コンテナ*に含めなければならない。

*メッセージパッケージ*にアプリケーション搬送内容がない場合、*搬送コンテナ*を含めてはならない。

各*搬送コンテナ*の内容は、*SOAP **ボディ***内の ebXML メッセージの ***Manifest*** 要素で識別されなければならない(8.11項参照)。

ebXML メッセージ取扱サービス仕様には、アプリケーション搬送内容の構造または内容に関する規定や制限は一切ない。搬送内容は、単なるテキストオブジェクトでも、ネストされた複雑なマルチパートのオブジェクトでも差し支えない。搬送内容オブジェクトの構造および構成の仕様は、*ebXML メッセージ取扱サービス*を使用してビジネスプロセスや情報の交換を定めるプロジェクトに委ねられている。

### 7.4.1 搬送コンテナ例

以下の記述は*搬送コンテナ*と搬送内容の例である。

```
   Content-ID: <domainname.example.com>  -------------| ebXML MIME        |
   Content-Type: application/xml         -------------|                   |
                                                      |                   | 搬送内容
   <Invoice>                             -------------|                   | コンテナ
     <Invoicedata>                                    |  搬送内容         |
       …                                              |                   |
     </Invoicedata>                                   |                   |
   </Invoice>                            -------------|                   |
```

## 7.5 MIME 追加パラメータ

本書で説明されている MIME 部には、[RFC2045]の仕様に従い、MIME 追加ヘッダを含めることができる。実装では、本書で定められていない MIME ヘッダを無視することができる。認識されない MIME ヘッダは無視しなければならない。

例えば、実装ではメッセージに `content-length` を含めることができるが、`content-length` 付きのメッセージの受領者は、これを無視することができる。

## 7.6 MIME エラーの報告

*メッセージパッケージ*内で MIME エラーが検出された場合は、[SOAP]で定められている通り報告しなければならない。

# 8 ebXML SOAP 拡張

ebXML メッセージ取扱サービス仕様では、名前空間に適格な SOAP エンベロープ内 SOAP ***Header*** および ***Body*** 要素の拡張一式を定めている。以下の場合、通常、別の ebXML SOAP 拡張要素が使用される。

- ebXML SOAP 拡張要素の生成に、異なるソフトウェアコンポーネントが使用される可能性がある場合
- ebXML SOAP 拡張要素が常に含まれているとは限らない場合
- ebXML SOAP 拡張要素に含まれるデータに、他の ebXML SOAP 拡張要素とは別のデジタル署名が示されている可能性がある場合

## 8.1 XML Prolog

SOAP *メッセージ*の XML Prolog がある場合、これに XML 宣言文を含めることができる。本書では、XML Prolog に見られる追加コメントや処理命令は定義されない。以下に例を示す。

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

<?xml version="1.0" encoding="UTF-8"?>
```

### 8.1.1 XML 宣言文

XML 宣言文は、SOAP *メッセージ*に含めることができる。この場合は、XML 勧告[XML]: バージョン「1.0」で必要とされているバージョン仕様を含めなければならず、また符号化宣言文を含めることも可能である。以下で説明する意味情報は、準拠した *ebXML メッセージ取扱サービス*で実装しなければならない。

### 8.1.2 符号化宣言文

符号化宣言文と *ヘッダコンテナ* MIME 文字セットの両方が含まれている場合、SOAP *メッセージ*の XML Prolog には、*ヘッダコンテナ*の MIME `Content-Type` の `charset` 属性と同じ符号化宣言文が含まれていなければならない(7.3項参照)。

符号化宣言文がある場合、これに SOAP *メッセージ*作成の際に使用された符号化と一致しない値を含めることはできない。SOAP *メッセージ*を符号化する際は、UTF-8 を使用することを推奨する。

XML プロセッサが[XML]の 4.3.3 項で定めている規則を使用して符号化文字を決定できない場合、XML 宣言文とそれに含まれる符号化宣言文は、ebXML SOAP ***ヘッダ***文書で提供しなければならない。

注意: XML 仕様バージョン 1.0 [XML]に従い、XML 文書には符号化宣言文は必要ない。

## 8.2 ebXML SOAP エンベロープ拡張

[SOAP]仕様に従って、拡張要素の内容はすべて、適格な名前空間でなければならない。本書で定められている SOAP 拡張要素の内容は、8.2.1 項で定められている通り、すべて ebXML SOAP ***エンベロープ***拡張の名前空間に適格な名前空間でなければならない。

ebXML SOAP 拡張の名前空間宣言文(xmlns 擬似属性)は、SOAP ***Envelope***、***Header***、または ***Body*** 要素、もしくは直接それぞれの ebXML SOAP 拡張要素に含めることができる。

### 8.2.1 名前空間擬似属性

ebXML SOAP***エンベロープ***拡張の名前空間宣言文(***xmlns***擬似属性) ([XML Namespace]参照)には、「http://www.ebxml.org/namespace/messageHeader」の値が必要である。。

### 8.2.2 xsi:schemaLocation 属性

次の SOAP 名前空間は、W3C XML スキーマ仕様の初期草案に従ったスキーマに変わる。

```
http://schemas.xmlsoap.org/soap/envelope/
```

これは具体的に次の URI で明らかにされている。

```
http://www.w3.org/1999/XMLSchema
```

W3C XML スキーマ仕様[XMLSchema]は、2000 年 10 月 24 日をもって勧告候補の位置付けに進み、最近では、2001 年 3 月 30 日付けで提議勧告となっている。ツールのほとんどとは言わないまでも、その多くがスキーマの確認を支持しており、本書作成時点で入手可能な確認用の XML パーサは、XML スキーマ仕様[XMLSchema]の勧告候補の草案をサポートするよう設計されている。さらに、ebXML SOAP 拡張要素のスキーマは、XML スキーマ仕様[XMLSchema]の草案を使用して定義している(付録 A 参照)。

確認用パーサおよびさまざまなスキーマ確認ツールが適切に ebXML SOAP *メッセージ*を処理およびパースできるようにするため、ebXML TR&P チームは、XML スキーマ仕様[XMLSchema]の W3C 草案と同等の、更新された SOAP スキーマを採用する必要がある。ebXML MSH の実装では、XML スキーマインスタンスの名前空間に適格な ***schemaLocation*** 属性を ***Envelope*** 要素に含め、文書を有効とするために使用すべきスキーマ文書の場所を確認パーサに示すようにすることを強く推奨する。***schemaLocation*** 属性を含めなかった場合、*受信側 MSH* の実装は、受信したメッセージを確認できなくなる恐れがある。

以下に例を示す。

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
      http://ebxml.org/project_teams/transport/envelope.xsd" ...>
```

また同様に、ebXML SOAP ***Header*** および ***Body*** 拡張要素の内容も、確認サーバが ebXML の名前空間に適格な SOAP 拡張要素の定義が含まれているスキーマ文書を発見する場所を明らかにできるようになっていなければならない。従って、XML スキーマインスタンスの名前空間に適格な ***schemaLocation*** 属性では、ebXML SOAP ***Envelope*** 拡張名前空間を宣言する同じ要素内のスキーマ文書に、ebXML SOAP ***Envelope*** 拡張名前空間がマッピングされていなければならない。

***schemaLocation*** 属性を正しく使用し、複数の名前空間にスキーマを決定することのできないツールでも、ebXML SOAP *メッセージ*を確認することができるようにするため、別の ***schemaLocation*** 属性を使用することを推奨する。以下に例を示す。

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
      http://ebxml.org/project_teams/transport/envelope.xsd" ...>
  <SOAP-ENV:Header xmlns:eb="http://www.ebxml.org/namespaces/messageHeader"
    xsi:schemaLocation="http://www.ebxml.org/namespaces/messageHeader
      http://ebxml.org/project_teams/transport/messageHeaderv0_99.xsd" ...>
    <eb:MessageHeader ...> ...
    </eb:MessageHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:eb="http://www.ebxml.org/namespaces/messageHeader"
    xsi:schemaLocation="http://www.ebxml.org/namespaces/messageHeader
      http://ebxml.org/project_teams/transport/messageHeaderv0_99.xsd" ...>
    <eb:Manifest ...> ...
    </eb:Manifest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 8.2.3 ebXML SOAP 拡張

ebXML メッセージは、以下の主要な拡張要素で SOAP *メッセージ*を拡張する。

- SOAP ***Header*** 拡張
  - ***MessageHeader*** – メッセージのルーティング情報(To/From など)や、メッセージに関するその他のコンテキスト情報を含む必須要素
  - ***TraceHeaderList*** – メッセージを送受信するメッセージ取扱サービスハンドラを識別するための項目を含む要素。この要素は省略可能である。
  - ***ErrorList*** – 前回のメッセージに対して報告されたエラーのリストを含む要素。***ErrorList*** 要素は、前回のメッセージでエラーが報告された場合のみ使用する。この要素は省略可能である。
  - ***Signature*** – メッセージに関連したデータ署名[XMLDSIG]に準拠したデジタル署名が含まれる要素。この要素は省略可能である。
  - ***Acknowledgment*** – *受領側 MSH* が *送信側 MSH* にメッセージの受領を通知するのに使用する要素。この要素は省略可能である。
  - ***Via*** – メッセージを受け取る次の ebXML メッセージ取扱サービスハンドラに情報を送るために使用される要素。この要素は省略可能である。
- SOAP ***Body*** 拡張
  - ***Manifest*** – *搬送コンテナ*またはそれ以外の場所(例えばウェブ)にあるデータを指し示す要素。この要素は省略可能である。
  - ***StatusRequest*** – 位置付けが要求されているメッセージの識別に使用される要素。この要素は省略可能である。
  - ***StatusResponse*** – 前回受領したメッセージの位置付けの要求に応じる際、MSH が使用する要素。この要素は省略可能である。
  - ***DeliveryReceipt*** – メッセージを受信する *To 側当事者*が、メッセージを送信した *From 側当事者* にメッセージの受信を知らせるために使用する要素。この要素は省略可能である。

### 8.2.4 #wildcard 要素のコンテンツ

ebXML SOAP 拡張要素の一部は、適格な外部の名前空間要素の内容を追加して、拡張性を高めることが可能である。拡張要素の内容は、[XMLNamespaces]に準拠した適格な名前空間でなければならず、また、外部名前空間に属していなければならない。外部名前空間とは、
http://www.ebxml.org/namespaces/messageHeaderに含まれていないものである。

追加された適格な外部名前空間要素には、SOAP の ***mustUnderstand*** 属性が含まれていなければならない。SOAP ***mustUnderstand*** 属性がない場合、既定値は「0」(偽)となる。。実装された MSH が要素の名前空間を認識せず、SOAP ***mustUnderstand*** 属性が「1」(真)の場合、MSH は、***errorCode*** を ***NotSupported*** に、そして ***severity*** を ***error*** に設定して、エラーを報告しなければならない(11項参照)。***mustUnderstand*** 属性の値が「0」の場合、もしくは ***mustUnderstand*** 属性がない場合、実装された MSH は適格な名前空間要素とその内容を無視することができる。

### 8.2.5 id 属性

先にリストされた ebXML SOAP 拡張要素には、オプションの ***id*** 属性が含まれる。この ***id*** 属性は、SOAP *メッセージ*内の要素を一意に識別できるようにするために加えることが可能な XML ID である。個々の ebXML SOAP 拡張要素は ***Reference*** 要素の「#<idvalue>」の URI を指定することによって追加したり削除したりできるので、ebXML SOAP *メッセージ*にデジタル署名を適用する際に使用できる。

## 8.3 SOAP ヘッダ要素

SOAP ***Header***要素は、SOAP ***Envelope***要素の最初の子要素である。これには、「http://schemas.xmlsoap.org/soap/envelope/」に含まれる名前空間のSOAP ***Envelope***名前空間宣言文と一致する名前空間限定詞がなければならない。以下に例を示す。

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" …>
```

```
    <SOAP-ENV:Header>…</SOAP-ENV:Header>
    <SOAP-ENV:Body>…</SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

SOAP ***Header*** 要素には、上で識別され、また以下の項で説明される ebXML SOAP ***Header*** 拡張要素の内容を含む。

## 8.4 MessageHeader 要素

***MessageHeader*** 要素はすべての ebXML メッセージに必要であり、SOAP ***Header*** 要素の子要素として含めなければならない。

***MessageHeader*** 要素は、次の 10 個の従属要素から成る複合要素である。

- ***From***
- ***To***
- ***CPAId***
- ***ConversationId***
- ***Service***
- ***Action***
- ***MessageData***
- ***QualityOfServiceInfo***
- ***SequenceNumber***
- ***Description***

***MessageHeader*** 要素には、次の 2 つの属性が必要である。

- SOAP ***mustUnderstand***
- ***Version***

また、***MessageHeader*** 要素には ***id*** 属性を含めることもできる。詳細は8.2.5項を参照のこと。

### 8.4.1 From および To 要素

必須要素 ***From*** はメッセージ送信側を識別し、必須要素 ***To*** はメッセージの受信側を識別する。***To*** および ***From*** には、DUNS 番号などの論理識別子、あるいは電子メールアドレスなどの物理的な場所も示す識別子を使用できる。

***From*** および ***To*** 要素には、それぞれ ***PartyId*** という子要素を 1 つ以上含む。

***From*** または ***To*** 要素のいずれかに複数の ***PartyId*** が含まれている場合、リストのすべてのメンバは同じ組織を識別しなければならない。1 つの ***type*** 値が複数の識別システムを参照しているのでない限り、***PartyId*** 要素の 1 つのリスト内に ***type*** 属性の値が 2 回以上現れてはいけない。

注意: この仕組みは、当事者間のメッセージの転送に複数の中継が含まれる際、特に役立つ(8.5.4項「マルチホップの TraceHeader サンプル」および10.3項「ebXML 高信頼性メッセージ取扱プロトコル」参照)。一般的には、特定の識別システムを優先する場合のある中継先や宛先をサポートするため、*From 側*は既知のすべてのドメイン ID を提供するようにすべきである。

#### 8.4.1.1 PartyID element

***PartyId***要素には、文字列の値からなる***type***という要素が1つある。***type***属性は、***PartyId***要素に含まれる文字列が属する名前のドメインを示す。***type***属性の値は、*当事者*間で相互に同意され、理解されていなければならない。***type***属性の値はURIとすることを推奨する。また、これらの値は、EDIRA (ISO 6523)、EDIFACT ISO 9735、またはANSI ASC X12 I05レジストリから取得することを推奨する。

***PartyId type***属性がない場合、***PartyId***要素の内容はURIでなければならない[RFC2396]。これ以外の場合、*受信側MSH*は***errorCode***を***Inconsistent***、***severity***を***Error***にそれぞれ設定し、エラーを報告しなければならない(11項参照)。***PartyID***要素はURIとすることを強く推奨する。

以下の記述は ***From*** および ***To*** 要素の例である。。

```
<eb:From>
  <eb:PartyId eb:type="urn:duns">123456789</eb:PartyId>
  <eb:PartyId eb:type="SCAC">RDWY</PartyId>
</eb:From>
<eb:To>
  <eb:PartyId>mailto:joe@example.com</eb:PartyId>
</eb:To>
```

### 8.4.2 CPAId 要素

必須要素 ***CPAId*** は、当事者間のメッセージ交換を制御するパラメータを識別するための文字列である。メッセージ受信者は、メッセージの送信者を考慮しつつ、***CPAId*** を個々のパラメータセットに分解できなければならない。

***CPAId*** 要素は、当事者間で合意された名前空間内で唯一の値でなければならない。この値には、***From*** および ***To PartyId*** の値を連結したものや、どちらか一方の当事者のインターネットドメイン名を付したURI、あるいは他の名前付けサービスやレジストリサービスが提供し、管理する名前空間などを使用できる。***CPAId*** には URI を使用することを推奨する。

***CPAId*** は、ebXML コラボレーションプロトコルプロフィールおよびコラボレーションプロトコル合意書[ebCPP]で定める通り、*CPA* のインスタンスを参照することが可能である。。***CPAId*** 要素の例を以下に示す。

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

双方の当事者が *CPA* を使用している場合、高信頼性メッセージングパラメータは、***CPAId*** 要素で識別される通り、適切な *CPA* 要素によって決定される。

メッセージが *CPA* に一致しないと受信側が判断した場合の適切な処理については、本書では定めない。従って、受信側がこの不一致に対処できることが明らかでない限り、送信側はそのようなメッセージを生成すべきではない。

検出された不一致の結果、受信側がエラーの生成を選択した場合、***errorCode*** を ***Inconsistent***、***severity*** を ***Error*** に設定してエラーを報告しなければならない。***CPAId*** が認識されないという理由でエラーを生成することを選択した場合は、***errorCode*** を ***NotRecognized***、***severity*** を ***Error*** に設定してエラーを報告しなければならない。

### 8.4.3 ConversationId 要素

必須要素 ***ConversationId*** は、取引*当事者*間の会話を構成する関連メッセージ一式を識別する文字列である。この要素は、***From*** 側および ***To*** のペアで唯一のものでなければならない。会話を開始する側は、その会話に関するすべてのメッセージを反映することになる ***ConversationId*** 要素の値を決定する。

メッセージ受信側は、***ConversationId*** により、これまでの会話内のメッセージを生成または処理したアプリケーションやプロセスのインスタンスを識別できるようになる。。***ConversationId*** は 1 つの会話に含まれるすべてのメッセージで不変である。

***ConversationId*** に使用される値は、実装によって異なる。以下に ***ConversationId*** 要素の例を示す。

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

注意: 特定の会話に関連した会話位置付けの識別および保存方法は実装ごとに選択できる。ただし、自身の識別スキーマと他の実装によって生成された ***ConversationId*** は容易にマッピングできるようにしておく必要がある。

### 8.4.4 Service 要素

必須要素 ***Service*** は、メッセージに作用するサービスを識別する。これは、サービスの設計者によって指定される。サービスの設計者としては以下が考えられる。

- 規格化組織
- 個人もしくは企業

注意: ebXML ビジネスプロセスモデルでは、*action* は[ebBPSS] (要求または応答の役割)で最も可能性の低い役割に基づく活動に相当し、*service* は当事者内で許可された役割に関連する一連の活動である。

***Service*** 要素の例を以下に示す。

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

注意: 名前空間が ***uri:www.ebxml.org/messageService/***で始まる ***Service*** 要素の URI は、本書で使用するために予約されている。

***Service*** 要素には ***type*** 属性がある。

#### 8.4.4.1 type 属性

***type***属性があるということは、メッセージの送信側および受信側が、別の手段で***Service***要素の内容を解釈する方法を知っていることを示す。両当事者は、解釈の一助としてこの***type***属性の値を使用できる。

***type*** 属性がない場合、***Service*** 要素の内容は URI でなければならない[RFC2396]。URI でない場合は、***errorCode*** を ***Inconsistent***、***severity*** を ***Error*** に設定してエラーを報告するようにする(11項参照)。

### 8.4.5 Action 要素

必須要素 ***Action*** は、メッセージを処理する ***Service*** 内のプロセスを識別する。***Action*** は、定義される ***Service*** 内で唯一のものでなければならない。***Action*** 要素の例を以下に示す。

```
   <eb:Action>NewOrder</eb:Action>
```

### 8.4.6 MessageData 要素

必須要素 ***MessageData*** は、ebXML メッセージを独自に識別する方法を提供する。この要素には、以下の 4 つの従属要素を含む。

- ***MessageId***
- ***Timestamp***
- ***RefToMessageId***
- ***TimeToLive***

以下の記述は ***MessageData*** 要素の構造を示す。

```
   <eb:MessageData>
     <eb:MessageId>20001209-133003-28572@example.com</eb:MessageId>
     <eb:Timestamp>2001-02-15T11:12:12Z</eb:Timestamp>
     <eb:RefToMessageId>20001209-133003-28571@example.com</eb:RefToMessageId>
   </eb:MessageData>
```

#### 8.4.6.1 MessageId 要素

必須要素 ***MessageId*** は、[RFC2392]に準拠したメッセージの一意の識別子である。[RFC2392]で定める識別子の「ローカル部」は、実装により異なる。

#### 8.4.6.2 Timestamp 要素

必須要素 ***Timestamp*** は、[XMLSchema]の timeInstant に従って、メッセージヘッダが作成された時間を表す値である。

#### 8.4.6.3 RefToMessageId 要素

***RefToMessageId*** 要素の多重度は 0 または 1 である。***RefToMessageId*** 要素がある場合は、そのメッセージが関連する、以前の ebXML メッセージの、***MessageId*** の値でなければならない。関連するメッセージが以前にない場合は、この要素を含めることができない。

エラーメッセージには ***RefToMessageId*** 要素が必要で、その値はエラーを含むメッセージの ***MessageId*** 値でなければならない(11項で定める通り)。

受領通知メッセージには ***RefToMessageId*** 要素が必要であり、その値は受領が通知された ebXML メッセージの ***MessageId*** 値でなければならない。8.13.4項および10項も参照のこと。

***StatusRequest*** または ***StatusResponse*** 要素内に含まれる場合、***RefToMessageId*** は現在の位置付けが照会されているメッセージを識別する(9.1項参照)。

#### 8.4.6.4 要素

***TimeToLive*** 要素は、メッセージが *To 側*に配信され、処理されるまでの時間を示す。***TimeToLive*** 要素については、10項の高信頼性メッセージングで説明される。

### 8.4.7 QualityOfServiceInfo 要素

***QualityOfServiceInfo*** 要素は、メッセージ配信のサービス品質を識別する。この要素には次の 3 つの属性がある。

- ***deliverySemantics***
- ***messageOrderSemantics***
- ***deliveryReceiptRequested***

この要素内のいずれかの属性を既定以外の値に設定する必要がある場合、***QualityOfServiceInfo*** 要素が必要である。***deliverySemantics*** 属性は、高信頼性メッセージングをサポートする。10項で詳しく説明する。***deliverySemantics*** 属性は、メッセージを信頼度の高い方法で送信するか否かを示す。

#### 8.4.7.1 deliveryReceiptRequested 属性

***deliveryReceiptRequested*** 属性は、*To 側*がメッセージを受け取り、***DeliveryReceipt*** 要素を含む受領メッセージを返信するようにするかどうかを示すために、*From 側*が使用する。

注意: ***DeliveryReceipt を***含む受領通知メッセージと高信頼性メッセージングの受領通知は次のように区別される。(1) ***DeliveryReceip を***含む受領通知メッセージは、*To 側*がメッセージを受信したことを示し、(2) 高信頼性メッセージングの受領通知は、MSH (中継 MSH だけの可能性もある)がメッセージを受信したことを示す。

***deliveryReceiptRequested*** の値を設定する前に、*From 側*は、要求された種類の配信受領書を *To 側*がサポートしているかどうか確認する必要がある([ebCPP]も参照のこと)。

有効な ***deliveryReceiptRequested*** の値は以下の通りである。

- ***Unsigned*** - 署名のない配信受領書が要求されることを要求する。
- ***Signed*** - 署名付きの配信受領書が必須であることを要求する。
- ***None*** - 配信受領書が必須でないことを示す。

***deliveryReceiptRequested*** の値はデフォルトで ***None*** である。

***deliveryReceiptRequested*** 属性が ***Signed*** または ***Unsigned*** に設定されたメッセージを受け取った場合、*To 側*は要求された配信受領書の種類をサポートできるか否かを確認する必要がある。

要求された種類の配信受領書を生成することができる場合、*To 側*は、***DeliveryReceipt*** 要素を含むメッセージを *From 側*に返信しなければならない。

要求された種類の配信受領書を返信することができない場合、*To 側*は、***errorCode*** を ***NotSupported***、***severity*** を ***Error*** に設定して、*From 側*にエラーを報告しなければならない。

受信したメッセージにエラーがなく、搬送内容データが含まれたメッセージの一部としてではなく ***DeliveryReceipt*** がそれ自体で送信された場合、***Service*** および ***Action*** は次のように設定しなければならない。

- ***Service*** 要素は ***uri:www.ebXML.org/messageService/***に設定しなければならない。
- ***Action*** 要素は ***DeliveryReceipt*** に設定しなければならない。

***deliveryReceiptRequested*** の例を以下に示す。

```
<eb:QualityOfServiceInfo eb:deliverySemantics="OnceAndOnlyOnce"
            eb:messageOrderSemantics="Guaranteed"
            eb:deliveryReceiptRequested="Unsigned"/>
```

#### 8.4.7.2 messageOrderSemantics 属性

***messageOrderSemantics*** 属性は、送信側アプリケーションが指定した順序で、メッセージが受信側アプリケーションに引き渡されるようにするかどうかを示すのに使用する。有効な値は以下の通りである。

- ***Guaranteed*** - メッセージは、送信側アプリケーションが指定した順序で受信側アプリケーションに引き渡される。
- ***NotGuaranteed*** - メッセージは、送信側アプリケーションが指定した順序とは異なる順序で受信側アプリケーションに渡される可能性がある。

***messageOrderSemantics*** の既定値は、*CPA* もしくは ***MessageHeader*** で指定される。指定されていない場合、値は既定で ***NotGuaranteed*** となる。

***messageOrderSemantics*** が ***Guaranteed*** に設定されている場合、*To 側*の MSH は、***ConversationId*** によって定められた会話の ***SequenceNumber*** の値を使用して、無効なメッセージの順序を修正できる。***messageOrderSemantics*** が ***Guaranteed*** に設定されている場合は、***SequenceNumber*** 要素がなければならない。

***deliverySemantics*** が ***OnceAndOnlyOnce*** ではなく、***messageOrderSemantics*** が ***Guaranteed*** に設定されている場合は、***errorCode*** を ***Inconsistent***、***severity*** を ***Error*** に設定して *From 側*にエラーを報告する(10項および11項を参照)。

***DeliverySemantics*** 属性に ***OnceandOnlyOnce*** という値を持つ、同一会話内で送信されたメッセージは、それぞれに同じ値の ***messageOrderSemantics*** (***Guaranteed*** または ***NotGuaranteed***)が含まれていなければならない。

***messageOrderSemantics*** が ***NotGuaranteed*** に設定されている場合、*To 側*の MSH はメッセージの無効な順序を修正する必要はない。

要求された ***messageOrderSemantics*** の種類をサポートできない場合、*To 側*は、***errorCode*** を ***NotSupported***、***secerity*** を ***Error*** に設定して *From 側*にエラーを報告しなければならない。***messageOrderSemantics*** の例を以下に示す。

```
<eb:QualityOfServiceInfo eb:deliverySemantics=”OnceAndOnlyOnce”
    eb:messageOrderSemantics=”Guaranteed”/>
```

### 8.4.8 SequenceNumber 要素

***SequenceNumber*** 要素は、*受信側 MSH* が処理しなければならないメッセージの順序を示す。***SequenceNumber*** は、***ConversationId*** および MSH で一意である。。*From 側*の MSH が ***ConversationID*** 内に設定するのと同様、*From 側*および *To 側*の MSH は、それぞれ独立した ***SequenceNumber*** を設定する。その MSH からの会話の最初のメッセージにはゼロが設定され、それ以降、メッセージが送信されるごとに 1 ずつ加えられる。

***SequenceNumber*** 要素は、***deliverySemantics*** の値が ***OnceAndOnlyOnce*** で、***messageOrderSemantics*** の値が ***Guaranteed*** の時にのみ含まれなければならない。この規準を満たさない場合、From 側 MSH には、***errorCode*** を ***Inconsisitent***、***severity*** を ***Error*** に設定してエラーを報告しなければならない。

***SequenceNumber*** 要素付きのメッセージを受信する MSH は、順序を外れたメッセージの保存に必要な格納域が、実装で定められた制限内に収まっている限り、***SequenceNumber*** の低いすべてのメッセージが受領され、アプリケーションに送られるまで、メッセージをアプリケーションに渡すことはできない。

順序を外れたメッセージの保存に必要な格納域が、実装で定められた制限に達した場合、*受信側 MSH* は、***errorCode*** を ***DeliveryFailure***、***severity*** を ***Error*** に設定して配信の失敗を*送信側 MSH* に示さなければならない(11項参照)。

***SequenceNumber***要素は、アプリケーションで作成されMSHが送信したメッセージそれぞれについて、*送信側MSH*が(例えば0, 1, 2, 3, 4…のように) 1ずつ増やしていく***ConversationId***内の整数値である。99999999の次の値は「0」になる。***SequenceNumber***の値は、0～99999999のASCIIの序数から成る。以下のケースでは、***SequenceNumbe***の値が「0」になる。。

1) 会話内の*送信側 MSH* の最初のメッセージ
2) *送信側 MSH* が ***SequenceNumber*** 情報をリセットした後の最初のメッセージ
3) 値が一巡した後の最初のメッセージ(99999999 の次の値)

***SequenceNumber*** 要素には、***status*** という属性が 1 つある。この属性には、以下のいずれかの値が必要である。

- ***Reset*** – ***SequenceNumber*** は、上述の 1 または 2 で示されている通りリセットされる。
- ***Continue*** – ***SequenceNumber*** は順序に続く(上の 3 を含む)

上の 1 または 2 により、***SequenceNumber*** が 0 に設定されたら、*送信側 MSH* はメッセージの ***status*** 属性を ***Reset*** に設定しなければならない。3 の場合を含むその他のケースでは、***status*** 属性を ***Continue*** に設定しなければならない。

*送信側 MSH* は、以前送信されたメッセージの会話の*受領通知メッセージ*をすべて受け取るまで、その会話の ***SequenceNumber*** のリセットを待たなければならない。送信されたすべてのメッセージの受領が通知された場合のみ、*送信側 MSH* は ***SequenceNumber*** をリセットできる。***SequenceNumber*** の例を以下に示す。

```
<eb:SequenceNumber eb:status=”Reset”>0</eb:SequenceNumber>
```

### 8.4.9 Description 要素

***Description*** 要素は、提示されなかったり、あるいは ***MessageHeader*** の子要素として何度も提示されたりする場合がある。この要素の目的は、人が判読可能な形でメッセージの目的や意図の説明を提供することにある。説明の言語は、必須属性 ***xml:lang*** によって定められる。***xml:lang*** 属性は、[XML]で指定されている言語識別規則に従わなければならない。***xml:lang*** の値は、毎回異なる値でなければならない。

### 8.4.10 Version 属性

必須属性 ***version*** は、ebXML SOAP ***ヘッダ***拡張が従う *ebXML メッセージ取扱サービス*ヘッダ仕様のバージョンを示す。その目的は、将来、バージョンの識別を可能にするためである。現在のところ、***version*** 属性の値は「1.0」であるが、この仕様の将来のバージョンには、この属性に異なる値が必要となる。version 属性は、上で定義された ebXML SOAP ***エンベロープ***拡張の名前空間に適格な名前空間でなければならない。

### 8.4.11 SOAP mustUnderstand 属性

SOAP 名前空間に適格な名前空間(http://schemas.xmlsoap.org/soap/envelope/)である必須属性 SOAP ***mustUnderstand*** は、***MessageHeader*** 要素の内容が受信側のプロセスによって理解されなければならない。もし理解されない場合は、[SOAP]に従ってメッセージが拒否される。また、この属性は「1」(真)でなければならない。

### 8.4.12 MessageHeader サンプル

以下の記述は、SOAP ***ヘッダ***内の ***MessageHeader*** 要素の構造を示す。

```
<eb:MessageHeader id="…" eb:version="1.0" SOAP-ENV:mustUnderstand="1">
  <eb:From><eb:PartyId>uri:example.com</eb:PartyId></eb:From>
  <eb:To eb:type="someType">
     <eb:PartyId eb:type="someType">QRS543</eb:PartyId>
  </eb:To>
  <eb:CPAId>http://www.ebxml.org/cpa/123456</eb:CPAId>
  <eb:ConversationId>987654321</eb:ConversationId>
  <eb:Service eb:type="myservicetypes">QuoteToCollect</eb:Service>
  <eb:Action>NewPurchaseOrder</eb:Action>
  <eb:MessageData>
    <eb:MessageId>mid:UUID-2</eb:MessageId>
    <eb:Timestamp>2000-07-25T12:19:05Z</eb:Timestamp>
    <eb:RefToMessageId>mid:UUID-1</eb:RefToMessageId>
  </eb:MessageData>
  <eb:QualityOfServiceInfo
      eb:deliverySemantics=”OnceAndOnlyOnce”
      eb:deliveryReceiptRequested=”Signed”/>
</eb:MessageHeader>
```

## 8.5 TraceHeaderList 要素

***TraceHeaderList*** 要素は、1 つ以上の ***TraceHeader*** 要素から成る。データ通信プロトコルでメッセージが送信される前に、***TraceHeaderList*** には既存の ***TraceHeader*** に続いて ***TraceHeader*** が 1 つ追加される。

以下の場合、***TraceHeaderList*** 要素は省略することができる。

- シングルホップ(8.5.3項参照)でメッセージが送信される場合
- 信頼性の高い方法でメッセージが送信されない場合(10項参照)

***TraceHeaderList*** 要素には、以下の 3 つの必須属性がある。

- SOAP ***mustUnderstand*** (詳細は8.4.11項参照)
- 値が「http://schemas.xmlsoap.org/soap/actor/next」の SOAP ***actor*** 属性
- ***Version*** (詳細は8.4.10項参照)

また、***TraceHeaderList*** 要素には ***id*** 属性を含めることもできる。詳細は8.2.5項を参照のこと。

### 8.5.1 SOAP actor 属性

***TraceHeaderList*** 要素には、値が http://schemas.xmlsoap.org/soap/actor/next の SOAP ***actor*** 属性が含まれていなければならず、[SOAP]仕様に定められている通り解釈および処理されなければならない。こ

れは、***TraceHeaderList*** 要素がメッセージを受信する MSH によって処理されなければならず、次の MSH に転送されてはならないことを意味する。***TraceHeaderList*** 要素を処理する MSH は、***TraceHeaderList*** に新たな ***TraceHeader*** 要素を加え、次の MSH へのメッセージに再度挿入する必要がある。

### 8.5.2 TraceHeader 要素

***TraceHeader*** 要素には、2 つの MSH インスタンス間で送信される 1 回のメッセージの情報を含む。*From 側の* MSH から *To 側の* MSH に到着するまで、メッセージが 1 つ以上の中間 MSH ノードを通過し、複数のホップを横断する場合、*送信側 MSH* により、それぞれの「ホップ」での送信で新たな ***TraceHeader*** 要素が追加される。

***TraceHeader*** 要素は、以下の従属要素から成る複合要素である。

- ***Sender***
- ***Receiver***
- ***Timestamp***
- ***#wildcard***

また、***TraceHeader*** 要素には ***id*** 属性を含むこともできる。詳細は8.2.5項を参照のこと。

#### 8.5.2.1 Sender 要素

***Sender*** 要素は、以下の従属要素から成る複合要素である。

- ***PartyId***
- ***Location***

***From*** および ***To*** 要素と同様に、***Sender*** 要素には複数の ***PartyId*** 要素をリストすることができる。これにより、受信システムは、事前に当事者全員が 1 つのスキームに合意していなくとも、好みの識別スキームを使用して、識別子を決定できる。

##### 8.5.2.1.1 PartyId 要素

この要素の構文および意味は、8.4.1.1項の ***PartyId*** 要素で説明されている。ここでは、識別される当事者はメッセージの送信者である。この要素は、メッセージの ***To*** 要素を含めることにより、この当事者を宛先とした後のメッセージに使用できる。

##### 8.5.2.1.2 Location 要素

この要素には、送信者のメッセージハンドラの URI が含まれる。CPA または ***MessageHeader*** (8.4.2 項) で別の URI が識別されていない限り、以下のような場合、メッセージの受信者はメッセージの送信にこの URI を使用する。

- 前のメッセージに返信する場合
- 前のメッセージの受領を通知する場合
- 前のメッセージのエラーを報告する場合

#### 8.5.2.2 Receiver 要素

***Receiver*** 要素は、以下の従属要素から成る複合要素である。

- ***PartyId***
- ***Location***

***From*** および ***To*** 要素と同様に、***Receiver*** 要素には複数の ***PartyId*** 要素をリストすることができる。これにより、送信システムは、事前に当事者全員が 1 つのスキームに合意していなくとも、好みの識別スキームを使用して、これらの識別子を決定できる。

***Receiver*** 要素の子要素(***PartyId*** および ***Location***)は、Sender 要素と同様の方法で実装される(8.5.2.1.1項および8.5.2.1.2項参照).

#### 8.5.2.3 Timestamp 要素

***Timestamp*** 要素は個々の ***TraceHeader*** が作成された時間であり、***MessageData*** 要素(8.4.6.2項)の ***Timestamp*** 要素と同じ形式である。

#### 8.5.2.4 #wildcard 要素

***#wildcard*** 要素の取り扱いに関する説明は、8.2.4項を参照のこと。

### 8.5.3 シングルホップの TraceHeader サンプル

図 8-1 にシングルホップメッセージ示す。

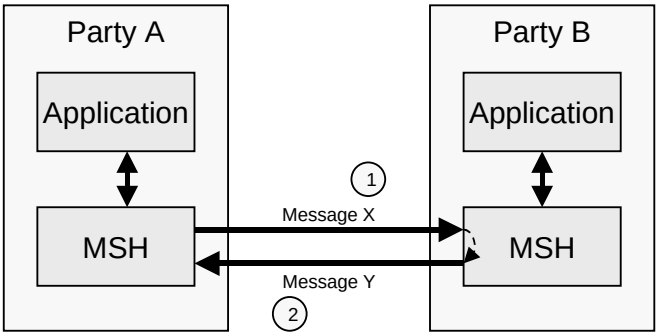



**図8-1 シングルホップメッセージ**

対応するメッセージの内容には、以下のものが含まれる場合がある。

- 送信 1 - 当事者 A から当事者 B へのメッセージ X

```
<eb:MessageHeader eb:id=”...” eb:version="1.0" SOAP-ENV:mustUnderstand="1">
  <eb:From>
     <eb:PartyId>urn:myscheme.com:id:PartyA-id</eb:PartyId>
  </eb:From>
  <eb:To>
     <eb:PartyId>urn:myscheme.com:id:PartyB-id</eb:PartyId>
  </eb:To>
  <eb:ConversationId>219cdj89dj2398djfjn</eb:ConversationId>
  ...
  <eb:MessageData>
    <eb:MessageId>29dmridj103kvna</eb:MessageId>
    ...
  </eb:MessageData>
   ...
</eb:MessageHeader>

<eb:TraceHeaderList eb:id=”...” eb:version="1.0" SOAP-ENV:mustUnderstand="1">
  <eb:TraceHeader>
    <eb:Sender>
       <eb:PartyId>urn:myscheme.com:id:PartyA-id</eb:PartyId>
      <eb:Location>http://PartyA.com/PartyAMsh</eb:Location>
    </eb:Sender>
    <eb:Receiver>
       <eb:PartyId>urn:myscheme.com:id:PartyB-id</eb:PartyId>
       <eb:Location>http://PartyB.com/PartyBMsh</eb:Location>
    </eb:Receiver>
    <eb:Timestamp>2000-12-16T21:19:35Z</eb:Timestamp>
  </eb:TraceHeader>
</eb:TraceHeaderList>
```

### 8.5.4　マルチホップの TraceHeader サンプル

マルチホップメッセージは、図 8-2 に示す通り、送信者から受信者へ直接送られず、中間の当事者を経て送信される。

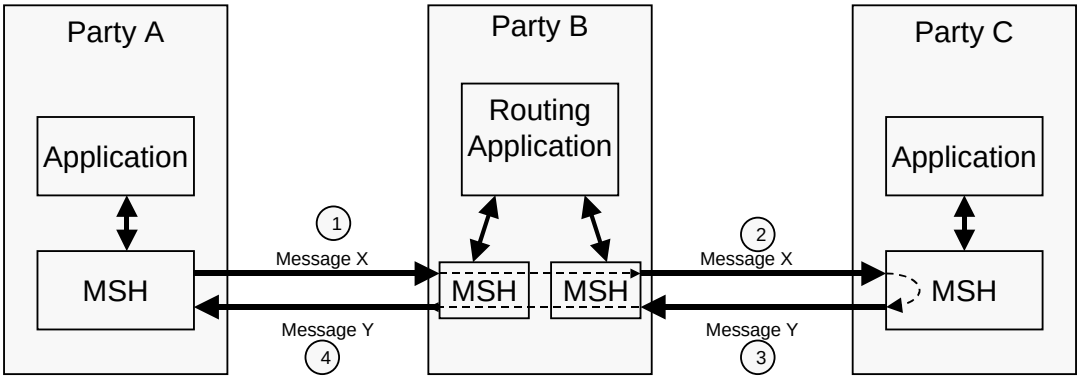



**図8-2 マルチホップメッセージ**

対応するメッセージの内容には、以下のものが含まれる場合がある。

- 送信 1 - 当事者 A から当事者 B へのメッセージ X

```
<eb:MessageHeader eb:id=”...” eb:version="1.0" SOAP-ENV:mustUnderstand="1">
  <eb:From>
     <eb:PartyId>urn:myscheme.com:id:PartyA-id</eb:PartyId>
  </eb:From>
  <eb:To>
     <eb:PartyId>urn:myscheme.com:id:PartyC-id</eb:PartyId>
  </eb:To>
  <eb:ConversationId>219cdj89dj2398djfjn</eb:ConversationId>
  ...
  <eb:MessageData>
    <eb:MessageId>29dmridj103kvna</eb:MessageId>
    ...
  </eb:MessageData>
   ...
</eb:MessageHeader>

<eb:TraceHeaderList eb:id=”...” eb:version="1.0" SOAP-ENV:mustUnderstand="1"
       SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next">
  <eb:TraceHeader>
    <eb:Sender>
       <eb:PartyId>urn:myscheme.com:id:PartyA-id</eb:PartyId>
       <eb:Location>http://PartyA.com/PartyAMsh</eb:Location>
    </eb:Sender>
    <eb:Receiver>
       <eb:Location>http://PartyB.com/PartyBMsh</eb:Location>
    </eb:Receiver>
    <eb:Timestamp>2000-12-16T21:19:35Z</eb:Timestamp>
  </eb:TraceHeader>
</eb:TraceHeaderList>
```

- 送信 2 - 当事者 B から当事者 C へのメッセージ X

```
<eb:MessageHeader eb:id=”...” eb:version="1.0" SOAP-ENV:mustUnderstand="1">
  <eb:From>
     <eb:PartyId>urn:myscheme.com:id:PartyA-id</eb:PartyId>
  </eb:From>
  <eb:To>
     <eb:PartyId>urn:myscheme.com:id:PartyC-id</eb:PartyId>
  </eb:To>
  <eb:ConversationId>219cdj89dj2398djfjn</eb:ConversationId>
  ...
```

```
  <eb:MessageData>
    <eb:MessageId>29dmridj103kvna</eb:MessageId>
    ...
  </eb:MessageData>
   ...
</eb:MessageHeader>

<eb:TraceHeaderList eb:id=”...” eb:version="1.0" SOAP-ENV:mustUnderstand="1"
    SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next">
  <eb:TraceHeader>
    <eb:Sender>
       <eb:PartyId>urn:myscheme.com:id:PartyA-id</eb:PartyId>
       <eb:Location>http://PartyA.com/PartyAMsh</eb:Location>
    </eb:Sender>
    <eb:Receiver>
       <eb:PartyId>urn:myscheme.com:id:PartyB-id</eb:PartyId>
       <eb:Location>http://PartyB.com/PartyBMsh</eb:Location>
    </eb:Receiver>
    <eb:Timestamp>2000-12-16T21:19:35Z</eb:Timestamp>
  </eb:TraceHeader>
  <eb:TraceHeader>
    <eb:Sender>
       <eb:PartyId>urn:myscheme.com:id:PartyB-id</eb:PartyId>
       <eb:Location>http://PartyB.com/PartyAMsh</eb:Location>
    </eb:Sender>
    <eb:Receiver>
       <eb:PartyId>urn:myscheme.com:id:PartyC-id</eb:PartyId>
       <eb:Location>http://PartyC.com/PartyBMsh</eb:Location>
    </eb:Receiver>
    <eb:Timestamp>2000-12-16T21:19:45Z</eb:Timestamp>
  </eb:TraceHeader>
</eb:TraceHeaderList>
```

## 8.6 Acknowledgment 要素

***Acknowledgment*** 要素は、別のメッセージ取扱サービスハンドラがメッセージを受信したことを示すためにメッセージ取扱サービスハンドラが使用する、オプションの要素である。***Acknowledgment*** 要素を含むメッセージの ***RefToMessageId*** は、***MessageId*** で受領通知がなされたメッセージを識別するために使用される。

***Acknowledgment*** 要素は、以下の要素および属性で構成される。

- ***Timestamp*** 要素
- ***From*** 要素
- 1 つ以上の ***ds:Reference*** 要素
- SOAP 必須属性 ***mustUnderstand*** (詳細は8.4.11項参照)
- SOAP 必須属性 ***actor***
- 必須属性 ***version*** (詳細は8.4.10項参照)
- ***id*** 属性 (詳細は8.2.5項参照)

### 8.6.1 Timestamp 要素

***Timestamp*** 要素は、受領通知メッセージを生成した*当事者*がメッセージを受信した時間を表す値であり、[XMLSchema]の timeInstant (8.4.6.2項)に従っていなければならない。

### 8.6.2 From 要素

これは、***MessageHeader*** 要素内の ***From*** 要素と同じ要素である。(8.4.1項参照)。しかし、***Acknowledgment*** 要素で使用される時は、*受領通知メッセージ*を生成している*当事者*の識別子が含まれる。

***From*** 要素が省略された場合、要素を送信した当事者は ***MessageHeader*** 要素の ***From*** 要素で識別される。

### 8.6.3 ds:Reference 要素

受領通知は、受領が通知されているメッセージから取得または派生した[XMLDSIG]名前空間 (http://www.w3.org/2000/09/xmldsig#)の ***Reference*** 要素を 1 つ以上含めることで、MSH が受信を拒否できないようにするために使用できる。***Reference*** 要素は、前述の名前空間に適格な名前空間でなければならず、また、XML 署名[XMLDSIG]の仕様に準拠していなければならない。

### 8.6.4 SOAP actor 属性

***Acknowledgment*** 要素には、値が http://schemas.xmlsoap.org/soap/actor/next の SOAP ***actor*** 属性が含まれていなければならず、[SOAP]仕様に定められている通り解釈および処理されなければならない。これは、***Acknowledgment*** 要素がメッセージを受信する MSH によって処理されなければならず、次の MSH に転送されてはならないことを意味する。

### 8.6.5 Acknowledgement サンプル

***Acknowledgement*** 要素の例を以下に示す。

```
<eb:Acknowledgment SOAP-ENV:mustUnderstand="1" eb:version="1.0"
    SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next">
  <eb:Timestamp>2001-03-09T12:22:30Z</eb:Timestamp>
  <eb:From>
     <eb:PartyId>uri:www.example.com</eb:PartyId>
  </eb:From>
</eb:Acknowledgment>
```

## 8.7 Via 要素

***Via*** 要素は、次にメッセージを受け取る ebXML メッセージ取扱サービスハンドラ(MSH)への情報の送信に使用される、*SOAP* ***ヘッダ***の ebXML 拡張である。

注意: この MSH は、中間の取引当事者が運営する場合と、*To 側*が運営する場合が考えられる。***Via*** 要素は、特にホップごとに異なる可能性のあるデータの保存に使用される。

***Via*** 要素には以下の属性が含まれていなければならない。

- ***id*** 属性( 8.2.5項参照)
- ***version*** 属性(詳細は 8.4.10項参照)
- SOAP ***MustUnderstand*** 属性
- SOAP ***actor*** 属性

また、***Via*** 要素には、以下の要素または属性が 1 つ以上含まれていなければならない。

- ***syncReply*** 属性
- ***reliableMessagingMethod*** 属性
- ***ackRequested*** 属性
- ***CPAId*** 要素

***Via*** 要素には、以下の要素を含めることもできる。

- ***Service*** 要素
- ***Action*** 要素

### 8.7.1 SOAP mustUnderstand 属性

SOAP エンベロープ名前空間に適格な名前空間(http://schemas.xmlsoap.org/soap/envelope/)である必須属性 SOAP ***mustUnderstand*** は、***Via*** 要素の内容が受領側のプロセスによって理解される必要があり、もし理解されない場合は、[SOAP]に従ってメッセージが拒否されなければならないことを示す。この属性は「1」(真)でなければならない。***Via*** 要素をサポートしない受信側の *ebXML メッセージ取扱サービス* は、[SOAP]仕様に従って、***faultCode*** を ***MustUnerstand*** に設定して SOAP ***Fault*** を返さなければならない。

### 8.7.2 SOAP actor 属性

***Via*** 要素には、http://schemas.xmlsoap.org/soap/actor/nextの値を持った SOAP ***actor*** 属性が含まれていなければいけない。また、[SOAP]仕様で定められている通りに解釈および処理されなければならない。これは、***Via*** 要素が、メッセージを受信する SOAP プロセッサによって処理されて、次の MSH に転送されてはならないことを意味する。

### 8.7.3 syncReply 属性

***syncReply*** 属性は、データ通信プロトコルが*同期*の場合(HTTP など)にのみ使用される。これは[XML Schema]のブール値である。通信プロトコルが*同期*でない場合、***syncReply*** の値は無視される。***syncReply*** 属性がない場合は、意味的に「失敗」の値が含まれていることと同じになる。。値が ***true*** の ***syncReply*** 属性がある場合、MSH は同期返信の搬送内容のアプリケーションまたはビジネスプロセスから応答しなければならない。[ebCPP]仕様の ***syncReply*** の説明も参照のこと。

### 8.7.4 reliableMessagingMethod 属性

***reliableMessagingMethod*** 属性には、以下のいずれかが含まれていなければならない。

- ***ebXML***
- ***Transport***

この属性のデフォルト値は ***ebXML*** である。

### 8.7.5 ackRequested 属性

***ackReqested*** 属性には、以下のいずれかが含まれていなければならない。

- ***Signed***
- ***Unsigned***
- ***None***

この属性の既定値は ***None*** である。この属性は、受領通知メッセージが送られるかどうか、そして受領通知が送られる場合、署名すべきかどうかを*受信側 MSH* に示すために使用される。この属性の詳しい使用方法は、10.2.5項を参照のこと。

### 8.7.6 CPAId 要素

***CPAId*** 要素は、2 つの MSH インスタンス間のメッセージ交換を制御するパラメータを識別するための文字列であり、***CPAId*** で識別されるパラメータが、***MessageHeader*** の ***To*** および ***From*** 要素で識別される*当事者*間ではなく、2 つの MSH インスタンス間のメッセージ交換にのみ適用されることを除き、***MessageHeader*** の ***CPAId*** と意味は同じである。(8.4.2参照)。これにより、メッセージが中継の MSH を通過する際、異なるパラメータや転送プロトコルを別のホップで使用できるようになる。

***CPAId*** 要素が含まれている場合、***MessageHeader*** 要素には ***CPAId*** で識別される値ではなく、このパラメータの値を使用しなければならない。

### 8.7.7 Service および Action 要素

MSH が *To 側*によって運営されているかどうかに関わらず、***Service*** および ***Action*** 要素が次の MSH に取り扱われることを除き、これらの要素は ***MessageHeader*** 要素と同じ意味である。(8.4.4項および 8.4.5項参照)。

***Service*** および ***Action*** に使用する値は、*ebXML メッセージ取扱サービス*を使用するサービスまたはビジネスプロセスの設計者が定める。

***Service*** および ***Action*** 要素はオプションである。しかし、***Service*** 要素が含まれている場合は ***Action*** 要素も含む必要がある。また、***Action*** 要素が含まれている場合は ***Service*** 要素も含まれていなければならない。

### 8.7.8 Via 要素サンプル

***Via*** 要素の例を以下に示す。

```
<eb:Via SOAP-ENV:mustUnderstand="1" eb:version="1.0"
   SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
   eb:syncReply="false">
    <eb:CPAId>yaddaydda</eb:CPAId>
    <eb:Service>urn:services:Proxy</eb:Service>
    <eb:Action>LogActivity</eb:Action>
</eb:Via>
```

## 8.8 ErrorList 要素

SOAP ***Header*** 要素内に ***ErrorList*** が含まれている場合、これは ***MessageHeader*** 要素の ***RefToMessageId*** によって識別されるメッセージ内にエラーが含まれていることを示す。

***ErrorList*** 要素は、以下の属性を持った 1 つ以上の ***Error*** 要素から成る。

- ***id*** 属性
- SOAP ***mustUnderstand*** 属性(詳細は8.4.11項参照)
- ***version*** 属性(詳細は8.4.10項参照)
- ***highestSeverity*** 属性

報告すべきエラーがない場合、***ErrorList*** 要素を含めることはできない。

### 8.8.1 id 属性

***id*** 属性は、文書内の ***ErrorList*** 要素を一意に識別する(8.2.5項参照)。

### 8.8.2 highestSeverity 属性

***highestSeverity*** 属性には、***Error*** 要素が含まれる。特に、***severity*** が ***Error*** に設定された ***Error*** 属性が含まれている場合は ***highestSeverity*** を ***Error*** に、それ以外の ***Error*** 属性が含まれている場合は ***highestSeverity*** を ***Warning*** に設定しなければならない。

### 8.8.3 Error 要素

***Error*** 要素は以下の属性から成る。

- ***codeContext***
- ***errorCode***
- ***severity***
- ***location***
- ***xml:lang***
- ***id*** (詳細は8.2.5項参照)

***Error*** 要素の内容にはエラーメッセージが含まれる。

#### 8.8.3.1 codeContext 属性

必須属性 ***codeContext*** は、***errorCode*** の名前空間またはスキームを識別する。この属性は URI でなければならない。デフォルト値は ***http://www.ebxml.org/messageServiceErrors*** の値である。デフォルト値が含まれていない場合は、本書の実装に独自の ***errorCode*** が使用されていることを表す。

***errorCode*** に ebXML 以外の値を使用することは推奨されない。また、既存の ***errorCode*** の意味が本項で定められている内容と同じか、非常に類似している場合、本書の実装に独自の ***errorCode*** を使用してはならない。

#### 8.8.3.2 errorCode 属性

必須属性 ***errorCode*** は、エラーが発生したメッセージ内のエラーの性質を識別する。***errorCode*** の有効値およびコードの意味は、8.8.5.1項および8.8.5.2項で説明する。

#### 8.8.3.3 severity 属性

必須属性 ***severity*** は、エラーの重大性を表す。有効な値は以下の通りである。

- ***Warning*** - エラーは発生しているものの、会話内の他のメッセージは通常の方法で生成される。
- ***Error*** - メッセージ内に回復不可能なエラーがあり、この会話ではこれ以上メッセージが生成されない。

#### 8.8.3.4 Location 属性

***location*** 属性は、エラーが発生したメッセージの箇所を指し示す。

エラーが ebXML 要素内にあり、この要素が「正しく形成」されている場合([XML]参照)、***location*** 属性の内容は[XPointer]でなければならない。

エラーが、SOAP エンベロープおよび ebXML 搬送内容を含んだ MIME エンベロープに関連している場合、***location*** には、`cid:23912480wsr` という形式で、エラーが発生した MIME の `content-id` が含まれる。この形式の「:」以降のテキストは、MIME 部の `content-id` の値になる。

#### 8.8.3.5 Error 要素の内容

エラーメッセージの内容は、***xml:lang*** 属性で定められた言語により、エラーについて説明する。これは、通常、XML パーサまたはメッセージの確認を行うその他のソフトウェアによって生成されるメッセージである。従って、その内容は ***Error*** 要素を生成するソフトウェアのベンダや開発者が定めることになる。

***xml:lang*** 属性は、[XML]で定められた言語識別規則に従っていなければならない。

***Error*** 要素の内容には、何も含まれない場合もある。

### 8.8.4 ErrorList サンプル

***ErrorList*** 要素の例を以下に示す。

```
<eb:ErrorList eb:id='3490sdo9', eb:highestSeverity="error" eb:version="1.0"
          SOAP-ENV:mustUnderstand="1">
  <eb:Error eb:errorCode='SecurityFailure' eb:severity="Error"
     eb:location='URI_of_ds:Signature_goes_here' xml:lang="us-en">
     Validation of signature failed </eb:Error>
  <eb:Error ...> ... </eb:Error>
</eb:ErrorList>
```

### 8.8.5 errorCode の値

本項では、*エラーを報告するメッセージ*に使用する、***errorCode*** 要素(8.8.3.2項参照)の値の説明をする。これらの値は、3 つの項目で構成される表に示される。

- 最初の段には、例えば ***SecurityFailure*** など、***errorCode*** として使用される値が含まれる。
- 2 番目の段には、***errorCode*** の「概略」が含まれる。
  注意: この内容を ***Error*** 要素内で使用することはできない。
- 3 番目の段には、***errorCode*** の「説明」が含まれる。ここでは、エラーの意味と、特定の ***errorCode*** を使用すべき際の参考が示される。

#### 8.8.5.1 ebXML 要素内でのエラーの報告

以下のリストには、ebXML 要素に関連する可能性のあるエラーコードを示す。

| エラーコード | 概　略 | 説　　明 |
|---|---|---|
| ***ValueNotRecognized*** | 認識されない要素の内容または属性の値 | 文書の形式に問題はなく、有効であるものの、要素/属性に認識不可能な値が含まれており、*ebXML メッセージ取扱サービス*で使用できない。 |
| ***NotSupported*** | サポートされない要素または属性 | 文書の形式に問題はなく、有効である。しかし、本書の規則や制約には矛盾してはいないが、このメッセージを処理する *ebXML メッセージ取扱サービス*でサポートされていない要素/属性がある。 |
| ***Inconsistent*** | その他の要素や属性と矛盾した要素の内容または属性の値 | 文書は本書に含まれる規則や制約に従っており、その形式に問題はなく、有効であるものの、要素または属性の内容がその他の要素や属性の内容と一致しない。 |
| ***OtherXml*** | 要素内容または属性値のその他のエラー | 文書の形式に問題はなく、有効であるものの、本書の規則や制約に準拠しておらず、その他のエラーコードにも含まれていない値が要素の内容または属性値に含まれている。***Error*** 要素の内容を使用して、問題の性質を示す必要がある。 |

#### 8.8.5.2 XML 以外の文書のエラー

以下は、ebXML 要素と関連性のないエラーを識別するエラーコードである。

| エラーコード | 概　略 | 説　　明 |
|---|---|---|
| ***DeliveryFailure*** | メッセージ配信の失敗 | 次の宛先へ送信することが明らかに不可能な、あるいは不可能である可能性のあるメッセージを受け取った。注意: ***severity*** が ***Warning*** に設定されている場合、メッセージが配信される可能性は低くなる。 |
| ***TimeToLiveExpired*** | メッセージの有効期限切れ | ***MessageHeader*** 要素の ***TimeToLive*** 要素で指定された期限の後に到着したメッセージを受け取った。 |

| | | |
|---|---|---|
| ***SecurityFailure*** | メッセージのセキュリティチェック失敗 | メッセージの署名の確認、またはメッセージ送信者の認証や許可に失敗した。 |
| ***Unknown*** | 未知のエラー | その他のエラーの何れにも含まれていないエラーの発生を表す。***Error*** 要素の内容を使用し、問題の性質を示す必要がある。 |

## 8.9 ds:Signature 要素

ebXML メッセージは、セキュリティ対策のためデジタル署名が付されている場合がある。*SOAP ヘッダ*には、[XMLDSIG]で定義された名前空間に属する ***ds:Signature*** 要素が含まれていることがある。この ***ds:Signature*** 要素は、[XMLDSIG]に準じた適格な名前空間でなければならない。また、***ds:Signature*** 要素の構造および内容は、[XMLDSIG]仕様に準拠していなければならない。*SOAP* ***ヘッダ***内に複数の ***ds:Signature*** 要素が含まれている場合、最初の ***ds:Signature*** 要素は、第 12 節に従って *From 側の* MSH が署名した ebXML メッセージのデジタル署名でなければならない。それ以外の ***ds:Signature*** 要素も含めることができるが、本書ではこれらの署名の目的を定めていない。

ebXML メッセージにデジタル署名を付す際の ***ds:Signature*** 要素の構築方法についての詳細は、第 12 節を参照のこと。

## 8.10 SOAP Body 拡張

SOAP ***Body***要素は、SOAP ***Envelope***要素の2番目の子要素である。
「http://schemas.xmlsoap.org/soap/envelope/」に含まれる名前空間のSOAP ***Envelope***名前空間宣言文と一致する名前空間限定詞がなければならない。以下に例を示す。

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" …>
  <SOAP-ENV:Header>…</SOAP-ENV:Header>
  <SOAP-ENV:Body>…</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP ***Body***要素には、以下のebXML SOAP ***Body***拡張要素の内容が含まれる。

- ***Manifest*** 要素
- ***StatusRequest*** 要素
- ***StatusResponse*** 要素
- ***DeliveryReceipt*** 要素

以下の項でそれぞれの要素について定める。

## 8.11 Manifest 要素

***Manifest*** 要素は、1 つ以上の ***Reference*** 要素から成る複合要素である。各 ***Reference*** 要素は、搬送内容文書が*搬送コンテナ*に含まれる搬送内容文書としてメッセージの一部に含まれているか、あるいは URL でアクセス可能な遠隔リソースとしてメッセージの一部に含まれているかどうかの別を問わず、そのメッセージに関連するデータを識別する。SOAP ***Body*** には搬送内容データを含めないようにすることを推奨する。***Manifest*** の目的は以下の通りである。

- その ebXML メッセージに関連した特定の搬送内容を直接抽出しやすくする。
- アプリケーションが、パースを行う必要なく搬送内容にアクセスできるようにするかどうかを決定できるようにする。

***Manifest*** 要素は、以降の項で説明する以下の属性および要素から成る。

- ***id*** 属性

- 必須属性 ***version***(詳細は8.4.10項参照)
- 1 つ以上の ***Referenece*** 要素
- ***#wildcard***

### 8.11.1 id 属性

***Manifest*** 要素には、XML ID である ***id*** 属性が含まれていなければならない(8.2.5項参照)。

### 8.11.2 #wildcard 要素

#wildcard 要素の取り扱いに関する説明は、8.2.4項を参照のこと。

### 8.11.3 Reference 要素

***Reference***要素は、以下の従属要素から成る複合要素である。

- ***Schema*** - 親要素 ***Reference*** で識別されたインスタンス文書を定めるスキーマについての情報
- ***Description*** - 親要素 ***Reference*** で参照される搬送内容オブジェクトの説明
- ***#wildcard*** - 名前空間に適格な、外部名前空間に属する要素の内容

***Reference*** 要素自体は、単なる XLINK のリンクである。XLINK は、現在、W3C の勧告候補(CR)となっている。この状況での XLINK の使用は、関連性を説明する簡潔な語彙を用意するため選択されるに過ぎないことに注意すること。XLINK プロセッサやエンジンを使用する必要はないが、特定の実装には有益である場合がある。

***Reference*** 要素には、上述の要素の他に、以下の属性の内容が含まれる。

- ***id*** - ***Reference*** 要素の XML ID
- ***xlink:type*** - この属性は、XLINK 単純リンクとなる要素を定める。この値は、「simple」に固定される。
- ***xlink:href*** - この必須属性には、参照される搬送内容ブジェクトの URI となる値が含まれる。この属性の値は、[XLINK]仕様のシンプルリンクの規準に準拠していなければならない。
- ***xlink:role*** - この属性は、搬送内容オブジェクトやその目的を説明するリソースを識別する。この属性を含める場合、その値は[XLINK]仕様に準拠した有効な URI でなければならない。
- 名前空間に適格なその他の属性を含めることも可能である。受信側 MSH は、先に定められた名前空間以外の外部名前空間の属性を無視することができる。

#### 8.11.3.1 Schema 要素

参照される項目に、その項目を説明する何らかのスキーマが含まれている場合(例えば、XML スキーマ、DTD、またはデータベーススキーマ)、***Schema*** 要素は ***Reference*** 要素の子要素として表される。この要素は、親要素 ***Reference*** によって識別される搬送内容ブジェクトを定めるスキーマとそのバージョンを識別する方法を提供する。***Schema*** 要素には、以下の属性が含まれる。

- ***location*** - スキーマの必須 URI
- ***version*** – スキーマのバージョン識別子

#### 8.11.3.2 Description 要素

***Reference*** 要素には、任意で ***Description*** 要素を含めることができる。***Description*** は親要素 ***Reference*** が参照する搬送内容オブジェクトの説明である。説明の言語は、必須属性 ***xml:lang*** によって定められます。***xml:lang*** 属性は、[XML]で指定されている言語識別規則に従わなければならない。この要素は、親要素 ***Reference*** によって識別される搬送内容オブジェクトの説明を判読できるようにする。複数の ***Description*** 要素がある場合は、それぞれに一意の ***xml:lang*** 属性値が含まれていなければならない。***Description*** 要素の例を以下に示す。

```
    <eb:Description xml:lang=”en-gb”>Purchase Order for 100,000 widgets</eb:Description>
```

#### 8.11.3.3 #wildcard 要素

#wildcard 要素の取り扱いに関する説明は、8.2.4項を参照のこと。

### 8.11.4 Manifest に含まれる参照

***Manifest*** が参照する搬送内容データと、***xlink:role*** に使用する値は、ebXML メッセージングを利用するビジネスプロセスまたは情報交換の設計者が決定する。

### 8.11.5 Manifest の確認

***xlink:href*** 属性に内容の id (URI スキーム「`cid`」)である URI が含まれている場合、メッセージの*搬送コンテナ*にその `content-id` が付された MIME 部がなければならない。これがない場合は、***errorCode*** を ***MimeProblem***、***severity*** を ***Error*** に設定して *From 側*にエラーを報告する必要がある。

***xlink:href*** 属性に内容の id (URI スキーム「`cid`」)ではない URI が含まれており、URI を決定できない場合、エラーを報告するかどうかは実装ごとに決定される。エラーを報告する場合は、***errorCode*** を ***MimeProblem***、***severity*** を ***Error*** に設定して *From 側*に報告する必要がある。

### 8.11.6 Manifest サンプル

以下の記述は、搬送内容MIMEボディ部が1つ付されたメッセージの典型的な***Manifest***を示す。

```
<eb:Manifest eb:id="Manifest" eb:version="1.0">
  <eb:Reference eb:id="pay01"
    xlink:href="cid:payload-1"
    xlink:role="http://regrep.org/gci/purchaseOrder">
    <eb:Schema eb:location="http://regrep.org/gci/purchaseOrder/po.xsd" eb:version="1.0"/>
    <eb:Description xml:lang="en-us">Purchase Order for 100,000 widgets</eb:Description>
  </eb:Reference>
</eb:Manifest>
```

## 8.12 StatusRequest 要素

***StatusRequest*** 要素は、SOAP ***Body*** の子要素である。テータスが要求されている以前のメッセージを識別するのに使用される(9.1項参照)。

***StatusRequest*** 要素は、以下の要素および属性で構成される。

- 必須要素 ***RefToMessageId***
- 必須属性 ***version*** (詳細は8.4.10項参照)
- **id** 属性(詳細は8.2.5項参照)

### 8.12.1 StatusRequest サンプル

***StatusRequest*** 要素の例を以下に示す。

```
<eb:StatusRequest eb:version="1.0" >
    <eb:RefToMessageId>323210:e52151ec74:-7ffc@xtacy</eb:RefToMessageId>
</eb:StatusRequest>
```

## 8.13 StatusResponse 要素

***StatusResponse*** 要素は、前に送信したメッセージの処理状態の要求に応じるため、MSH の 1 つが使用する(9.1項も参照)。

***StatusResponse*** 要素は、以下の要素および属性から成る。

- 必須要素 ***RefToMessageId***
- ***Timestamp*** 要素

- 必須属性 ***version***(詳細は8.4.10項参照)
- ***messageStatus*** 属性
- ***id*** 属性(詳細は8.2.5項参照)

### 8.13.1 RefToMessageId 要素

必須要素 ***RefToMessageId*** 要素には、位置付けが報告されているメッセージの ***MessageId*** が含まれる。

### 8.13.2 Timestamp 要素

***Timestamp*** 要素には、位置付けが報告されているメッセージが受け取られた時間が含まれる(8.4.6.2項)。位置付けが報告されているメッセージが ***NotRecognized*** の場合や、要求が ***UnAuthorized*** の場合は、この要素を省かなければならない。

### 8.13.3 MessageStatus 属性

***messageStatus*** 属性は、***RefToMessageId*** 要素で識別されるメッセージの位置付けを識別する。この属性は、以下のいずれかの値に設定されなければならない。

- ***UnAuthorized*** – メッセージ位置付け要求は許可または承認されない。
- ***NotRecognized*** – ***StatusResponse*** 要素の ***RefToMessageId*** 要素で識別されたメッセージが認識されない。
- ***Received*** – MSH は、***StatusResponse*** 要素の ***RefToMessageId*** 要素で識別されたメッセージを受け取った。

注意: 照会されているメッセージの送信後、***persistDuration*** で示される時間が経過した後にメッセージ位置付け要求が送信された場合、メッセージ位置付け応答は、***MessageId*** が保存場所にないので ***MessageId*** は ***NotRecognized*** であることを示すことができる。

### 8.13.4 StatusResponse サンプル

以下に、***StatusResponse***要素の例を示す。

```
<eb:StatusResponse eb:version="1.0" eb:messageStatus="Received">
  <eb:RefToMessageId>323210:e52151ec74:-7ffc@xtacy</eb:RefToMessageId>
  <eb:Timestamp>2001-03-09T12:22:30Z</eb:Timestamp>
</eb:StatusResponse>
```

## 8.14 DeliveryReceipt 要素

***DeliveryReceipt*** 要素は、元のメッセージの送信元である *From 側*にメッセージを受け取ったことを示すため、*To 側*が使用するオプションの要素である。。***DeliveryReceipt*** 要素を含むメッセージの ***RefToMessageId*** 要素は、***MessageId*** によるメッセージ受領通知を識別するのに使用される。

***DeliveryReceipt*** 要素は、以下の要素および属性から成る。

- **id** 属性(8.2.5項参照)
- 必須属性 ***version***(詳細は8.4.10項参照)
- ***Timestamp*** 要素
- 1 つ以上の ***ds:Reference*** 要素

### 8.14.1 Timestamp 要素

***Timestamp*** 要素は、***DeliveryReceipt*** 要素が生成されているメッセージを *To 側*が受け取った時間を表す値である。これは、[XMLSchema]の timeInstant に準拠していなければならない。

### 8.14.2 ds:Reference element

受領通知は、受領が通知されているメッセージから取得または派生した[XMLDSIG]名前空間(http://www.w3.org/2000/09/xmldsig#)の ***Reference*** 要素を 1 つ以上含めることで、MSH が受信を拒否できないようにするために使用できる。***Reference*** 要素は、前述の名前空間に適格な名前空間でなければならず、また、XML 署名[XMLDSIG]の仕様に準拠していなければならない。

### 8.14.3 DeliveryReceipt サンプル

以下の記述は、***DeliveryReceipt*** 要素の例である。

```
<eb:DeliveryReceipt eb:version="1.0">
  <eb:Timestamp>2001-03-09T12:22:30Z</eb:Timestamp>
  <ds:Reference URI="cid://blahblahblah/">
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
    <ds:DigestValue>...</ds:DigestValue>
  </ds:Reference>
</eb:DeliveryReceipt>
```

## 8.15 ebXML SOAP 拡張要素の結合

本項では、さまざまな ebXML SOAP 拡張要素を組み合わせて使用する方法を説明する。

### 8.15.1 Manifest 要素

*ヘッダコンテナ*に含まれていないメッセージに関連するデータがある場合は、***Manifest*** 要素が含まれていなければならない。これは、特に*搬送コンテナ*や、例えばウェブなど、それ以外の場所にあるデータに適用される。

### 8.15.2 MessageHeader 要素

***MessageHeader*** 要素は、すべてのメッセージに含まれていなければならない。

### 8.15.3 TraceHeaderList 要素

***TraceHeaderList*** 要素は、どのメッセージにも含めることができる。信頼性の高い方法(10項)、または複数のホップ(8.5.4項参照)でメッセージが送信される場合、この要素が含まれていなければならない。

### 8.15.4 StatusRequest 要素

***StatusRequest*** 要素は、以下の要素と一緒に含めることはできない。

- ***Manifest*** 要素
- ***ErrorList*** 要素

### 8.15.5 StatusResponse 要素

この要素は、以下の要素と一緒に含めることはできない。

- ***Manifest*** 要素
- ***StatusRequest*** 要素
- ***HighestSeverity*** 属性が ***Error*** に設定された ***ErrorList*** 要素

### 8.15.6 ErrorList 要素

***ErrorList*** の ***highestSeverity*** 属性が ***Warning*** に設定されている場合は、他の要素と一緒にこの要素を含めることができる。

***ErrorList*** の ***highestSeverity*** 属性が ***Error*** に設定されている場合は、以下の要素と一緒にこの要素を含めることはできない。

- ***Manifest*** 要素
- ***StatusResponse*** 要素

### 8.15.7 Acknowledgment 要素

***Acknowledgement*** 要素は、どのメッセージにも含めることができる。

### 8.15.8 Delivery Receipt 要素

***DeliveryReceipt*** 要素は、どのメッセージにも含めることができる。

### 8.15.9 Signature 要素

***ds:Signature*** 要素は、どのメッセージにも 1 つ以上含めることができる。

### 8.15.10 Via 要素

***Via*** 要素は、どのメッセージにも 1 つだけ含めることができる。

# 9 メッセージ取扱サービスハンドラのサービス

メッセージ取扱サービスハンドラは、メッセージハンドリングサービスの実装の円滑な運用のため、次の 2 つのサービスをサポートすることが可能である。

- メッセージ位置付け要求
- メッセージ取扱サービスハンドラピン

要求されたサービスをサポートしない場合、*受信側 MSH* は、***faultCode*** を ***MustUnderstand*** に設定して SOAP 失敗を返さなければならない。以下にそれぞれのサービスについて説明する。

## 9.1 メッセージ位置付け要求サービス

メッセージ位置付け要求サービスは、以下のものから成る。

- 前に送信されたメッセージに関する詳細を含むメッセージ位置付け要求メッセージは、メッセージ取扱サービスハンドラ(MSH)に送信される。
- 要求を受け取るメッセージ取扱サービスハンドラは、メッセージ位置付け応答メッセージで応答する。

メッセージ取扱サービスハンドラは、信頼性の高い方法(10項参照)で送信されたメッセージのメッセージ位置付け要求に応答しなければならず、また、***RefToMessageId*** の ***MessageId*** は*永続的な保存場所*(10.1.1項参照)に含まれる。

メッセージ取扱サービスハンドラは、信頼性の高い方法で送信されていないメッセージ位置付け要求に応答することも可能である。

また、メッセージ取扱サービスは、高信頼性メッセージングのためにメッセージ位置付け要求を使用してはならない。

### 9.1.1 メッセージ位置付け要求メッセージ

メッセージ位置付け要求のメッセージは、SOAP ***ヘッダ***に *ebXML 搬送内容*および以下の要素が含まれていない ebXML メッセージから成る。

- ***MessageHeader*** 要素
- ***TraceHeaderList*** 要素
- ***StatusRequest*** 要素
- ***ds:Signature*** 要素

***TraceHeaderList*** および ***ds:Signature*** 要素は、省略することができる(8.5項および8.15.8項参照)。

***MessageHeader*** 要素には、以下の要素が含まれていなければならない。

- メッセージ位置付け要求のメッセージを作成した*当事者*を識別する ***From*** 要素
- メッセージを受け取ることになっている*当事者*を識別する ***To*** 要素。位置付け確認済みのメッセージに ***TraceHeader*** がある場合、これはそのメッセージからの ***Receiver*** でなければならない。***Receiver*** 要素内の ***PartyId*** 要素は、すべてこの ***To*** 要素に含まれていなければならない。
- ***uri:www.ebxml.org/messageService*** を含んだ ***Service*** 要素
- ***StatusRequest*** を含んだ ***Action*** 要素。

メッセージは ***To*** 側に送信される。

SOAP ***Body*** の ***StatusRequest*** 要素内の ***RefToMessageId*** 要素には、位置付けが照会されているメッセージの ***MessageId*** が含まれる。

### 9.1.2　メッセージ位置付け応答メッセージ

メッセージ位置付け要求メッセージを受け取ったら、*To 側*は、ebXML 搬送内容および以下の要素で構成されないメッセージ位置付け応答メッセージを SOAP ***ヘッダ***および***ボディ***に生成しなければならない。

- ***MessageHeader*** 要素
- ***TraceHeaderList*** 要素
- ***Acknowledgment*** 要素
- ***StatusResponse*** 要素(8.13項参照)
- ***ds:Signature*** 要素

***TraceHeaderList***、***Acknowledgement***、および ***ds:Signature*** 要素は省略することができる(8.5項、8.15.7項、および8.15.8項参照)。

***MessageHeader*** 要素には以下のものを含まなければならない。

- メッセージ位置付け応答メッセージの送信者を識別する ***From*** 要素
- メッセージ位置付け要求メッセージの ***From*** 要素の値に設定された ***To*** 要素
- ***uri:www.ebxml.org/messageService/***の値が含まれた ***Service*** 要素
- ***StatusResponse*** を含む ***Action*** 要素
- メッセージ位置付け要求メッセージを識別する ***RefToMessageId***

メッセージは *To 側*に送信される。

### 9.1.3　セキュリティについて

メッセージ位置付け要求のメッセージを受け取る当事者は、必ずメッセージに応答しなければならない。しかし、メッセージの送信者が権限を持っていないことが考えられる場合、応答する代わりに、***messageStatus*** を ***Unauthorized*** に設定してメッセージを無視することも可能である。この場合の処理を決める方法は、実装ごとに異なる。

## 9.2　メッセージ取扱サービスハンドラのピンサービス

メッセージ取扱サービスハンドラピンサービスにより、MSH は他の MSH が動作しているかどうか調べることが可能になる。このサービスは次のように動作する。

- MSH にメッセージ取扱サービスハンドラピンメッセージを送信
- ピンメッセージを受け取った MSH がメッセージ取扱サービスハンドラポンメッセージで応答

### 9.2.1　メッセージ取扱サービスハンドラのピンメッセージ

メッセージ取扱サービスハンドラピン(MSH Ping)メッセージは、ebXML 搬送内容が含まれていない *ebXML メッセージ*と、SOAP ***ヘッダ***の以下の要素から成る。

- ***MessageHeader*** 要素
- ***TraceHeaderList*** 要素
- ***ds:Signature*** 要素

***TraceHeaderList*** および ***ds:Signature*** 要素は省略することができる(8.5項および8.15.8項参照)。

***MessageHeader*** 要素には、以下が含まれていなければならない。

- MSH ピンメッセージを作成した*当事者*を識別する ***From*** 要素
- MSH ピンメッセージを受け取る*当事者*を識別する ***To*** 要素
- ***CPAId*** 要素
- ***ConversationId*** 要素
- ***uri:www.ebxml.org/messageService/***を含む ***Service*** 要素

- ***Ping*** を含む ***Action*** 要素

メッセージは *To 側*に送信される。

### 9.2.2　メッセージ取扱サービスハンドラのポンメッセージ

MSH Ping メッセージを受け取った後、*To 側*は、ebXML 搬送内容が含まれていない ebXML メッセージと、SOAP ***ヘッダ***の以下の要素から成るメッセージ取扱サービスハンドラポンメッセージ(MSH Pong)を生成することができる。

- ***MessageHeader*** 要素
- ***TraceHeaderList*** 要素
- ***Acknowledgment*** 要素
- オプションの ***ds:Signature*** 要素

***TraceHeaderList***、***Acknowledgement***、および ***ds:Signature*** 要素は、省略することができる(8.5項、8.15.7項、および8.15.8項参照)。

***MessageHeader*** 要素には、以下が含まれていなければならない。

- MSH Pong メッセージの作成者を識別する ***From*** 要素
- MSH Ping メッセージを生成した*当事者*を識別する ***To*** 要素
- ***CPAId*** 要素
- ***ConversationId*** 要素
- ***uri:www.ebxml.org/messageService/***の値を含む ***Service*** 要素
- ***Pong*** の値を含む ***Action*** 要素
- MSH Ping メッセージを識別する ***RefToMessageId***

メッセージは、*To 側*に送信される。

### 9.2.3　セキュリティについて

MSH Ping メッセージの受信側は、必ずメッセージに応答しなければならない。しかし、MSH Ping メッセージを使用して、MSH のセキュリティ攻撃の一部としてメッセージ取扱サービスハンドラの有無を明らかにしようとする当事者が存在する危険もある。従って、MSH Ping の受信者は、受信したメッセージの送信者が権限を持っていないことが考えられる場合、あるいはメッセージが何らかの攻撃の一部であると考えられる場合、メッセージを無視することも可能である。この場合の処理を決める方法は、実装ごとに異なる。

# 10 高信頼性メッセージング

高信頼性メッセージングは、2 つのメッセージ取扱サービスハンドラ(MSH)が「高信頼性メッセージング」の意味情報を使用して送信されたメッセージを「高い信頼性」で交換することによって、*To 側*がメッセージを 1 度だけ受け取ればよくなるような、相互運用可能なプロトコルを定める。

高い信頼性は、*受信側 MSH* が *受領通知メッセージ*付きでメッセージに応答することによって実現される。

### 10.1.1 永続的記憶装置★Persistent Storage★とシステム障害

高信頼性メッセージングをサポートする MSH は、高い信頼性で送受信されたメッセージを*永続的記憶装置*に保存しておかなければならない。ここで、*永続的記憶装置*とは、システムの障害や中断後も情報を失うことのないデータ保存方法を指する。

本書では、データを残しておくために使用される技術によって、信頼性の程度は異なることをが認識されている。しかし、少なくともハードディスク(もしくはこれと同等)の信頼性を有する永続的記憶装置を使用する必要がある。本書の実装者には、ハードウェアやソフトウェアコンポーネントの障害に強い技術を使用することを強く推奨する。

MSH は、システムの中断または障害が発生しても、問題が発生しなかった場合と同様に、確実に処理されなければならない。これをどのように行うかは、実装ごとに決定される。

複製メッセージのフィルタリングをサポートするため、*受信側 MSH* は ***MessageId*** を*永続的記憶装置*に保存しておく必要がある。また、*永続的記憶装置*には以下の情報も保存しておくことを推奨する。

- 完全なメッセージ(少なくともアプリケーションや、メッセージの処理に必要なその他のプロセスにメッセージが渡されるまで)
- メッセージ位置付け要求(9.1項)への応答を生成するのに使用できるように、メッセージを受信した日時
- 完全な応答メッセージ

### 10.1.2 高信頼性メッセージングの実装方法

高信頼性メッセージングのサポートは、以下のいずれかの方法で実装できる。

- ebXML 高信頼性メッセージ取扱プロトコルを使用する
- 別のプロトコルを使用して、信頼度の高いメッセージ配信を実現する市販のソフトウェアとともに、ebXML SOAP 構造を使用する

## 10.2 高信頼性メッセージングパラメータ

本項では、高信頼性メッセージングの管理に必要なパラメータについて説明する。このパラメータの情報は、*CPA* または ***MessageHeader*** (8.4.2項)で指定することができる。

### 10.2.1 配信意味情報

*From 側*MSH は、***deliverySemantics*** の値を使用して、信頼度の高い方法でメッセージを送信すべきかどうかを示さなければならない。有効な値は以下の通りである。

- ***OnceAndOnlyOnce*** - メッセージは ***reliableMessagingMethod*** を使用して送信されなければならない。この場合、*To 側*のアプリ－ケーションまたはその他のプロセスはメッセージを 1 回だけ受け取ることになる。
- ***BestEffort*** - 信頼度の高い配信意味情報は使用されない。この場合、***reliableMessagingMethod*** の値は無視される。

***deliverySemantics*** の値は、CPA または ***MessageHeader*** (8.4.2項)で指定される。***deliverySemantics*** の既定値は ***BestEffort*** となる。

***deliverySemantics*** が ***OnceAndOnlyOnce*** に設定されている場合、*From 側*および *To 側*の MSH は、障害が発生し、複製が無視された際にメッセージを再送する方法について説明する高信頼性メッセージングを採用しなければならない。***deliverySemantic*** に ***OnceAndOnlyOnce*** という値が指定されると、複製メッセージは無視される。

***deliverySemantics*** が ***BestEffort*** に設定されている場合、配信不可能なメッセージを受け取った MSH は、障害回復の処理を行ったり、他の方法でその問題について通知したりすることはできない。メッセージを送信する MSH は、一切、障害の回復を試みてはならない。これは、メッセージの複製がアプリケーションに配信される可能性があること、そしてメッセージの永続的記憶は必要ないことを意味する。

要求された配信意味情報をサポートできない場合、*To 側*は、***ErrorCode*** に ***NotSupported***、***Severity*** に ***Error*** を使用して *From 側*にエラーを報告しなければならない。

### 10.2.2 mshTimeAccuracy

***mshTimeAccuracy*** パラメータは、例えば ***TimeToLive*** の確認の際に*受信側 MSH* が使用する時計の最低精度を示す。分および秒の精度は、「mm:ss」の形式で示される。

### 10.2.3 TimeToLive

***TimeToLive*** の値は、*To 側*にメッセージが配信され、処理されなければならない期限を表す。これは XML スキーマの timeInstant に従っていなければならない。

*受信側 MSH* の内部時計の時刻がメッセージの ***TimeToLive*** の値を超えた場合、***TimeToLive*** は期限切れとなる。

***TimeToLive*** の値を設定する際は、*From 側*の MSH が、それ自体の内部時計の精度とともに、*受信側 MSH* の内部時計の精度を示す MSH ***TimeAccuracy*** パラメータを考慮するようにすることを推奨する。MSH が内部時計の精度をどの程度に保つようにするかについては、実装ごとに決定される。

***TimeToLive*** の期限が切れたメッセージを受け取った場合、*To 側*の MSH は *From 側*の MSH にメッセージを送信し、メッセージの ***TimeToLive*** が期限切れであることを報告しなければならない。このメッセージは、***errorCode*** 属性が ***TimeToLiveExpired***、***severity*** 属性が ***Error*** に設定されたエラーを含んだ ***ErrorList*** で構成されていなければならない。

### *10.2.4* reliableMessagingMethod

***reliableMessagingMethod*** 属性は、以下のいずれかの値がなければならない。

- ***ebXML***
- ***Transport***

この属性の規定値は ***ebXML*** で、大文字と小文字が区別される。この属性の使用については、8.7.4項を参照のこと。

### 10.2.5 ackRequested

***AckRequested*** の値は、*送信側 MSH* が、*受信側 MSH* に ***type*** が ***Acknowledgement*** の ***Acknowledgement*** 要素が付された*受領通知メッセージ*の返信を求めるのに使用する。

有効な ***AckRequested*** の値は以下の通りである。

- ***Unsigned*** - 署名なしの受領通知が要求される。
- ***Signed*** - 署名付きの受領通知が要求される。
- ***None*** - 受領通知が要求されていないことを示す。

既定値は ***None*** である。

### 10.2.6 retries

***retries*** の値は、同じ通信プロトコルを使用して*送信側 MSH* が受領通知未収メッセージの再配信を試行しなければならない最大回数を指定する整数値である。

### 10.2.7 retryInterval

***retryInterval*** の値は、[XMLSchema]の timeDuration データ種類に従った期間で表される時間である。この値は、*受領通知メッセージ*が届かない場合に、*送信側 MSH* が次の再配信を行うまでに取るべき最少の時間間隔を定める。

### 10.2.8 persistDuration

***persistDuration*** は、[XMLSchema]の timeDuration で表される時間で、*受信側 MSH* が信頼度の高い方法で送信された*メッセージ*のデータを*永続的記憶装置*に保存しておく最低の期間を定める。

最初にメッセージが送信されてからすでに ***persistDuration*** が過ぎている場合、*送信側 MSH* は同じ ***MessageId*** 付きのメッセージを再送してはならない。

***persistDuration*** が経過する前にメッセージを正しく送信できなかった場合、*送信側 MSH* は配信失敗を報告しなければならない(10.4項)。

## 10.3 ebXML 高信頼性メッセージ取扱プロトコル

***deliverySemantics*** パラメータ/要素が ***OnceAndOnlyOnce*** に、***reliableMessagingMethod*** パラメータ/要素が ***ebXML*** に設定されている場合は(既定値)、本項で説明される ebXML 高信頼性メッセージ取扱プロトコルに従わなければならない。

ebXML 高信頼性メッセージ取扱プロトコルは図 10-1 の通りである。

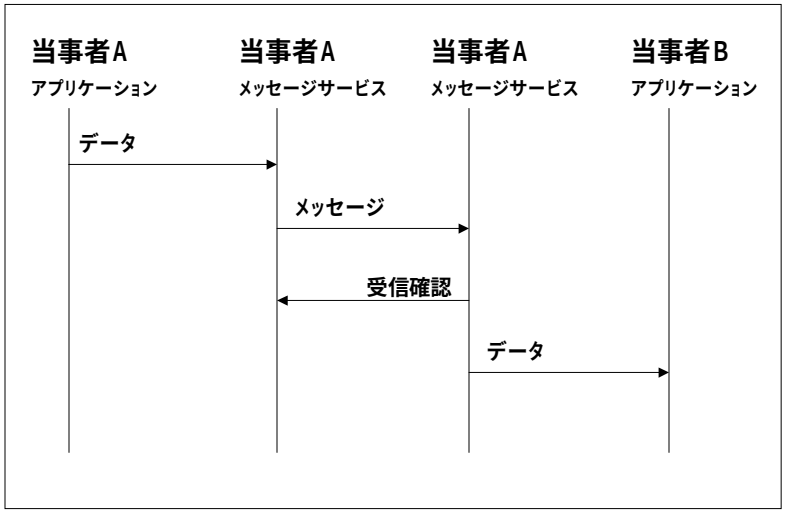



**図10-1 メッセージの受領通知**

*受領通知メッセージ*の受領書は、*受信側 MSH* が受領通知の行われたメッセージを正しく受け取ったうえ、処理または保存したことを示す。

*受領通知メッセージ*には、*受領通知が行われたメッセージの* ***MessageId*** 要素と同じ値を含む ***RefToMessageId*** 付きの ***MessageData*** 要素が含まれていなければならない。

### 10.3.1 メッセージ送信の動作

信頼度の高い方法で送信する必要のあるデータをアプリケーションから受け取った場合 (***deliverySemantics*** が ***OnceAndOnlyOnce*** に設定されている場合)、MSH は以下のことを行わなければならない。

1. 8.5.2項の ***TraceHeader*** 要素で説明されている通り、アプリケーションから受け取った送信側と受信側の URI を識別する ***TraceHeader*** 要素を含んだコンポーネントからメッセージを作成する。
2. *永続的記憶装置*にメッセージを保存する(10.1.1項参照)。
3. *受信側 MSH* にメッセージを送信する。
4. *受信側 MSH* は*受領通知メッセージ*が返信されるまで待つ。*受領通知メッセージ*が返信されない場合、または一時的なエラーが返された場合は、10.3.4項の説明に従って適切に処理する。

### 10.3.2 メッセージ受信の動作

受信したメッセージの ***deliverySemantics*** が ***OnceAndOnlyOnce*** に設定されている場合は、以下のことを行う。

1. メッセージが単なる受領通知の場合(***Service*** 要素が http://www.ebxml.org/namespace/messageService/MessageAcknowledgement に設定され、***Action*** が ***Acknowledgement*** に設定されている場合)は、以下のことを行う。
   a) 受信したメッセージの ***RefToMessageId*** と同じ値の ***MessageId*** がある*永続的記憶装置*内のメッセージを探し出す。
   b) *永続的保存装置*内でメッセージが見つかった場合は、保存されているメッセージを配信済みとしてマークする。
2. メッセージが受領通知でない場合は、メッセージが複製かどうかを確認する(例えば、その ***MessageId*** と同じ値の ***MessageId*** が*永続的保存装置*に保存されているかどうか)。
   c) メッセージが複製でない場合は、以下のことを行う。
      i) 受信したメッセージの ***MessageId*** を*永続的記憶装置*に保存する。必要に応じ、実装ごとにメッセージ全体を保存しておくこともできる。
      ii) 受信したメッセージに ***RefToMessageId*** 要素が含まれている場合は、以下のことを行う。
         (1) 受信したメッセージの ***RefToMessageId*** と同じ値の ***MessageId*** がある*永続的記憶装置*内のメッセージを探し出す。
         (2) *永続的保存装置*内でメッセージが見つかった場合は、保存されているメッセージを配信済みとしてマークする。
      iii) *受領通知メッセージ*を生成し、返信する(10.3.3項参照)。
   d) メッセージが複製の場合は、以下のことを行う。
      i) 受信したメッセージに対する最初の応答(受領したメッセージの ***MessageId*** に一致する ***RefToMessageId*** が含まれる)を永続的記憶装置から探し出し、これに返信する。

ii) *永続的保存装置*内でメッセージが見つかった場合は、受信したメッセージを送信した MSH に、保存されているメッセージを送信しなおす。

iii) *永続的記憶装置*内にメッセージがない場合は、以下のことを行う。

(1) ***syncReply*** が ***True*** に設定され、CPA でアプリケーションの応答が含まれていることが示されている場合(そのメッセージに対してメッセージが生成されていないか、もしくは前のメッセージの処理が完了していない場合)、受領したメッセージを無視する。

(2) ***syncReply*** が ***False*** に設定されている場合は、*受領通知メッセージ*を生成する(10.3.3項参照)。

### 10.3.3 受領通知メッセージの生成

以下のメッセージを受信した場合は、必ず*受領通知メッセージ*が生成されなければならない。

- deliverySemantics が OnceAndOnlyOnce に設定されているメッセージ
- reliableMessagingMethod が ebXML に設定されているメッセージ(既定値)

*受領通知メッセージ*には、少なくとも*受領通知を行うメッセージ*の ***MessageId*** と同じ値を含む ***RefToMessageId*** の ***MessageData*** が含まれていなければならない。

受信したメッセージの ***Via*** の ***ackRequested*** が ***Signed*** または ***Unsigned*** に設定されている場合、受領通知メッセージには ***Acknowledgement*** 要素も含まれていなければならない。

***syncReply*** パラメータの値によっては、受信したメッセージに対する返信と同時に*受領通知メッセージ*を送信することも可能である。この場合、*受領通知メッセージ*の ***MessageHeader*** 要素は、サービスの設計者が定めることになる。

***Acknowledgement*** 要素だけで送信される場合、***MessageHeader*** 要素は以下のように設定しなければならない。

- Service 要素は uri:www.ebxml.org/messageService/に設定しなければならない。
- Action 要素は Acknowledgement に設定しなければならない。
- From 要素には受信したメッセージから抽出した To 要素を含めるか、または、最後に受信したメッセージの TraceHeader の Receiver を使用して設定することができる。どちらの場合も、この From 要素には、受信したメッセージの PartyId 要素がすべて含まれていなければならない。
- To 要素には受信したメッセージから抽出した From 要素を含めるか、または、最後に受信したメッセージの TraceHeader の Sender に設定することができる。どちらの場合も、この To 要素には、受信したメッセージの PartyId 要素がすべて含まれていなければならない。
- RefToMessageId 要素は、最後に受信したメッセージの MessageId に設定しなければならない。

### 10.3.4 消失したメッセージの再送と複製のフィルタリング

本項では、メッセージが消失した場合に必要となるメッセージの送信者および受信者の処理について説明する。送信側 MSH にメッセージに対する返信が届かない場合、メッセージは「消失」したことになる。例えば、次のように、メッセージが消えた可能性がある。

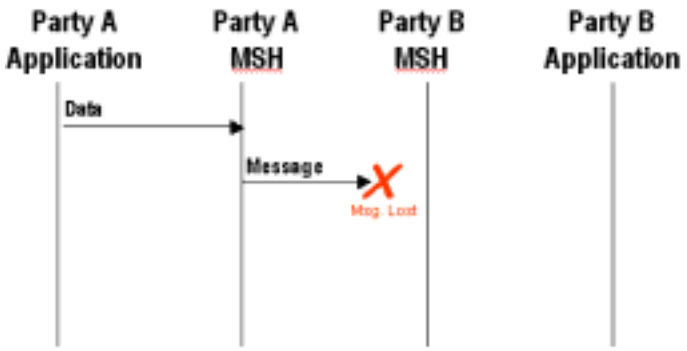



**図10-2 未配信メッセージ**

また、例えば以下のように*受領通知メッセージ*が消失した可能性もある。


**図10-3 受領通知メッセージの消失**

適用される規則は以下の通りである。

- *受信側 MSH* から*受領通知メッセージ*が届かず、以下の両方に該当する場合、*送信側 MSH* は元のメッセージを送信しなおさなければならない。

  a) メッセージを最後に送信しなおしてから、***retryInterval*** で指定する時間が経過した場合

  b) メッセージを再送した回数が、***retries*** パラメータで指定する回数に達していない場合。

- 再試行回数に達した後も送信側 MSH に*受領通知メッセージ*が届かない場合、*送信側 MSH* はアプリケーションやシステム管理機能に受領通知が未収であることを通知しなければならない。
- 転送プロトコルレベルで回復不可能な通信プロトコルのエラーを検出した場合、*送信側 MSH* はメッセージを送信しなおさなければならない。

### 10.3.5 複製メッセージの処理

この仕様における複製メッセージの定義は以下の通りである。

- 「同一メッセージ」とは、***TraceHeader*** 要素が追加される場合のあることを除き、送信されたメッセージと同じ *ebXML* SOAP ***ヘッダ***、***ボディ***、*そして ebXML 搬送内容*が含まれるメッセージである。
- 「複製メッセージ」とは、受信したメッセージと同じ ***MessageId*** が含まれるメッセージである。
- 「初回メッセージ」とは、複製メッセージと同じ ***RefToMessageId*** を持った ***MessageData*** 要素内の ***Timestamp*** が最も小さいメッセージである。

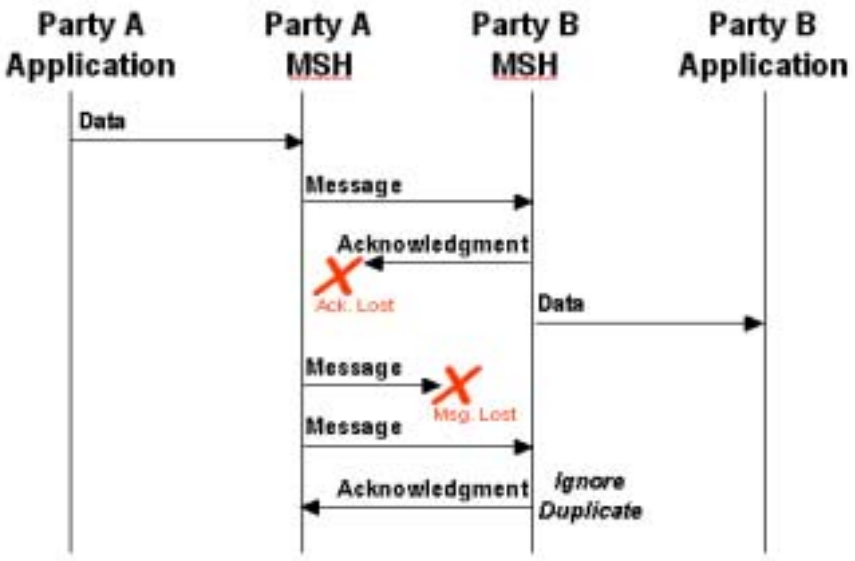



**図10-4 受領通知が行われていないメッセージの再送**

上の図および以下の説明は、***deliverySemantics*** を ***OnceAndOnlyOnce*** に設定して送信を行う*送信側 MSH* および*受信側 MSH* が従わなければならない動作を示す。

1) *受領通知メッセージ*が届かない場合、*メッセージ*の送信者(当事者 A)は「同一メッセージ」を送信しなおさなければならない。
2) 「複製メッセージ」を受け取ったら、*メッセージ*受信者(当事者 B)は送信者(当事者 A)に送信された*初回メッセージ*と同じメッセージを送信者(当事者 A)に送信しなおさなければならない。
3) *メッセージ*受信者(当事者 B)は、アプリケーション/プロセスに再度メッセージを転送してはならない。

## 10.4 メッセージ配信の失敗

***deliverySemantics*** を ***OnceAndOnlyOnce*** に設定して送信されたメッセージが配信されなかった場合、MSH またはプロセスは *From 側当事者*に配信失敗の通知を送らなければならない。配信失敗通知メッセージには以下が含まれる。

- 問題を検知した*当事者*を識別する ***From*** 要素
- 配信不可能なメッセージを作成した *From 側*を識別する ***To*** 要素
- 11.5項で説明される ***Serivice*** および ***Action*** 要素
- 以下の severity の ***Error*** 要素
  - 問題を検出した当事者がメッセージを送信できない場合は ***Error*** (例えば、通信トランスポートが使用できない場合)

- メッセージは送信されたものの、*受領通知メッセージ*が届かない場合は ***Warning***。これは、メッセージ配信の可能性が低く、メッセージが配信されていない公算が大きいことを意味する。

- ***DeliveryFailure*** の ***ErrorCode***

***ErrorCode*** が ***DeliveryFailure*** に設定された ***Error*** 要素付きのエラーメッセージは、何らかの理由で正しく配信できない可能性がある。そのような場合、エラーメッセージの最終的な配信先である *From 側*には、その他の方法で問題について通知しなければならない。これをどのように行うかは本書の範囲外である。

# 11 エラーの報告および処理

本項では、ebXML メッセージで検出したエラーを ebXML メッセージ取扱サービスハンドラ(MSH)が別の MSH に報告する方法を説明する。*ebXML メッセージ取扱サービス*のエラー報告および処理は、SOAP プロセッサレイヤの上のレイヤと考えられる。これは、ebXML MSH が SOAP プロセッサの観点では本質的にアプリケーションレベルの *SOAP メッセージ*ハンドラであることを意味する。メッセージの処理が不可能な場合、SOAP プロセッサは SOAP ***失敗***メッセージを生成することができる。*送信側 MSH* は、これらの SOAP ***失敗***メッセージを受け取って処理できるようになっていなければならない。

ebXML MSH ソフトウェアが SOAP 失敗メッセージを生成し、*SOAP メッセージ*の送信者に返信することもできる。この場合、返信されるメッセージは、[SOAP]仕様の SOAP ***失敗***メッセージの処理に関する指針に従っていなければならない。

***highestSeverity*** が ***Warning*** に設定されたエラーを報告する ebXML *SOAP メッセージ*は、SOAP ***失敗*** メッセージとして報告したり、返信したりすることはできない。

## 11.1 定義

混乱を避けるため、本項で使用される 2 つの句を以下に定義する。

- 「エラーメッセージ」 - 何らかのエラーが含まれる、あるいはエラーの原因となる*メッセージ*
- 「エラー報告メッセージ」 - エラーメッセージで発見されたエラーについて述べた ebXML ***ErrorList*** 要素が含まれる*メッセージ*

## 11.2 エラーの種類

一方の MSH は、例えば以下に関連するエラーなど、エラーメッセージの MSH エラーをもう一方の MSH に報告する必要がある。

- ebXML 名前空間に適格な *SOAP メッセージ*文書の内容(8項参照)
- 高信頼性メッセージングの失敗(10項参照)
- セキュリティ(**エラー! 参照元が見つかりません。**項参照)

この仕様では、これ以降、特に断りのない限り、「エラー」はすべて上述のエラー種類であるものとする。

データ通信プロトコル関連のエラーは、そのデータ通信プロトコルがサポートする標準的な仕組みで検出、報告され、ここに述べられているエラー報告方法は使用されない。

## 11.3 エラーメッセージが生成されるとき

以下のような場合、MSH によってメッセージ内にエラーが検出されたら、エラーが含まれたメッセージを送信した MSH にエラーを報告することを推奨する。

- エラー報告メッセージを送信すべきエラー報告場所(11.4項参照)が決定できる場合
- エラーメッセージに ***highestSeverity*** が ***Error*** に設定された ***ErrorList*** 要素が含まれていない場合

エラー報告場所が見つからない場合、または ***highestSeverity*** が ***Error*** に設定された ***ErrorList*** 要素がエラーメッセージに含まれている場合、以下のことを行うことを推奨する。

- エラーをログに記録する
- その他の方法でエラーを解決する
- それ以上処理を行わない

### 11.3.1 セキュリティについて

ヘッダにエラーが含まれたメッセージを受け取る当事者は、必ずメッセージに応答しなければならない。しかし、受信したメッセージが正式なものでなかったり、何らかのセキュリティ攻撃の一部であることが考えられる場合、メッセージを無視し、応答しないようにすることも可能である。この場合の処理を決める方法は、実装ごとに異なる。

## 11.4 エラー報告場所の確認

エラー報告場所とは、エラーメッセージの送信者が定めた URI で、*エラー報告メッセージ*の送信先を示す。

メッセージには、*CPA で示され*、メッセージの ***CPAId*** で識別される ***ErrorURI*** を使用しなければならない。CPA に ***ErrorURI*** が示されておらず、エラーメッセージに ***TraceHeaderList*** が含まれている場合は、一番上にある ***TraceHeader*** の ***Sender*** にある ***Location*** 要素の値を使用しなければならない。さもないと、***ErrorURI*** を解決するため、受領者はエラーメッセージの ***From*** 要素を使用してしまう可能性がある。これが不可能な場合、送信側*当事者*にエラーは報告されない。

エラーメッセージを十分に分析して調べられない場合でも、MSH 実装者は他の方法でエラー報告場所を決定するよう試みなければならない。どのようにこれを実施するかは、実装ごとに決定される。

## 11.5 Service および Action 要素の値

***ErrorList*** 要素は、前に送信されたメッセージの処理によって送信される*メッセージ*の SOAP ***Header*** に含めることができる。この場合、***Service*** および ***Action*** 要素の値は、サービスの設計者が定める。

また、***ErrorList*** 要素は、前に送信されたメッセージの処理によって送信されない SOAP ***Header*** に含めることもできる。この場合、***highestSeverity*** が ***Error*** に設定されていれば、***Service*** および ***Action*** 要素の値を以下のように設定しなければならない。

- ***Service*** 要素は、***uri:www.ebxml.org/messageService/***に設定しなければならない。
- ***Action*** 要素は ***MessageError*** に設定しなければならない。

***highestSeverity*** が ***Warning*** に設定されている場合は、***Service*** および ***Action*** 要素を使用することはできない。

# 12 セキュリティ

*ebXML メッセージ取扱サービス*では、その性質上、ある程度のセキュリティリスクが生じる。メッセージ取扱サービスは、以下のリスクに晒される可能性がある。

- 権限のないアクセス
- データの完全性や機密性に対する攻撃(間接的攻撃)
- サービス拒否およびスプーフィング

各セキュリティリスクについては、ebXML テクニカルアーキテクチャセキュリティ仕様[ebTASEC]で詳しく説明している。

これらのセキュリティリスクは、本項で述べる対策により、1 つまたは複数のアプリケーションを組み合わせ、全体または部分ごとに対処できる。本書では、選り抜きの対処方法を組み合わせた一式のプロファイルを説明する。これらの対策は、一般的に使用可能な技術に基づいて主なリスクに対処するために選び出されたものである。定められたプロファイルには、それぞれ対処されないリスクの説明を含む。

対策は、内在するリスクとリスクに晒されている可能性のある資産価値の評価に合わせて適用する必要がある。

## 12.1 セキュリティと管理

いかに技術が進歩しようとも、効果的なセキュリティ管理方針の適用と実践に代わる技術はない。

*ebXML メッセージ取扱サービス*のサイト管理者には、セキュリティの仕組み、サイトの(または物理的な)セキュリティ手続、暗号プロトコルのサポートと維持に努め、実装を更新し、必要に応じて調整することを強く推奨する。(http://www.cert.org/および http://ciac.llnl.gov/参照)

## 12.2 コラボレーションプロトコル合意書

MSH のセキュリティの設定は、CPA で定めることができる。CPA の 3 つの領域では、以下のようにセキュリティを定義する。

- 文書交換の節は、メッセージの搬送内容に適用するセキュリティについて述べたものである。。MSH はこのレベルで定められているセキュリティに対し責任を負わないが、メッセージの送信者にこれらのサービスを提供することは可能である。
- メッセージの節は、ebXML 文書全体に適用されるセキュリティについて述べたもので、これにはヘッダや搬送内容が含まれる。

## 12.3 対策技術

### 12.3.1 永続的デジタル署名

ebXML メッセージにデジタル署名を付すのに署名が使用される場合は、XML 署名[DSIG]を使用して、ebXML SOAP ***ヘッダ***と***ボディ***を ebXML 搬送内容またはそのメッセージに関連するウェブ上のデータに結合しなければならない。また、搬送内容自体へのデジタル署名にも XML 署名を使用することを強く推奨する。

ebXML メッセージ(ebXML SOAP ***ヘッダ***、***ボディ***、および関連する搬送内容オブジェクト)にデジタル署名を付すために使用可能な技術は、唯一、W3C/IETF XML 共同署名仕様[XMLDSIG]に準拠した技術で提供される。この仕様に準拠した XML 署名は、部分的に XML 文書に署名を付し、署名の有効性を保ちつつ文書を拡大していくことが可能である。

デジタル署名が必要な ebXML メッセージは、本項で定められたプロセスに従って署名を付さなければならず、また、[XMLDSIG]に完全に準拠していなければならない。

#### 12.3.1.1 署名の生成

1) [XMLDSIG]で定められている通り、SOAP ***ヘッダ***および必要な搬送内容ブジェクトに、***ds:SignatureMethod***、***ds:CanonicalizationMethod***、および ***ds:SignedInfo*** 要素で ***ds:SignedInfo*** 要素を作成する。
2) [XMLDSIG]で定められている通り、***ds:SignedInfo*** 上の ***ds:SignatureValue*** を正式化した後、***ds:SignedInfo*** で指定されたアルゴリズムに基づいて計算する。
3) [XMLDSIG]で定められている通り、***ds:SignedInfo***、***ds:KeyInfo*** (推奨)、および ***ds:SignatureValue*** 要素を含んだ ***ds:Signature*** 要素を構築する。
4) ***TraceHeaderList*** 要素に続いて、署名を付した SOAP ***ヘッダ***に、名前空間に適格な ***ds:Signature*** 要素を含める。

***ds:SignedInfo*** 要素は、***ds:CanonicalizationMethod*** (0 もしくは 1 つ)、***ds:SignatureMethod*** (0 もしくは 1 つ)、そして ***ds:Reference*** (1 つ以上)で構成されなければならない。

***ds:CanonicalizationMethod*** 要素は、[XMLDSIG]でオプションとして定めており、***ds:SignedInfo*** 要素のインスタンスに必ずしも含める必要はない。。その他の方法を指定する ***ds:Canonicalization*** 要素がない場合、署名されるデータに適用される既定の正規化方法は[XMLC14N]となる。これは、*ebXML メッセージ取扱サービス*でも既定の正規化方法となる。。

***ds:SignatureMethod*** は必須要素で、Algorithm 属性がなければならない。Algorythm 属性に推奨される値は次の通りである。

http://www.w3.org/2000/09/xmldsig#dsa-sha1

*ebXML メッセージ取扱サービス*に準拠したソフトウェアは、すべてこの推奨値をサポートしなければならない。

SOAP ***ヘッダ***文書の ***ds:Reference*** 要素には、***ds:Signature*** 要素を含む文書(SOAP ***ヘッダ***)に適用する署名を提供するため、URI 属性値""を含まなければならない。

ebXML 文書の ***ds:Reference*** 要素には、[XMLDSIG]に従い、値が「http://www.w3.org/2000/09/xmldsig#Object」の ***Type*** 属性を含めることができる。この属性は、情報提供のみを目的としたもので、省略可能である。ebXML MSH の実装では、どちらの場合も処理できるようにしなければならない。***ds:Reference*** 要素には、オプションの ***id*** 属性を含めることも可能である。。

SOAP ***ヘッダ***の ***ds:Reference*** 要素には、***ds:Signature*** という子要素が含まれていなければならない。***ds:Transforms*** 要素には 2 つの子要素、***ds:Transform*** が含まれていなければならない。最初の ***ds:Transform*** 要素には、以下の値を持った ***ds:Algorithm*** 属性がなければならない。

http://www.w3.org/2000/09/xmldsig#enveloped-signature

2 番目の ***ds:Transform*** 要素には、以下の値を持った ***ds:XPath*** という子要素が含まれていなければならない。

```
not(ancestor-or-self::eb:TraceHeaderList or
    ancestor-or-self::eb:Via)
```

最初の[XPath]ステートメントの結果、***ds:Signature*** 要素とその従属要素がすべて除外され、2 番目の[XPath]ステートメントで ***TraceHeaderList*** および ***Via*** 要素とその従属要素がすべて除外される。それは、これらの要素が変わる場合があるからである。

署名の必要な搬送内容オブジェクトは、その搬送内容オブジェクトに分解される ***URI*** 属性が含まれていなければならない ***ds:Reference*** 要素でそれぞれ表さなければならない。これには、搬送内容オブジェクト MIME ボディ内の `Content-Id` URI 搬送内容オブジェクトの MIME ボディ部の Content-Location 一致する URI、またはメッセージパッケージ外部の外部搬送内容オブジェクトに分解される URI のいずれかを使用することが可能である。。URI 属性の値は、対応する搬送内容オブジェクトの ***Manifest/Reference*** 要素の、xlink:href URI の値と一致するようにすることを強く推奨する。しかし、これは必須要件ではない。

デジタル署名が付された ebXML *SOAP メッセージ*の例を以下に示す。

```
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eb="http://www.ebxml.org/namespaces/messageHeader"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <SOAP-ENV:Header>
    <eb:MessageHeader eb:id="…" eb:version="1.0">
    ...
    </eb:MessageHeader>
    <eb:TraceHeaderList eb:id="…" eb:version="1.0">
      <eb:TraceHeader>
        ...
      </eb:TraceHeader>
    </eb:TraceHeaderList>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
        <ds:Reference URI="">
             <Transforms>
                <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
                   <XPath xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
                              not(ancestor-or-self::eb:TraceHeaderList or
                              ancestor-or-self::eb:Via)
                   </XPath>
                 </Transform>
             </Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
          <ds:DigestValue>...</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="cid://blahblahblah/">
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
          <ds:DigestValue>...</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>...</ds:SignatureValue>
       <ds:KeyInfo>...</ds:KeyInfo>
    </ds:Signature>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <eb:Manifest eb:id="Mani01" eb:version="1.0">
      <eb:Reference xlink:href="cid://blahblahblah"
        xlink:role="http://ebxml.org/gci/invoice">
        <eb:Schema eb:version="1.0" eb:location="http://ebxml.org/gci/busdocs/invoice.dtd"/>
      </eb:Reference>
    </eb:Manifest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 12.3.2 永続的署名付き受領書

デジタル署名が付された *ebXML メッセージ*の受領は、前項で説明している方法でそれ自体にもデジタル署名が付されている ***DeliveryReceipt*** 受領通知メッセージで通知できる。受領通知メッセージには、***Acknowledgement*** 要素の元のメッセージにある、***ds:Signature*** 要素内の ***ds: Reference*** 要素が含まれていなければならない。

### 12.3.3 非永続的認証

非永続的認証は、*ebXML メッセージ*の転送に使用される通信経路により提供される。この認証は、一方向的(セッション起動者から受信者へ)なものと、双方向なものが可能である。具体的な方法は、使用する通信プロトコルによって定められます。例えば、[RFC2246]や[IPSEC]のような安全なネットワークプロトコルを使用すれば、*ebXML メッセージ*の送信者は TCP/IP 環境で送信先を認証できる。

### 12.3.4 非永続的完全性

[RFC2246]や[IPSEC]のような安全なネットワークプロトコルの使用を設定し、ネットワーク接続を経て送信されるパケットの CRC の完全性を確認できる。

### 12.3.5 永続的な機密の保持

XML の暗号化は、厳選された XML 文書暗号化方法の仕様草案の作成に積極的に関わっている W3C と IETF の共同の取り組みである。この仕様は、来年度内に完成する予定である。ebXML トランスポート・ルーティングおよびパッケージングチームは、SOAP ***ヘッダ***を含む *ebXML メッセージ*内の要素に、永続的かつ選択性のある機密性を実現可能な唯一の方法として、この技術を識別している。

ebXML 搬送内容の機密性は MSH が処理する機能で提供される。しかし、本書では、ebXML 搬送内容にセキュリティを提供するのは MSH の役割の範囲外であると定めている。搬送内容の機密性は、XML 暗号化(使用可能な場合)、もしくは[S/MIME]、[S/MINEV3]、[PGS/MIME]など、関係当事者間で相互に合意されたその他の暗号化プロセスにより実現することが可能である。XML 暗号化は現在まだ使用できないので、ebXML 搬送内容には[S/MIME]による暗号化方法を使用することを推奨する。XML 暗号化が W3C の推薦を得た後は、XML 暗号化規格を既定の暗号化方法としなければならない。

### 12.3.6 非永続的な機密の保持

[RFC2246]や[IPSEC]などの安全なネットワークプロトコルを使用すると、2 つの ebXML MSH ノード間でメッセージが転送される際、一時的な機密性が提供される。

### 12.3.7 永続的許可

OASIS セキュリティサービス専門委員会(TC)は、NameAssertion や[SAML]に基づいた Entitlements を含む、セキュリティ上の証明を交換するための仕様の定義に積極的に携わっている。。採用が見込まれているこの仕様に基づいた技術を使用することで、仕様が利用できるようになった際に ebXML の認証を引き続き使用できる。ebXML は、この TC と正式なつながりがある。また、OASIS Security Services TC の積極的な参加メンバとなっている ebXML 会員組織や協力組織は、Sun、IBM、CommnerceOne、Cisco、および *ebXML メッセージ取扱サービス*に永続的な認証機能を実現するための要件を満たす仕様の確立に取り組んでいる他の組織など、多数ある。

### 12.3.8 非永続的許可

[RFC2246]や[IPSEC]などの安全なネットワークプロトコルは、セッションを確立する前に証明の相互認証を行えるように設定することができる。これにより ebXML MSH は、認証済みの *ebXML メッセージ*ソースとして送信元を見分けることが可能な接続先の証明を行うことができるようになる。。

### 12.3.9 信頼の置けるタイムスタンプ

本書の現段階では、信頼の置けるタイムスタンプ機能を提供するサービスが使用できるようになりつつある。これらのサービスが広く利用できるようになり、その使用法や表現法が規格によって定められれば、これらの規格、技術、そしてサービスが評価され、この機能の実現のために使用することが検討されるようになるであろう。

### 12.3.10 サポートされるセキュリティサービス

ebXML メッセージ取扱サービス仕様の全体的なアーキテクチャは、電子ビジネスで必要とされるすべてのセキュリティサービスをサポートするよう意図されている。以下の表は、メッセージ取扱サービスハンドラのセキュリティサービスをセキュリティプロファイル一式と組み合わせたものである。これらのプロファイルまたはプロファイルの組み合わせにより、ebXML ユーザコミュニティの特定のセキュリティポリシーがサポートされる。XML セキュリティ仕様が未完状態であることから、本書のこのバージョンではプロファイル 0 と 1 のみが必要となる。。これは、別のセキュリティ機能を利用して ebXML の情報交換を保護することを妨げるものではないが、プロファイル 0 および 1 以外のプロファイルを使用する場合、当事者間の相互運用性は保証されない。

| ベースライン MSH 内の有無 | | 永続的デジタル署名 | 非永続的認証 | 永続的署名付き受領 | 非永続的完全性 | 永続的機密性 | 非永続的機密性 | 永続的許可 | 非永続的許可 | 信頼の置けるタイムスタンプ | プロファイルの説明 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | Profile 0 | | | | | | | | | | データに適用されるセキュリティサービスはない。 |
| ✓ | Profile 1 | ✓ | | | | | | | | | *送信側 MSH* は XML/DSIG 構造をメッセージに適用する。 |
| | Profile 2 | | ✓ | | | | | | ✓ | | *送信側 MSH* は認証を行い、*受信側 MSH* は通信チャンネルの信用に基づいて送信者を許可する。 |
| | Profile 3 | | ✓ | | | | ✓ | | | | *送信側 MSH* は認証を行い、送信側、受信側双方の MSH は安全なデータ送信経路の交渉を行う。 |
| | Profile 4 | | ✓ | | ✓ | | | | | | *送信側 MSH* は認証を行い、*受信側 MSH* は通信プロトコルを使用して完全性の確認を行う。 |
| | Profile 5 | | ✓ | | | | | | | | *送信側 MSH* は通信経路の認証のみ行う。(例えば、TCP/IP 上の SSL 3.0)。 |
| | Profile 6 | ✓ | | | | | ✓ | | | | *送信側 MSH* はメッセージに XML/DSIG 構造を適用し、安全な通信チャンネルで引き渡す。 |
| | Profile 7 | ✓ | | ✓ | | | | | | | *送信側 MSH* はメッセージに XML/DSIG 構造を適用し、*受信側 MSH* は署名付き受領書を返信する。 |
| | Profile 8 | ✓ | | ✓ | | | ✓ | | | | プロファイル 6 および 7 の組み合わせ。 |
| | Profile 9 | ✓ | | | | | | | | ✓ | プロファイル 5、および信頼の置けるタイムスタンプを適用する。 |
| | Profile 10 | ✓ | | ✓ | | | | | | ✓ | プロファイル 9、および *受領側 MSH* が署名付き受領書を返信する。 |

| ベースライン MSH 内の有無 | | 永続的デジタル署名 | 非永続的認証 | 永続的署名付き受領 | 非永続的完全性 | 永続的機密性 | 非永続的機密性 | 永続的許可 | 非永続的許可 | 信頼の置けるタイムスタンプ | プロファイルの説明 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Profile 11 | ✓ | | | | | ✓ | | | ✓ | プロファイル 6、および*受領側 MSH* が信頼の置けるタイムスタンプを適用する。 |
| | Profile 12 | ✓ | | ✓ | | | ✓ | | | ✓ | プロファイル 8、および*受領側 MSH* が信頼の置けるタイムスタンプを適用する。 |
| | Profile 13 | ✓ | | | | ✓ | | | | | *送信側 MSH* はメッセージに XML/DSIG 構造を適用し、機密構造を適用(XML 暗号化)する。 |
| | Profile 14 | ✓ | | ✓ | | ✓ | | | | | プロファイル 13、および署名付き受領書。 |
| | Profile 15 | ✓ | | ✓ | | | | | | ✓ | *送信側 MSH* はメッセージに XML/DSIG 構造を適用し、信頼の置けるタイムスタンプがメッセージに追加され、*受信側 MSH* は署名付き受領書を返信する。 |
| | Profile 16 | ✓ | | | | ✓ | | | | ✓ | プロファイル 13、および信頼の置けるタイムスタンプを適用する。 |
| | Profile 17 | ✓ | | ✓ | | ✓ | | | | ✓ | プロファイル 14、および信頼の置けるタイムスタンプを適用する。 |
| | Profile 18 | ✓ | | | | | | ✓ | | | *送信側 MSH* はメッセージに XML/DSIG 構造を適用し、許可証明を転送する(SAML)。 |
| | Profile 19 | ✓ | | ✓ | | | | ✓ | | | プロファイル 18、および*受領側 MSH* が署名付き受領書を返信する。 |
| | Profile 20 | ✓ | | ✓ | | | | ✓ | | ✓ | プロファイル 19、および*送信側 MSH* のメッセージに信頼の置けるタイムスタンプを適用する。 |
| | Profile 21 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | プロファイル 19、および*送信側 MSH* が機密構造(XML 暗号化)を適用する。 |
| | Profile 22 | | | | | ✓ | | | | | *送信側 MSH* は機密構造(XML 暗号化)内にメッセージをカプセル化する。 |

# 13 関連図書

## 13.1 規範的関連図書


[RFC2119] Key Words for use in RFCs to Indicate Requirement Levels, Internet Engineering Task Force RFC 2119, 1997 年 3 月

[HTTP] IETF RFC 2068 - Hypertext Transfer Protocol -- HTTP/1.1, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, 1997 年 1 月

[RFC822] Standard for the Format of ARPA Internet text messages. D. Crocker. 1982 年 8 月.

[RFC2045] IETF RFC 2045. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, N Freed & N Borenstein, 1996 年 11 月刊行

[RFC2046] Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed, N. Borenstein. 1996 年 11 月.

[RFC2246] RFC 2246 - Dierks, T. and C. Allen, "The TLS Protocol", 1999 年 1 月.

[RFC2387] The MIME Multipart/Related Content-type. E. Levinson. 1998 年 8 月.

[RFC2392] IETF RFC 2392. Content-ID and Message-ID Uniform Resource Locators. E. Levinson, 1998 年 8 月刊行

[RFC2396] IETF RFC 2396. Uniform Resource Identifiers (URI): Generic Syntax. T Berners-Lee, 1998 年 8 月刊行

[RFC2487] SMTP Service Extension for Secure SMTP over TLS. P. Hoffman. 1999 年 1 月.

[RFC2554] SMTP Service Extension for Authentication. J. Myers. 1999 年 3 月.

[RFC2616] RFC 2616 - Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol, HTTP/1.1", , 1999 年 6 月.

[RFC2617] RFC2617 - Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., Sink, E. and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", 1999 年 6 月.

[RFC2817] RFC 2817 - Khare, R. and S. Lawrence, "Upgrading to TLS Within HTTP/1.1", 2000 年 5 月.

[RFC2818] RFC 2818 - Rescorla, E., "HTTP Over TLS", May 2000 [SOAP] Simple Object Access Protocol

[SMTP] IETF RFC 822, Simple Mail Transfer Protocol, D Crocker, 1982 年 8 月

[SOAP] W3C-Draft-Simple Object Access Protocol (SOAP) v1.1, Don Box, DevelopMentor; David Ehnebuske, IBM; Gopal Kakivaya, Andrew Layman, Henrik Frystyk Nielsen, Satish Thatte, Microsoft; Noah Mendelsohn, Lotus Development Corp.; Dave Winer, UserLand Software, Inc.; 2000 年 5 月 W3C メモ, http://www.w3.org/TR/SOAP

[SOAPATTACH] SOAP Messages with Attachments, John J. Barton, Hewlett Packard Labs; Satish Thatte and Henrik Frystyk Nielsen, Microsoft, 2000 年 10 月 9 日刊行 http://www.w3.org/TR/SOAP-attachments

[SSL3] A. Frier, P. Karlton, and P. Kocher, "The SSL 3.0 Protocol", Netscape Communications Corp., 1996 年 11 月 18 日.

[UTF-8] UTF-8 is an encoding that conforms to ISO/IEC 10646. See [XML] for usage conventions.

[XLINK] W3C XML Linking Candidate Recommendation, http://www.w3.org/TR/xlink/

[XML] W3C Recommendation: Extensible Markup Language (XML) 1.0 (Second Edition), 2000 年 10 月, http://www.w3.org/TR/2000/REC-xml-20001006

[XML Namespace] W3C Recommendation for Namespaces in XML, World Wide Web Consortium, 1999 年 1 月 14 日, http://www.w3.org/TR/REC-xml-names

[XMLDSIG] Joint W3C/IETF XML-Signature Syntax and Processing specification, http://www.w3.org/TR/2000/CR-xmldsig-core-20001031/

[XMLMedia] IETF RFC 3023, XML Media Types. M. Murata, S. St.Laurent, 2001 年 1 月

## 13.2 非規範的関連図書

[ebCPP] ebXML コラボレーションプロトコルプロファイルおよびコラボレーションプロトコル合意書仕様, Version 1.0, 2001 年 5 月 11 日刊行

[ebBPSS] ebXML ビジネスプロセス仕様スキーマ, version 1.0, 2001 年 4 月 27 日刊行.

[ebTA] ebXML テクニカルアーキテクチャ, version 1.04 2001 年 2 月 16 日、金曜日刊行

[ebTASEC] ebXML テクニカルアーキテクチャリスク評価技術報告 t, version 0.36 2001 年 4 月 20 日刊行

[ebRS] ebXML レジストリサービス仕様, version 0.84

[ebMSREQ] ebXML トランスポート・ルーティングおよびパッケージング: 概要と要件, Version 0.96, 2000 年 5 月 25 日刊行

[ebGLOSS] ebXML 用語集, http://www.ebxml.org, 2001 年 5 月 11 日刊行.

[IPSEC] IETF RFC2402 IP Authentication Header. S. Kent, R. Atkinson. 1998 年 11 月. RFC2406 IP Encapsulating Security Payload (ESP). S. Kent, R. Atkinson. 1998 年 11 月.

[PGP/MIME] IETF RFC2015, "MIME Security with Pretty Good Privacy (PGP)", M. Elkins. 1996 年 10 月.

[SAML] Security Assertion Markup Language, http://www.oasis-open.org/committees/security/docs/draft-sstc-use-strawman-03.html

[S/MIME] IETF RFC2311, “S/MIME Version 2 Message Specification”, S. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, L. Repka. 1998 年 3 月.

[S/MIMECH] IETF RFC 2312, “S/MIME Version 2 Certificate Handling”, S. Dusse, P. Hoffman, B. Ramsdell, J. Weinstein. 1998 年 3 月.

[S/MIMEV3] IETF RFC 2633 S/MIME Version 3 Message Specification. B. Ramsdell, Ed.. 1999 年 6 月.

[TLS] RFC2246, T. Dierks, C. Allen. 1999 年 1 月.

[XMLSchema] W3C XML Schema Candidate Recommendation,
http://www.w3.org/TR/xmlschema-0/
http://www.w3.org/TR/xmlschema-1/
http://www.w3.org/TR/xmlschema-2/

[XMTP] XMTP - Extensible Mail Transport Protocol
http://www.openhealth.org/documents/xmtp.htm

# 14 連絡先情報

**チームリーダー**


| | |
|---|---|
| 名前 | Rik Drummond |
| 所属 | Drummond Group, Inc. |
| 住所 | 5008 Bentwood Ct.<br>Fort Worth, Texas 76132<br>USA |
| 電話 | +1 (817) 294-7339 |
| 電子メール | rik@drummondgroup.com |


**チーム副リーダー**


| | |
|---|---|
| 名前 | Christopher Ferris |
| 所属 | Sun Microsystems |
| 住所 | One Network Drive<br>Burlington, MA 01803-0903<br>USA |
| 電話 | +1 (781) 442-3063 |
| 電子メール | chris.ferris@sun.com |


**チーム編集者**


| | |
|---|---|
| 名前 | David Burdett |
| 所属 | Commerce One |
| 住所 | 4400 Rosewood Drive<br>Pleasanton, CA 94588<br>USA |
| 電話 | +1 (925) 520-4422 |
| 電子メール | david.burdett@commerceone.com |


**著者**


| | |
|---|---|
| 名前 | Dick Brooks |
| 所属 | Group 8760 |
| 住所 | 110 12th Street North, Suite F103<br>Birmingham, Alabama 35203 |
| 電話 | +1 (205) 250-8053 |
| 電子メール | dick@8760.com |

| | |
|---|---|
| 名前 | David Burdett |
| 所属 | Commerce One |
| 住所 | 4400 Rosewood Drive<br>Pleasanton, CA 94588<br>USA |
| 電話 | +1 (925) 520-4422 |
| 電子メール | david.burdett@commerceone.com |

| | |
|---|---|
| 名前 | Christopher Ferris |
| 所属 | Sun Microsystems |
| 住所 | One Network Drive<br>Burlington, MA 01803-0903<br>USA |

電話 +1 (781) 442-3063
電子メール chris.ferris@east.sun.com

名前 John Ibbotson
所属 IBM UK Ltd
住所 Hursley Park
Winchester SO21 2JN
United Kingdom
電話 +44 (1962) 815188
電子メール john_ibbotson@uk.ibm.com

名前 Masayoshi Shimamura
所属 Fujitsu Limited
住所 Shinyokohama Nikko Bldg., 15-16, Shinyokohama 2-chome
Kohoku-ku, Yokohama 222-0033, Japan
電話 +81-45-476-4590
電子メール shima@rp.open.cs.fujitsu.co.jp

**文書編集チーム**

名前 Ralph Berwanger
所属 bTrade.com
住所 2324 Gateway Drive
Irving, TX 75063
USA
電話 +1 (972) 580-3970
電子メール rberwanger@btrade.com

名前 Colleen Evans
所属 Progress/Sonic Software
住所 14 Oak Park
Bedford, MA 01730
USA
電話 +1 (720) 480-3919
電子メール cevans@progress.com

名前 Ian Jones
所属 British Telecommunications
住所 Enterprise House, 84-85 Adam Street
Cardiff, CF24 2XF
United Kingdom
電話 +44 29 2072 4063
電子メール ian.c.jones@bt.com

名前 Martha Warfelt
所属 DaimlerChrysler Corporation
住所 800 Chrysler Drive
Auburn Hills, MI
USA
電話 +1 (248) 944-5481
電子メール maw2@daimlerchrysler.com

| | |
|---|---|
| 名前 | David Fischer |
| 所属 | Drummond Group, Inc |
| 住所 | 5008 Bentwood Ct<br>Fort Worth, TX  76132 |
| 電話 | +1 (817-294-7339 |
| 電子メール | david@drummondgroup.com |

# 付録 A ebXML SOAP 拡張要素のスキーマ

ebXML SOAP 拡張要素のスキーマは、XML スキーマ仕様[XMLSchema]の勧告候補草案を用いて定められている。ebXML がメッセージ形式に SOAP 1.1 を採用し、初期の XML スキーマ仕様草案には SOAP 1.1 名前空間 URI によって決定された SOAP 1.1 スキーマが記述されていることから、ebXML TRP チームは、W3C XML スキーマ勧告候補仕様[XMLSchema]に準拠したスキーマの語彙を用いて定めた SOAP 1.1 エンベロープスキーマのバージョンを作成した。

また、XLINK 属性の語彙と XML xml:lang 属性にもスキーマを作成する必要があった。

最後に、編集ツールによってはスキーマをインポートする際、ローカル要素が正しく決定されないので、この付録で定義されている ebXML SOAP 拡張スキーマで W3C 署名コアスキーマも提供され、参照されている。

これらの代替スキーマは、以下の URL で入手可能である。

XML 署名コア – http://ebxml.org/project_teams/transport/xmldsig-core-schema.xsd

Xlink - http://ebxml.org/project_teams/transport/xlink.xsd

xml:lang - http://ebxml.org/project_teams/transport/xml_lang.xsd

SOAP1.1 - http://ebxml.org/project_teams/transport/envelope.xsd

注意: 仕様とこのスキーマとの間に矛盾が生じている場合は、仕様が優先される。

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.ebxml.org/namespaces/messageHeader"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:tns="http://www.ebxml.org/namespaces/messageHeader" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="http://www.w3.org/2000/10/XMLSchema" version="1.0">
  <import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.ebxml.org/project_teams/transport/xmldsig-core-schema.xsd"/>
  <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="http://www.ebxml.org/project_teams/transport/xlink.xsd"/>
  <import namespace="http://schemas.xmlsoap.org/soap/envelope/"
schemaLocation="http://www.ebxml.org/project_teams/transport/envelope.xsd"/>
  <import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.ebxml.org/project_teams/transport/xml_lang.xsd"/>
  <!-- MANIFEST -->
  <element name="Manifest">
    <complexType>
      <sequence>
        <element ref="tns:Reference" maxOccurs="unbounded"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute ref="tns:id"/>
      <attribute ref="tns:version"/>
      <anyAttribute namespace="http://www.w3.org/2000/10/XMLSchema-instance"
          processContents="lax"/>
    </complexType>
  </element>
  <element name="Reference">
    <complexType>
      <sequence>
        <element ref="tns:Schema" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="tns:Description" minOccurs="0" maxOccurs="unbounded"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute ref="tns:id"/>
      <attribute ref="xlink:type" use="fixed" value="simple"/>
      <attribute ref="xlink:href" use="required"/>
```

```
    <attribute ref="xlink:role"/>
  </complexType>
</element>
<element name="Schema">
  <complexType>
    <attribute name="location" type="uriReference" use="required"/>
    <attribute name="version" type="tns:non-empty-string"/>
  </complexType>
</element>
<!-- MESSAGEHEADER -->
<element name="MessageHeader">
  <complexType>
    <sequence>
      <element ref="tns:From"/>
      <element ref="tns:To"/>
      <element ref="tns:CPAId"/>
      <element ref="tns:ConversationId"/>
      <element ref="tns:Service"/>
      <element ref="tns:Action"/>
      <element ref="tns:MessageData"/>
      <element ref="tns:QualityOfServiceInfo" minOccurs="0"/>
      <element ref="tns:Description" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="tns:SequenceNumber" minOccurs="0"/>
    </sequence>
    <attribute ref="tns:id"/>
    <attribute ref="tns:version"/>
    <attribute ref="soap:mustUnderstand"/>
    <anyAttribute namespace="http://www.w3.org/2000/10/XMLSchema-instance"
      processContents="lax"/>
  </complexType>
</element>
<element name="CPAId" type="tns:non-empty-string"/>
<element name="ConversationId" type="tns:non-empty-string"/>
<element name="Service">
  <complexType>
    <simpleContent>
      <extension base="tns:non-empty-string">
        <attribute name="type" type="tns:non-empty-string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="Action" type="tns:non-empty-string"/>
<element name="MessageData">
  <complexType>
    <sequence>
      <element ref="tns:MessageId"/>
      <element ref="tns:Timestamp"/>
      <element ref="tns:RefToMessageId" minOccurs="0"/>
      <element ref="tns:TimeToLive" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
<element name="MessageId" type="tns:non-empty-string"/>
<element name="TimeToLive" type="timeInstant"/>
<element name="QualityOfServiceInfo">
  <complexType>
    <attribute name="deliverySemantics" type="tns:deliverySemantics.type" use="default"
      value="BestEffort"/>
    <attribute name="messageOrderSemantics" type="tns:messageOrderSemantics.type"
      use="default" value="NotGuaranteed"/>
    <attribute name="deliveryReceiptRequested" type="tns:signedUnsigned.type"
      use="default" value="None"/>
  </complexType>
</element>
<!-- TRACE HEADER LIST -->
<element name="TraceHeaderList">
  <complexType>
    <sequence>
      <element ref="tns:TraceHeader" maxOccurs="unbounded"/>
    </sequence>
```

```
    <attribute ref="tns:id"/>
    <attribute ref="tns:version"/>
    <attribute ref="soap:mustUnderstand" use="required"/>
    <attribute ref="soap:actor" use="required"/>
    <anyAttribute namespace="http://www.w3.org/2000/10/XMLSchema-instance"
      processContents="lax"/>
  </complexType>
</element>
<element name="TraceHeader">
  <complexType>
    <sequence>
      <element ref="tns:Sender"/>
      <element ref="tns:Receiver"/>
      <element ref="tns:Timestamp"/>
      <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute ref="tns:id"/>
  </complexType>
</element>
<element name="Sender" type="tns:senderReceiver.type"/>
<element name="Receiver" type="tns:senderReceiver.type"/>
<element name="SequenceNumber" type="positiveInteger"/>
<!-- DELIVERY RECEIPT -->
<element name="DeliveryReceipt">
  <complexType>
    <sequence>
      <element ref="tns:Timestamp"/>
      <element ref="ds:Reference" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute ref="tns:id"/>
    <attribute ref="tns:version"/>
    <anyAttribute namespace="http://www.w3.org/2000/10/XMLSchema-instance"
        processContents="lax"/>
    <!-- <attribute name="signed" type="boolean"/> -->
  </complexType>
</element>
<!-- ACKNOWLEDGEMENT -->
<element name="Acknowledgment">
  <complexType>
    <sequence>
      <element ref="tns:Timestamp"/>
      <element ref="tns:From" minOccurs="0"/>
      <element ref="ds:Reference" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute ref="tns:id"/>
    <attribute ref="tns:version"/>
    <attribute ref="soap:mustUnderstand" use="required"/>
    <attribute ref="soap:actor" use="required"/>
    <anyAttribute namespace="http://www.w3.org/2000/10/XMLSchema-instance"
      processContents="lax"/>
  </complexType>
</element>
<!-- ERROR LIST -->
<element name="ErrorList">
  <complexType>
    <sequence>
      <element ref="tns:Error" maxOccurs="unbounded"/>
    </sequence>
    <attribute ref="tns:id"/>
    <attribute ref="tns:version"/>
    <attribute ref="soap:mustUnderstand" use="required"/>
    <attribute name="highestSeverity" type="tns:severity.type"
      use="default" value="Warning"/>
    <anyAttribute namespace="http://www.w3.org/2000/10/XMLSchema-instance"
       processContents="lax"/>
  </complexType>
</element>
<element name="Error">
  <complexType>
    <attribute ref="tns:id"/>
    <attribute name="codeContext" type="uriReference" use="required"/>
```

```
    <attribute name="errorCode" type="tns:non-empty-string" use="required"/>
    <attribute name="severity" type="tns:severity.type" use="default" value="Warning"/>
    <attribute name="location" type="tns:non-empty-string"/>
    <attribute ref="xml:lang"/>
  </complexType>
</element>
<!-- STATUS RESPONSE -->
<element name="StatusResponse">
  <complexType>
    <sequence>
      <element ref="tns:RefToMessageId"/>
      <element ref="tns:Timestamp" minOccurs="0"/>
    </sequence>
    <attribute ref="tns:id"/>
    <attribute ref="tns:version"/>
    <attribute name="messageStatus" type="tns:messageStatus.type"/>
    <anyAttribute namespace="http://www.w3.org/2000/10/XMLSchema-instance"
      processContents="lax"/>
  </complexType>
</element>
<!-- STATUS REQUEST -->
<element name="StatusRequest">
  <complexType>
    <sequence>
      <element ref="tns:RefToMessageId"/>
    </sequence>
    <attribute ref="tns:id"/>
    <attribute ref="tns:version"/>
    <anyAttribute namespace="http://www.w3.org/2000/10/XMLSchema-instance"
      processContents="lax"/>
  </complexType>
</element>
<!-- VIA  -->
<element name="Via">
  <complexType>
    <sequence>
      <element ref="tns:CPAId" minOccurs="0"/>
      <element ref="tns:Service" minOccurs="0"/>
      <element ref="tns:Action" minOccurs="0"/>
    </sequence>
    <attribute ref="tns:id"/>
    <attribute ref="tns:version"/>
    <attribute ref="soap:mustUnderstand" use="required"/>
    <attribute ref="soap:actor" use="required"/>
    <attribute name="syncReply" type="boolean"/>
    <attribute name="deliveryReceiptRequested" type="tns:signedUnsigned.type"
      use="default" value="None"/>
    <attribute name="reliableMessagingMethod" type="tns:rmm.type"/>
    <attribute name="ackRequested" type="boolean"/>
    <anyAttribute namespace="http://www.w3.org/2000/10/XMLSchema-instance"
      processContents="lax"/>
  </complexType>
</element>
<!-- COMMON TYPES -->
<complexType name="senderReceiver.type">
  <sequence>
    <element ref="tns:PartyId" maxOccurs="unbounded"/>
    <element name="Location" type="uriReference"/>
  </sequence>
</complexType>
<simpleType name="messageStatus.type">
  <restriction base="NMTOKEN">
    <enumeration value="UnAuthorized"/>
    <enumeration value="NotRecognized"/>
    <enumeration value="Received"/>
    <enumeration value="Processed"/>
    <enumeration value="Forwarded"/>
  </restriction>
</simpleType>
<simpleType name="type.type">
  <restriction base="NMTOKEN">
```

```
    <enumeration value="DeliveryReceipt"/>
    <enumeration value="IntermediateAck"/>
  </restriction>
</simpleType>
<simpleType name="messageOrderSemantics.type">
  <restriction base="NMTOKEN">
    <enumeration value="Guaranteed"/>
    <enumeration value="NotGuaranteed"/>
  </restriction>
</simpleType>
<simpleType name="deliverySemantics.type">
  <restriction base="NMTOKEN">
    <enumeration value="OnceAndOnlyOnce"/>
    <enumeration value="BestEffort"/>
  </restriction>
</simpleType>
<simpleType name="non-empty-string">
  <restriction base="string">
    <minLength value="1"/>
  </restriction>
</simpleType>
<simpleType name="rmm.type">
  <restriction base="NMTOKEN">
    <enumeration value="ebXML"/>
    <enumeration value="Transport"/>
  </restriction>
</simpleType>
<simpleType name="signedUnsigned.type">
  <restriction base="NMTOKEN">
    <enumeration value="Signed"/>
    <enumeration value="Unsigned"/>
    <enumeration value="None"/>
  </restriction>
</simpleType>
<simpleType name="severity.type">
  <restriction base="NMTOKEN">
    <enumeration value="Warning"/>
    <enumeration value="Error"/>
  </restriction>
</simpleType>
<!-- COMMON ATTRIBUTES and ELEMENTS -->
<attribute name="id" type="ID" form="unqualified"/>
<attribute name="version" type="tns:non-empty-string" use="fixed" value="1.0"/>
<element name="PartyId">
  <complexType>
    <simpleContent>
      <extension base="tns:non-empty-string">
        <attribute name="type" type="tns:non-empty-string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="To">
  <complexType>
    <sequence>
      <element ref="tns:PartyId" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
<element name="From">
  <complexType>
    <sequence>
      <element ref="tns:PartyId" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
<element name="Description">
  <complexType>
    <simpleContent>
      <extension base="tns:non-empty-string">
        <attribute ref="xml:lang"/>
```

```
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="RefToMessageId" type="tns:non-empty-string"/>
  <element name="Timestamp" type="timeInstant"/>
</schema>
```

# 付録 B 通信プロトコルのバインド

## B.1 はじめに

ebXML トランスポート・ルーティングおよびパッケージングチームの目的の 1 つは、さまざまなネットワークやアプリケーションレベルの通信プロトコルで使用可能なメッセージ取扱サービスを設計することにある。これらのプロトコルは、ebXML メッセージの「搬送媒体」の役割を果たすもので、二者間の完全な ebXML メッセージ交換を実施するために必要な基本的サービスを提供する。HTTP、FTP、Java Message Service (JMS)、および SMTP は、アプリケーションレベルの通信プロトコルの例であり、TCP および SNA/LU6.2 はネットワーク転送プロトコルの例である。通信プロトコルがサポートするデータ内容、その処理動作、そしてエラーの取扱と報告の方法はさまざまである。例えば、HTTP ではバイナリデータをそのまま送信するのが通例である。が、SMTP の場合は、通常、バイナリデータが 7 ビットの形式に「エンコード」される。HTTP は同期または非同期のメッセージ交換を両方とも実行できるが、SMTP では非同期でメッセージ交換が行われるようである。本項では、特定の通信プロトコルでこの抽象的な ebXML メッセージ処理サービスを実装するのに必要な技術詳細について説明する。

本項では、以下の通信プロトコルで *ebXML メッセージ取扱サービス*のメッセージを搬送するための通信プロトコルのバインドおよび技術詳細を定める。

- HTTP (Hypertext Transfer Protocol)の*非同期*および*同期*転送
- SMTP (Simple Mail Transfer Protocol)の*非同期*転送

## B.2 HTTP

### B.2.1 HTTP プロトコルの最低レベル

使用しなければならないプロトコルの最低レベルは、Hypertext Transfer Protocol Version 1.1 [HTTP] (http://www.ietf.org/rfc2616.txt)である。

### B.2.2 HTTP による ebXML サービスメッセージの送信

HTTP の要求方法はいくつかあるが、本書では、HTTP で *ebXML メッセージ取扱サービス*のメッセージを送信するための、HTTP POST 要求の使用のみ定義する。ebXML MSH (例えば ebxmlhandler)の ID は、HTTP POST 要求に含めることができる。

```
POST /ebxmlhandler HTTP/1.1
```

HTTP での送信前に、ebXML メッセージは ebXML メッセージ取扱サービス仕様の7項および**8**項に従ってフォーマットしなければならない。また、メッセージは、RFC 2616 [HTTP]仕様の 19.4 項で定められている、HTTP 固有の標準的な MIME 形式の要件に従わなければならない(http://www.ietf.org/rfc2616.txt 参照)。

HTTP プロトコルは、元々、8 ビットとバイナリデータをサポートしている。従って、HTTP での送信の前に ebXML サービスメッセージの 8 ビット部およびバイナリデータ部の転送符号化はオプションとなる。しかし、それらの部分(例えば base64 符号化スキームの使用)のコンテンツ転送符号化は、本書では除外される。

ebXML サービスメッセージを含む HTTP メッセージの生成規則は以下の通りである。

- ebXML サービスメッセージエンベロープの **`Content-Type: Multipart/Related`** MIME ヘッダと関連パラメータは、HTTP ヘッダに含まなければならない。
- また、ebXML メッセージエンベロープを構成する、その他すべての MIME ヘッダも HTTP ヘッダに含まなければならない。
- さらに、HTTP ヘッダには、必須の SOAPAction HTTP ヘッダフィールドも含まなければならず、その値は「ebXML」とすることができる。

  `SOAPAction`: **”ebXML”**

- Content-Transfer-Encoding など、意味情報が MIME 仕様で定められているその他のヘッダを HTTP ヘッダとして含めることはできない。特に、「MIME-Version: 1.0」ヘッダは HTTP ヘッダとして示してはならない。しかし、HTTP 1.1 で定められている HTTP 固有の MIME のようなヘッダは、HTTP 仕様で定められている意味情報とともに使用できる。
- MIME 境界文字列を含む、ebXML メッセージエンベロープに従った ebXML サービスメッセージ部は、HTTP エンティティーのボディとなる。これには、SOAP エンベロープおよび構成要素となる ebXML 部や、追跡 MIME 境界文字列を含む、添付事項を含む。

以下は、HTTP POST 化された ebXML サービスメッセージのインスタンスの例である。。

```
POST /servlet/ebXMLhandler HTTP/1.1
Host: www.example2.com
SOAPAction: "ebXML"
Content-type: multipart/related; boundary="BoundarY"; type="text/xml";
        start=" <ebxhmheader111@example.com>"

--BoundarY
Content-ID: <ebxhmheader111@example.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns:eb='http://www.ebxml.org/namespaces/messageHeader'>
<SOAP-ENV:Header>
  <eb:MessageHeader SOAP-ENV:mustUnderstand="1" eb:version="1.0">
    <eb:From>
      <eb:PartyId>urn:duns:123456789</eb:PartyId>
    </eb:From>
    <eb:To>
      <eb:PartyId>urn:duns:912345678</eb:PartyId>
    </eb:To>
    <eb:CPAId>20001209-133003-28572</eb:CPAId>
    <eb:ConversationId>20001209-133003-28572</eb:ConversationId>
    <eb:Service>urn:services:SupplierOrderProcessing</eb:Service>
    <eb:Action>NewOrder</eb:Action>
    <eb:MessageData>
      <eb:MessageId>20001209-133003-28572@example.com</eb:MessageId>
      <eb:Timestamp>2001-02-15T11:12:12Z</Timestamp>
    </eb:MessageData>
    <eb:QualityOfServiceInfo eb:deliverySemantics="BestEffort"/>
  </eb:MessageHeader>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <eb:Manifest SOAP-ENV:mustUnderstand="1" eb:version="1.0">
    <eb:Reference xlink:href="cid:ebxmlpayload111@example.com"
         xlink:role="XLinkRole"
         xlink:type="simple">
        <eb:Description xml:lang="en-us">Purchase Order 1</eb:Description>
    </eb:Reference>
  </eb:Manifest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--BoundarY
Content-ID: <ebxmlpayload111@example.com>
Content-Type: text/xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<purchase_order>
  <po_number>1</po_number>
  <part_number>123</part_number>
  <price currency="USD">500.00</price>
</purchase_order>

--BoundarY--
```

### B.2.3 HTTP 応答コード

HTTP レベルの応答コードの返信には、通常、[RFC2616]で定められた HTTP による通信の意味情報に従わなければならない。受領側の HTTP エンティティが HTTP Post メッセージの受領に成功したら、2xx コードが返されなければならないが、以下の SOAP エラー状態の例外に注意すること。同様に、3xx、4xx、5xx の範囲の他の HTTP コードが、対応する状態の際に返される場合もある。しかし、ebXML サービスの処理の際に発生したエラー状態は、ebXML メッセージ取扱サービス仕様で定めるエラー機構を使用して報告しなければならない(11項参照)。

### B.2.4 SOAP エラー状態と同期交換

SOAP 1.1 仕様では次のように述べられている。

*「要求の処理中に SOAP エラーが発生した場合、SOAP HTTP サーバは HTTP 500 "内部サーバエラー" 応答を発行し、SOAP 処理エラーを示す SOAP Fault 要素が含まれたその応答内に SOAP メッセージを含めなければならない。」*

しかし、SOAP 1.1 仕様の範囲は、HTTP でのメッセージ同期交換モードに限られているのに対し、ebXML メッセージ取扱サービス仕様では HTTP での同期および非同期のメッセージ交換モードが定められます。従って、メッセージ同期交換モードは SOAP 1.1 仕様に従い、SOAP 処理のエラーを示す SOAP Fault 要素が含まれた SOAP メッセージは、「HTTP 500 内部サーバエラー」の応答コードとともに HTTP 応答内で返されなければならない。メッセージ非同期交換モードが使用されている場合、メッセージの受信に成功したときは 2xx の範囲の HTTP 応答コードを返し、エラー状態(SOAP エラーを含む)は別の HTTP Post で返さなければならない。

### B.2.5 同期と非同期の比較

**Via** 要素の **syncReply** パラメータが「true」に設定されている場合、上で説明している適切な HTTP 応答コードとともに、受信した要求と同じ HTTP 接続で応答メッセージを返さなければならない。
**syncReply** パラメータが「false」に設定されている場合、応答メッセージは受信した要求と同じ HTTP 接続で返されず、独立した HTTP Post 要求が使用される。HTTP Post に対しては、B.2.3 項で定められた応答コードと空の HTTP ボディが付された HTTP 応答が返されなければならない。

### B.2.6 アクセス制御

実装者は、アクセス制御の仕組みを用いて、権限のないアクセスからebXMLメッセージ取扱サービスハンドラを保護できる。「HTTP Authentication: Basic and Digest Access Authentication」[RFC2617]で説明しているHTTPアクセス認証プロセスは、権限のないアクセスからebXMLメッセージ取扱サービスハンドラを保護することが可能な、アクセス制御の仕組みが定めている。

実装者は、[RFC2617]で定めたアクセス制御スキームをすべてサポートできるが、アクセス制御を使用する際は、2項で説明する通り、基本的な認証の仕組みを必ずサポートしなければならない。

また、アクセス制御の基本的認証を使用する実装者は、本書の「機密性と通信プロトコルレベルのセキュリティ」の項で定められている通信プロトコルレベルのセキュリティも使用すべきである。

### B.2.7 機密性と通信プロトコルレベルのセキュリティ

ebXML メッセージ取扱サービスハンドラは、トランスポート層での暗号化を使用し、ebXML メッセージおよび HTTP トランスポートヘッダの機密性を保護できる。IETF トランスポート層セキュリティ仕様[RFC2246]では、ebXML メッセージ取扱サービスハンドラで使用できる具体的な技術詳細と使用可能なオプションの一覧を示す。ebXML メッセージ取扱サービスハンドラは、[RFC2246]の付録 E で定める通り、SSL [SSL3]との下位互換で動作できなければならない。

ebXML メッセージ取扱サービスハンドラは、[RFC2246]で定める使用可能な暗号化アルゴリズムと暗号化キーを使用できる。ebXML メッセージ取扱サービスハンドラは、少なくとも[SSL3]との下位互換性に必要な暗号化キーサイズとアルゴリズムをサポートしなければならない。

40 ビットの暗号化キー/アルゴリズムを使用することもできるが、より強力な暗号化キー/アルゴリズムを使用することを推奨する。

[RFC2246]および[SSL3]では、サーバ側でデジタル証明を使用することが求められる。また、クライアント側の証明に基づく認証も許可される。ebXML メッセージ取扱サービスハンドラは、階層およびピアツーピアによる高信頼性モデルをサポートしなければならない。

## B.3 SMTP

SMTP (Simple Mail Transfer Protocol) [SMTP]とその付属文書、[RFC822]および[ESMTP]は、一般的にインターネット電子メールとして参照される仕様一式を構成する。これらの仕様は、基本書の「上部レイヤ」となる追加機能を定めたその他の仕様により、長年にわたって強化されてきた。これらの仕様には以下のものが含まれる。

- MIME (Multipurpose Internet Mail Extensions) [RFC2045], [RFC2046], [RFC2387]
- 認証のための SMTP サービス拡張[RFC2554]
- TLS での安全な SMTP のための SMTP サービス拡張[RFC2487]

インターネット電子メール実装は、普通、以下の 2 つの「エージェント」種類から成る。

- メッセージ転送エージェント(MTA): MUA に代わって他の MTA とメッセージの送受信を行うプログラム。Microsoft Exchange Server は MTA の一例である。
- メールユーザエージェント(MUA): 電子メールプログラムを使用して、電子メールメッセージを作成し、MTA と通信してメールのメッセージを送信したり検索したりする。Microsoft Outlook は MUA の一例である。

MTA はしばしば「メールハブ」の役割を果たし、数百以上の MUA にサービスを提供することができるようになっている。

MUA は、上述のインターネット電子メール仕様に従って、電子メールを構築する。本項では、MUA の観点から、電子メールで転送するための ebXML 互換メッセージの「バインド」について説明する。MTA の観点からの SMTP での ebXML メッセージ交換のバインドについては定めない。

### B.3.1 サポートされるプロトコルの最低レベル

- Simple Mail Transfer Protocol [RFC821]および[RFC822]
- MIME [RFC2045]および[RFC2046]
- Multipart/Related MIME [RFC2387]

### B.3.2 SMTP による ebXML メッセージの送信

SMTP 上でメッセージを送信する前に、ebXML メッセージは ebXML メッセージ取扱サービス仕様の 7 項および 8 項に従ってフォーマットしなければならない。また、メッセージは、MIME [RFC2045]、[RFC2056]、そして[RFC2387]で定める構文、形式、および符号化規則に準拠しなければならない。

電子メールで転送されるデータ種類の多くは、8 ビットの文字もしくはバイナリデータで表される。そのようなデータは、メールのメッセージが 1 行 1000 文字(追跡 CRLF 行セパレタを含む)以内の 7 ビット US-ASCII データに制限される SMTP [SMTP]で送信できない。受信側の MTA や中間の MTA の処理が 7 ビットのデータに制限されていることを送信側のメッセージ取扱サービスハンドラが分かっている場合、[RFC2045]の 6 項で定められている符号化規則に従って、8 ビット(またはバイナリ)で表された文書の部分を「変換」しなければならない。受領側 MTA や中間 MTA のすべてが 8 ビットのデータを処理できることが分かっている場合、メッセージ取扱サービスハンドラは ebXML メッセージを変換する必要はない。

SMTP で転送するための ebXML メッセージの生成規則は以下の通りである。。

- [RFC821]の限定転送パスを使用する場合は、[RFC2045]の 6 項で定められた符号化規則に従って、ebXML メッセージで転送されるデータをすべて 8 ビットデータに符号化して転送する。
- 関連するパラメータが付された ebXML メッセージエンベロープの `Content-Type: Multipart/Related` MIME ヘッダは、eMail MIME ヘッダとして示さなければならない。
- eMail MIME ヘッダには、ebXML メッセージエンベロープを構成するその他すべての MIME ヘッダも含まれていなければならない。
- eMail MIME ヘッダには、SOAPAction MIME ヘッダフィールドも含まれていなければならず、その値は ebXML とすることができる。

  `SOAPAction`: **”ebXML”**

  Service と Action は、対応する ebXML ***MessageHeader*** の要素の値である。
- eMail MIME ヘッダとして、「MIME-Version 1.0」ヘッダが示されていなければならない。
- eMail ヘッダ「To:」には、[RFC822]に準拠した ebXML メッセージ取扱サービスハンドラの電子メールアドレスが含まれなければならない。
- eMail ヘッダ「From:」には、[RFC822]に準拠した送信者の ebXML メッセージ取扱サービスハンドラの電子メールアドレスが含まれなければならない。
- 電子メールヘッダ「Date:」は[RFC822]に従って作成する。
- 電子メールメッセージのヘッダには、[RFC822]および[RFC2045]に従って、その他のヘッダを含めることができるが、ebXML メッセージ取扱サービスハンドラはそれらのヘッダを無視することができる。

ebXML メッセージが含まれる電子メールメッセージの例を以下に示す。

```
From: ebXMLhandler@example.com
To: ebXMLhandler@example2.com
Date: Thu, 08 Feb 2001 19:32:11 CST
MIME-Version: 1.0
SOAPAction: "ebXML"
Content-type: multipart/related; boundary="BoundarY"; type="text/xml";
```

```
        start="<ebxhmheader111@example.com>"

--BoundarY
Content-ID: <ebxhmheader111@example.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns:eb='http://www.ebxml.org/namespaces/messageHeader'>
<SOAP-ENV:Header>
  <eb:MessageHeader SOAP-ENV:mustUnderstand="1" eb:version="1.0">
    <eb:From>
      <eb:PartyId>urn:duns:123456789</eb:PartyId>
    </eb:From>
    <eb:To>
      <eb:PartyId>urn:duns:912345678</eb:PartyId>
    </eb:To>
    <eb:CPAId>20001209-133003-28572</eb:CPAId>
    <eb:ConversationId>20001209-133003-28572</eb:ConversationId>
    <eb:Service>urn:services:SupplierOrderProcessing</eb:Service>
    <eb:Action>NewOrder</eb:Action>
    <eb:MessageData>
      <eb:MessageId>20001209-133003-28572@example.com</eb:MessageId>
      <eb:Timestamp>2001-02-15T11:12:12Z</Timestamp>
    </eb:MessageData>
    <eb:QualityOfServiceInfo eb:deliverySemantics="BestEffort"/>
  </eb:MessageHeader>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <eb:Manifest SOAP-ENV:mustUnderstand="1" eb:version="1.0">
    <eb:Reference xlink:href="cid:ebxmlpayload111@example.com"
         xlink:role="XLinkRole"
         xlink:type="simple">
        <eb:Description xml:lang="en-us">Purchase Order 1</eb:Description>
    </eb:Reference>
  </eb:Manifest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--BoundarY
Content-ID: <ebxhmheader111@example.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<purchase_order>
  <po_number>1</po_number>
  <part_number>123</part_number>
  <price currency="USD">500.00</price>
</purchase_order>

--BoundarY--
```

### B.3.3 応答メッセージ

エラーや受領通知を含む、ebXML メッセージ取扱サービスハンドラ間の ebXML 応答メッセージは、すべて非同期で配信される。各応答メッセージは、本書の「SMTP による ebXML メッセージの送信」の項で定められる規則に従って作成しなければならない。

ebXML メッセージ取扱サービスハンドラは、MTA によって送信される配信失敗通知メッセージを受信できなければならない。配信失敗通知メッセージを受信する MSH は、メッセージを検査し、メッセージの配信に失敗した ebXML メッセージと送信側 MSH を明らかにしなければならない。また、MSH は、配信失敗の原因となる違反メッセージを送信したアプリケーションの識別に努めなければならない。さらに、MSH は、メッセージ配信失敗の発生をアプリケーションに通知するようにしなければならない。違反メッセージの発生源を特定できない場合は、MSH 管理者に通知しなければならない。

受領したメッセージを有効な ebXML メッセージまたはメッセージ配信失敗として識別することができない MSH は、「無効レター」フォルダに識別不可能なメッセージを保存しなければならない。

MSH は、処分した受信メッセージをそれぞれ監査ログに記録しなければならない。

### B.3.4 アクセス管理

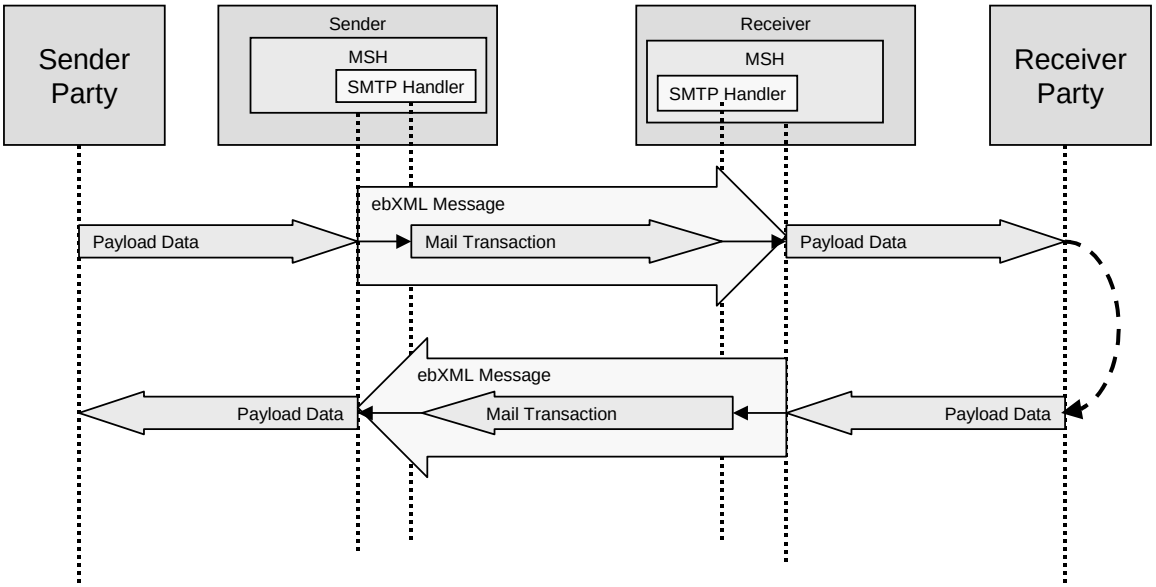
実装者は、アクセス制御の仕組みを用いて、権限のないアクセスからebXMLメッセージ取扱サービスハンドラを保護できる。「SMTP Service Extension for Authentication」[RFC2554]で説明されているSMTPアクセス認証プロセスは、権限のないアクセスからebXMLメッセージ取扱サービスハンドラを保護するために推奨されるアクセス管理の仕組みが定められている。

### B.3.5 機密性と通信プロトコルレベルのセキュリティ

ebXML メッセージ取扱サービスハンドラは、トランスポート層での暗号化を使用して、ebXML メッセージの機密性を保護できる。IETF「SMTP Service Extension for Secure SMTP over TLS」仕様[RFC2487]では、具体的な技術詳細と使用可能なオプションの一覧が示されている。

### B.3.6 SMTP モデル

ebXML メッセージ取扱サービスのメッセージは、以下の図で示す通り、すべて[SMTP]メールトランザクションのメールとして搬送される。




## B.4 高信頼性メッセージングの際の通信エラー

送信者または受信者が転送プロトコルレベルのエラー(HTTP、SMTP、または FTP のエラーなど)を検出したとき、高信頼性メッセージングが使用されている場合は、適切な転送回復ハンドラが障害回復順序

を実行する。このシーケンスで障害が回復できない場合のみ、高信頼性メッセージングの障害回復が行われる(10項参照)。

# 免責

本書の見解および仕様は著者のものであり、その雇用者の見解や仕様ではない。著者およびその雇用者は、この設計の正確または不正確な実装あるいは使用から生じた問題に対して責任を負わない。

# 著作権について



# Copyright Statement