

インターネットを利用した  
通信プロトコル利用ガイドライン  
JX 手順版

V1.0.0

インターネット EDI 普及推進協議会  
Japan internet EDI Association (JiEDIA)

## 目 次

1	インターネットを利用した通信プロトコル利用ガイドライン JX 手順版 作成の目的.....	3
2	対象プロトコルの概要.....	4
2.1	JX 手順.....	4
2.1.1	JX 手順概要 .....	4
2.1.2	JX 手順メッセージ構造.....	5
2.1.3	エラー通知.....	32
2.1.4	セキュリティ仕様.....	37
2.1.5	メッセージサンプル.....	40
2.2	推奨通信プロトコルパラメータセット .....	54
	改訂の要約.....	57

## 1 インターネットを利用した通信プロトコル利用ガイドライン JX 手順版 作成の目的

平成 17 年 3 月に財団法人流通システム開発センターよりリリースされた、メッセージ交換ガイドラインを踏襲する形でより詳細に規定が必要な項目について記述したのが、平成 20 年初版発行、平成 21 年第 2 版発行の「通信プロトコル標準化に関する調査研究報告書 インターネットを利用した通信プロトコル利用ガイドライン」である。

この度 JX 手順の仕様管理が財団法人流通システム開発センターからインターネット EDI 普及推進協議会（Japan internet EDI Association(略称：JiEDIA/ジェディア)）に移管された。

これに伴い、「インターネットを利用した通信プロトコル利用ガイドライン」から JX 手順にかかる記載を別紙としたのが本書である。

本書の内容は逐次改定を加える予定である。本書を引用する場合は、

「出典：「インターネットを利用した通信プロトコル利用ガイドライン JX 手順版」**VX.X.X**（インターネット EDI 普及推進協議会）」と出典を明記していただきたい。

## 2 対象プロトコルの概要

### 2.1 JX 手順

#### 2.1.1 JX 手順概要

ビジネス文書（データ）を転送するための3つのメソッドより構成されるシンプルな通信プロトコルである。

ビジネス文書を転送する仕組みとして SOAP(Simple Object Access Protocol)-RPC(SOAP-Remote Procedure Call)を用い、SOAP を転送するプロトコルとしては、HTTP(HyperText Transfer Protocol)を用いる。

#### (1) メソッド概要

プロトコルモデルは、クライアントを起点としサーバに対して行う以下の3つのメソッドにより構成される。

メソッド	機能
PutDocument	1 ビジネス文書をサーバに送信する機能
GetDocument	サーバにある自分宛ての未取得ビジネス文書のうち、古いものから1つ取得する機能
ConfirmDocument	取得したビジネス文書の識別 ID をサーバに通知し、取得したことを通知する機能

図表 1 JX 手順を構成する3つのメソッド

#### (2) 特徴

- サーバへインターネット経由で接続し、ビジネス文書のアップロードや、ダウンロードを実現する通信プロトコル。
- クライアント側は、安価でインターネットに接続できる PC 環境で実現可能。
- 処理の起点はクライアント側であり、クライアントからサーバへの接続により処理が開始され、ビジネス文書の送信や受信を行う。
- 低コストでインターネット EDI を実現したい企業向け。

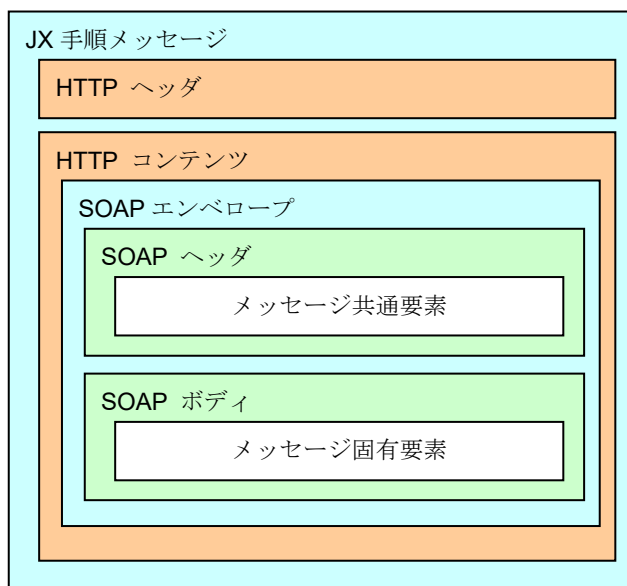
## 2.1.2 JX 手順メッセージ構造

### 2.1.2.1 シンタックスルール

PutDocument/GetDocument/ConfirmDocument の3つのメソッドは、それぞれ HTTP リクエストと HTTP レスポンス毎に以下の6つのメッセージが定義されている。

- PutDocument メッセージ
- PutDocumentResponse メッセージ
- GetDocument メッセージ
- GetDocumentResponse メッセージ
- ConfirmDocument メッセージ
- ConfirmDocumentResponse メッセージ

これらのメッセージの構成は、下図のようになる。



図表 2 メッセージ構成

(1) HTTP ヘッダの記述形式

JX 手順の HTTP ヘッダ仕様は、RFC2616 に準拠する。

HTTP ヘッダ要素	説明
POST	クライアント企業、サーバ企業が相互に決めた URL
Host	サーバ企業のドメイン名およびポート番号
Content-Length	メッセージボディの長さ (バイト数)
Content-type	エンティティボディのメディアタイプ
SOAPAction	SOAP-RPC 要求の意図

図表 3 HTTP ヘッダの記述形式

HTTP ヘッダの具体例を次に示す。

```
POST /SOAP-RPC HTTP/1.1
Host: www.xxx.co.jp
Content-Length: 1024
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/PutDocument"
```

ここに挙げた以外の HTTP ヘッダ要素については、RFC2616 を参照されたい。

## (2) SOAP ヘッダの記述形式

SOAP ヘッダ内のメッセージ共通の要素には **MessageHeader** 要素があり、すべてのメッセージは、**MessageHeader** 要素を含む必要がある。

MessageHeader 要素	説明
From	メッセージ送信元の URI、又は、メールアドレス、GLN 等の、企業をユニークに識別できる情報を記載する。
To	メッセージ送信先の URI、又は、メールアドレス、GLN 等の、企業をユニークに識別できる情報を記載する。
MessageId	メッセージの一意性を保持するための識別子 形式は、“ドメイン内でユニークとなる文字列@ドメイン名”を推奨する。
Timestamp	メッセージを作成した世界協定時（UTC）による日時 形式は、“YYYY-MM-DDThh:mm:ss”とする。
OptionalFormatType(*)	<b>GetDocument</b> メッセージの際、取得対象となるビジネス文書の形式を限定するためのフォーマットタイプを指定する。オプション。
OptionalDocumentType(*)	<b>GetDocument</b> メッセージの際、取得対象となるビジネス文書の種別を限定するためのドキュメントタイプを指定する。オプション。

図表 4 SOAP ヘッダの記述形式

(\*)OptionalFormatType 要素、OptionalDocumentType 要素の使用上の注意

- OptionalFormatType 要素、OptionalDocumentType 要素の両方を指定するか、両方を省略するかのどちらかである。片方だけの指定は認めない。
- **GetDocument** の場合にのみ有効となる。**GetDocument** 以外のリクエストメッセージ、及び、レスポンスメッセージの場合は、これらの要素を無視し、機能上に影響しないものとする。
- 両方を省略した場合の **GetDocument** は、ビジネス文書の形式と種別を指定しない **GetDocument** 仕様となる。すなわち自企業宛のすべてのビジネス文書が対象となる。

## (3) SOAP ボディの記述形式

SOAP ボディ内のメッセージ固有の要素には、以下の要素があり、いずれかの要素が含まれる。

- PutDocument 要素
- PutDocumentResponse 要素
- GetDocument 要素
- GetDocumentResponse 要素
- ConfirmDocument 要素
- ConfirmDocumentResponse 要素

主要な要素の説明を以下に記す。

### ① PutDocument 要素

PutDocument 要素は、PutDocument メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

PutDocument 要素	説明
MessageId	送信ビジネス文書の一意性を保持するための識別子 重複送信検出に用いる。形式は、“ドメイン内でユニークとなる文字列@ドメイン名”を推奨する。
Data	Base64 にエンコードされた送信ビジネス文書
SenderId	ビジネス文書送信企業の識別子
ReceiverId	ビジネス文書受信企業の識別子
FormatType(*)	業界毎に定めたビジネス文書の形式
DocumentType(*)	業界毎に定めたビジネス文書の種別
CompressType	ビジネス文書の圧縮・解凍形式 ビジネス文書の圧縮・解凍は、上位アプリケーションで実施する。 ビジネス文書が圧縮されている場合、その圧縮形式を指定する。 [図表 11 圧縮形式指定の代表的な MIME メディアタイプ]参照

図表 5 PutDocument の要素

(\*)FormatType 要素、DocumentType 要素の使用上の注意

- ・ PutDocument メッセージの、FormatType と、DocumentType の両要素の値が、サーバ側で登録されているものかを確認する。登録がない場合、SOAP Fault を返す。

尚、業界毎に定義しているビジネス文書以外のビジネス文書を交換する場合は、当事者間で事前に使用するビジネス文書の形式と種別の文字列を決定し、サーバに登録しておくこと。



② PutDocumentResponse 要素

PutDocumentResponse 要素は、PutDocumentResponse メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

PutDocumentResponse 要素	説明
PutDocumentResult	PutDocument メソッドの結果  true か false のどちらかを指定する。サーバ側はビジネス文書を正常に受信できた場合、true を返す。同じ MessageId により重複受信を検知した場合は、false を返す。

図表 6 PutDocumentResponse の要素

③ GetDocument 要素

GetDocument 要素は、GetDocument メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

GetDocument 要素	説明
ReceiverId	ビジネス文書受信企業の識別子

図表 7 GetDocument の要素

#### ④ GetDocumentResponse 要素

GetDocumentResponse 要素は、GetDocumentResponse メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

GetDocumentResponse 要素	説明
GetDocumentResult	GetDocument メソッドの結果  true か false のどちらかを指定する。サーバ側はビジネス文書が存在して送信できた場合、true を返す。ビジネス文書が存在しない場合は、false を返す。  尚、GetDocument における、ビジネス文書の形式と種別限定の指定方法に関しては表下 (※) 注意を参照。
MessageId	受信ビジネス文書の一意性を保持するための識別子  形式は、“ドメイン内でユニークとなる文字列@ドメイン名”を推奨する。クライアントがビジネス文書を正常に受信出来たことをサーバに通知するために、この識別子を受信確定通知で用いる。
Data	Base64 にエンコードされた受信ビジネス文書
SenderId	ビジネス文書送信企業の識別子
ReceiverId	ビジネス文書受信企業の識別子
FormatType(*)	業界毎に定めたビジネス文書の形式
DocumentType(*)	業界毎に定めたビジネス文書の種別
CompressType	ビジネス文書の圧縮・解凍形式。  ビジネス文書の圧縮・解凍は、上位アプリケーションで実施する。  ビジネス文書が圧縮されている場合、その圧縮形式を指定する。  [図表 11 圧縮形式指定の代表的な MIME メディアタイプ]参照

図表 8 GetDocumentResponse の要素

(※)注意

J X 手順クライアント動作：

- 取得するビジネス文書の形式と種別を限定しない場合は、SOAP ヘッダ部に OptionalFormatType、OptionalDocumentType の両要素のない GetDocument リクエストメッセージを送信する。
- ビジネス文書の形式と種別を限定する場合は、SOAP ヘッダ部の OptionalFormatType、OptionalDocumentType の両要素に、取得したいビジネス文書の形式と種別を指定する。

J X 手順サーバ動作：

- SOAP ヘッダ部に OptionalFormatType、OptionalDocumentType 両要素がない場合は、サーバ側のビジネス文書を蓄えている永続記憶に存在するビジネス文書中で、未受信、かつ、最も登録の古いものを送信する。

ビジネス文書が存在しない場合は、**GetDocumentResult** 要素で **false** を返す。

- **SOAP** ヘッダ部に **OptionalFormatType**、**OptionalDocumentType** 両要素が指定された場合は、ビジネス文書を蓄えている永続記憶に存在するビジネス文書中で、指定されたビジネス文書の形式と種別を持つ、未受信、かつ、最も登録の古いものを送信する。指定されたビジネス文書の形式と種別を持つビジネス文書が存在しない場合、**GetDocumentResult** 要素で **false** を返す。
- 尚、**OptionalFormatType**、**OptionalDocumentType** が片方だけの指定の場合、**SOAP Fault** を応答する。
- **GutDocument** メッセージの、**OptionalFormatType**、**OptionalDocumentType** の両要素の値が、サーバ側で登録されているものかを確認する。登録がない場合、**SOAP Fault** を返す。尚、業界毎に定義しているビジネス文書以外のビジネス文書を交換する場合は、当事者間で事前に使用するビジネス文書の形式と種別の文字列を決定し、サーバに登録しておくこと。

⑤ ConfirmDocument 要素

ConfirmDocument 要素は、ConfirmDocument メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

ConfirmDocument 要素	説明
MessageId	GetDocument メソッドで取得したビジネス文書の MessageId を指定する。
SenderId	ビジネス文書送信企業の識別子
ReceiverId	ビジネス文書受信企業の識別子

図表 9 ConfirmDocument の要素

⑥ ConfirmDocumentResponse 要素

ConfirmDocumentResponse 要素は、ConfirmDocumentResult メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

ConfirmDocumentResponse 要素	説明
ConfirmDocumentResult	ConfirmDocument メソッドの結果  true か false のどちらかを指定する。サーバ側は正常に受信確定した場合、true を返す。同じ MessageId が重複して通知された場合は、false を返す。

図表 10 ConfirmDocumentResponse の要素

#### ※ compressType 要素

ビジネス文書の圧縮は、業務アプリケーション(レベル)で実施する。JX 手順では、送信されるビジネス文書が圧縮されているか否かを指定する。ビジネス文書が圧縮されている場合、PutDocument メッセージ等の CompressType 要素にその圧縮形式を指定する。圧縮されていない場合、長さ 0 の文字列を指定する。また、圧縮対象は 1 ファイルのみとする。圧縮形式（アーカイブ形式）は、IANA(The Internet Assigned Numbers Authority: <https://www.iana.org/>)によって管理されている公式の MIME メディアタイプを使用して指定する。ここで IANA は、インターネット上のプロトコルに関する様々なパラメタ(数値や記号)を管理する団体である。

代表的な MIME メディアタイプを下記に示す。

圧縮形式	拡張子	説明	MIME メディアタイプ
JAR	jar	Java Archiver 形式	application/java-archiver
Tar	tar	Tape Archiver 形式	application/x-tar
ZIP	zip	ZIP 形式	application/zip
GZIP	gz	GNU ZIP 形式	application/gzip

図表 11 圧縮形式指定の代表的な MIME メディアタイプ（参考）

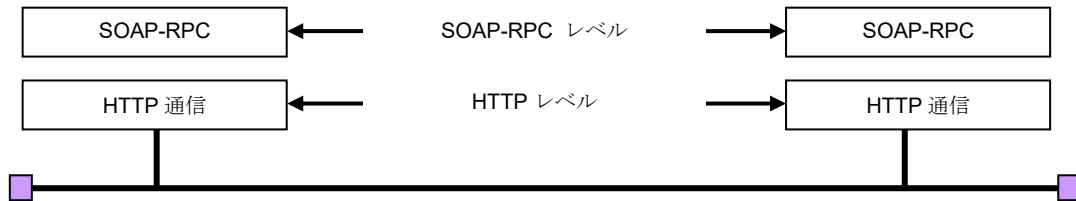
尚、圧縮されているビジネス文書を伸張(解凍)する際にパスワードの入力が必要な場合には、事前に通信相手にその旨を通知しなければいけない。

### 2.1.2.2 シーケンス

#### (1) 階層別のシーケンス

本節では、JX 手順における通信のシーケンスについて述べる。

JX 手順によるビジネス文書の送受信は、図表 12 のようなレイヤ構造になる。尚、SOAP-RPC レベルでは、ビジネス文書の保存や状態変更の業務アプリケーションレベルの内容まで含まれる。

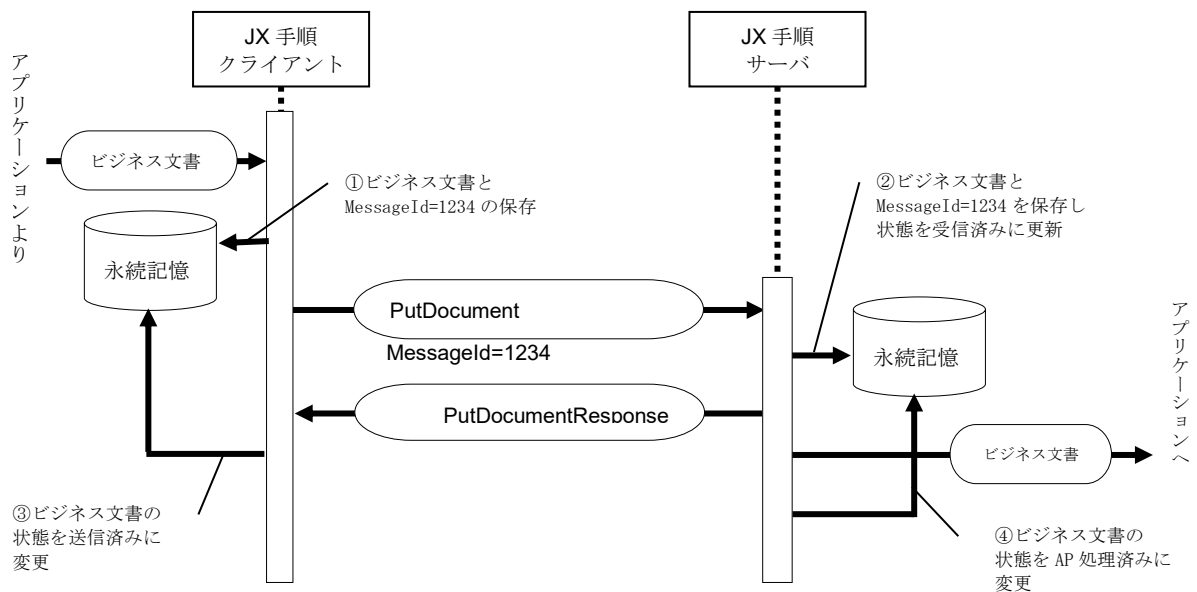


図表 12 JX 手順通信レイヤ構造

#### (a) JX 手順レベル（SOAP-RPC レベル）のシーケンス

##### ● ビジネス文書の送信

JX 手順で規定されている送信の流れを図表 13 に示す。この例は通信経路上で異常が発生しなかった場合である。



図表 13 JX 手順レベルの送信シーケンス

クライアント側は、送信するビジネス文書と MessageId を永続記憶に保存（①）し、PutDocument でビジネス文書を送信する。

サーバ側は、受信したビジネス文書と MessageId を永続記憶に保存（②）後、ビジネス文書の状態を受信済みに更新し PutDocumentResponse を返す。尚、PutDocument に異常があった場合は、PutDocumentResponse の代わりに SOAP Fault を返す。

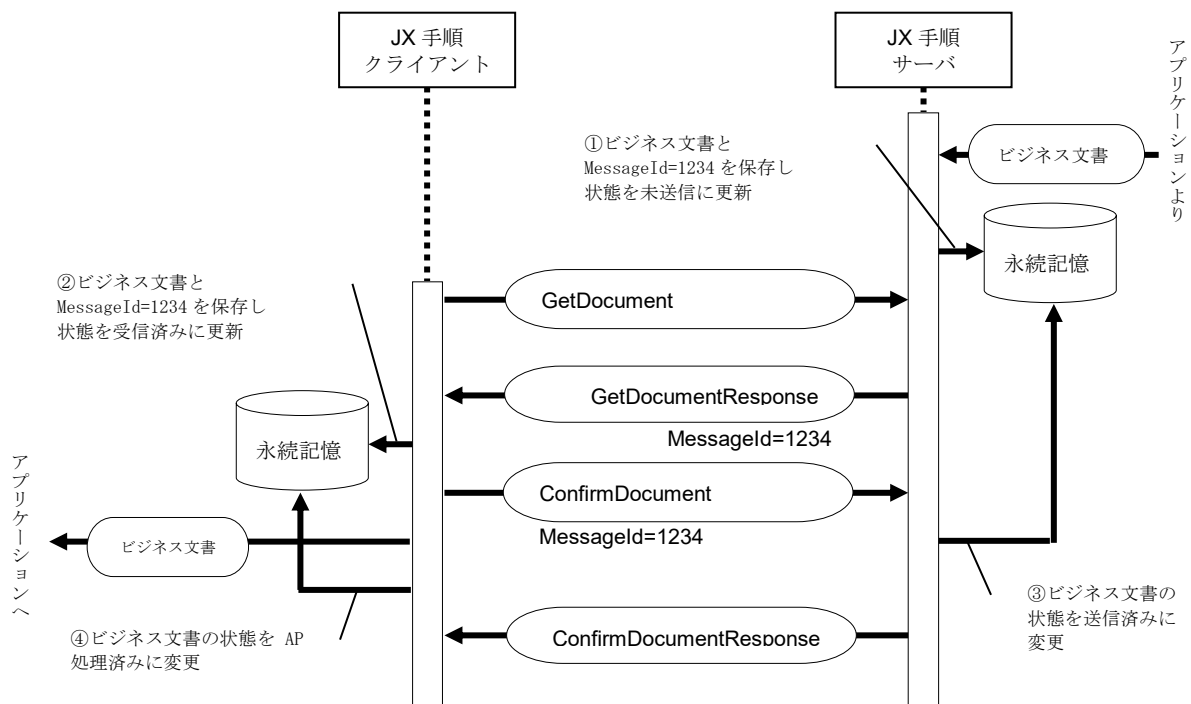
クライアント側は、PutDocumentResponse によりビジネス文書の状態を送信済みに変更する（③）。

サーバ側は、ビジネス文書をアプリケーションに渡した後に、ビジネス文書の状態をアプリケーション処理済みに変更する (④)。

JX 手順では、永続記憶を利用した信頼性保証機能を実現している。信頼性保証機能の詳細については、(2) 信頼性保証機能を参照されたい。

- ビジネス文書の受信

JX 手順で規定されている受信の流れを図表 14 に示す。この例は通信経路上で異常が発生しなかった場合である。



図表 14 JX 手順レベルの受信シーケンス

サーバ側は、送信するビジネス文書と MessageId を永続記憶に保存し、ビジネス文書の状態を未送信にする (①)。

クライアント側より GetDocument でビジネス文書の受信要求を行う。

サーバ側は、未送信の古いビジネス文書から GetDocumentResponse でクライアント側にビジネス文書を渡す。尚、送信ビジネス文書が無い場合は、GetDocumentResponse で GetDocumentResult= false を返さなければならない。また、GetDocument に異常があった場合は、GetDocumentResponse の代わりに SOAP Fault を返す。

クライアント側は、GetDocumentResponse により受信したビジネス文書と MessageId を永続記憶に保存 (②) 後、ビジネス文書の状態を受信済みに更新し ConfirmDocument で受信確定通知をサーバ側に送信する。

サーバ側は、ビジネス文書の状態を送信済みに変更 (③) し、ConfirmDocumentResponse をクライアント側に返す。尚、ConfirmDocument に異常があった場合は、ConfirmDocumentResponse の代わりに SOAP Fault を返す。

クライアント側は、ビジネス文書をアプリケーションに渡した後に、ビジネス文書の状態をアプリケーション処理済みに変更する (④)。

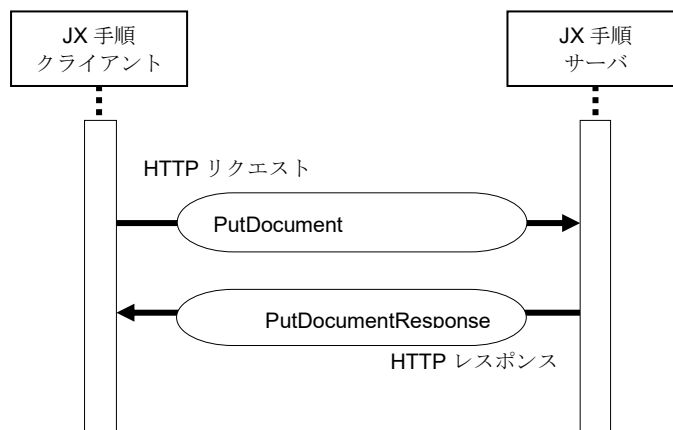
サーバ側に未送信のビジネス文書が複数あった場合は、GetDocument と ConfirmDocument を交互に呼び出す必要がある。尚、GetDocument 後、ConfirmDocument を実行せずに GetDocument を実行すると前回の GetDocument で受信したビジネス文書が受信される。

JX 手順では、永続記憶を利用した信頼性保証機能を実現している。信頼性保証機能の詳細については、(2) 信頼性保証機能を参照されたい。

(b) HTTP レベルのシーケンス

- ビジネス文書の送信

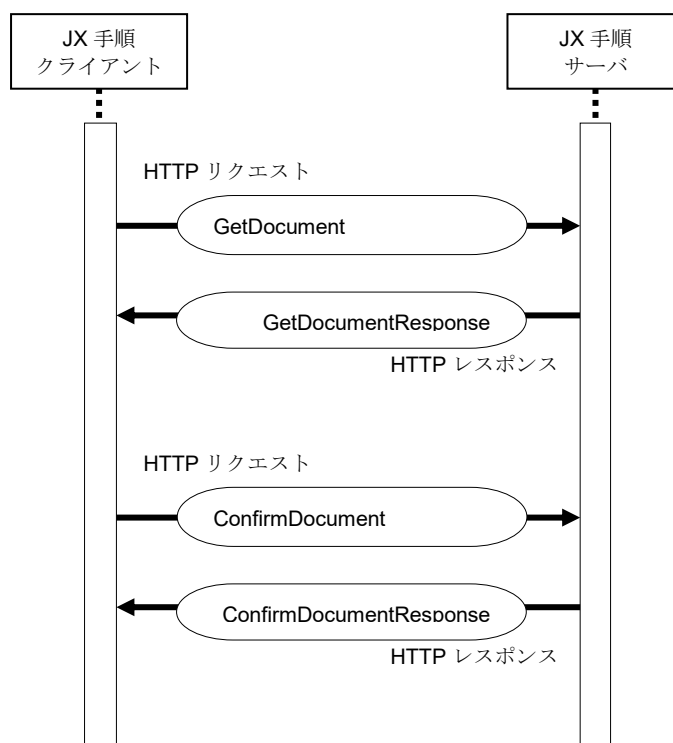
HTTP レベルでのビジネス文書の送信シーケンスを図表 15 に示す。



図表 15 HTTP レベルの送信シーケンス

- ビジネス文書の受信

HTTP レベルでのビジネス文書の受信シーケンスを図表 16 に示す。





図表 16 HTTP レベルの受信シーケンス

(2) 信頼性保証機能

インターネットでは相手システムまでの通信経路の信頼性を保証することができない。そのため、JX 手順では、ビジネス文書を確実に送り届けるための仕組みである信頼性保証機能を備えている。そのため、JX 手順レベルの階層で通信の信頼性を保証することができる。

JX 手順で規約されている信頼性保証機能は以下の通りである。

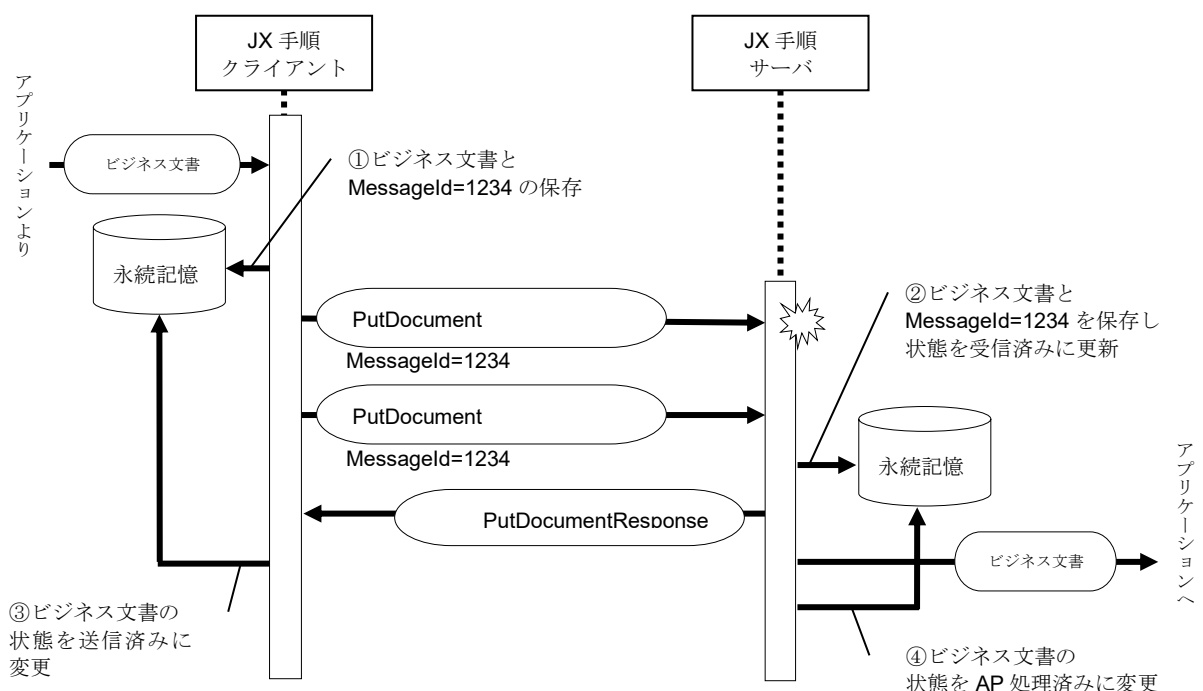
種別	内容
欠落防止	通信経路上の異常により、送信されたデータが受信側に到達しなかった場合、それを検出して再度データを送信する。
重複破棄	先発のデータと再送したデータの両方が受信側システムに到達した場合に、受信側のアプリケーションに同じデータを渡さない。

図表 17 信頼性保証機能

(a) 欠落防止

● ビジネス文書の送信

クライアント側は、PutDocument でビジネス文書を送信後、SOAP Fault を受け取った場合や PutDocumentResponse がクライアントで定める所定時間を過ぎても返って来なかった場合には、PutDocument を再送信しなければならない。尚、再送信はクライアントが定める所定回数リトライを行い、それでも送信できない場合はリトライオーバーとする。

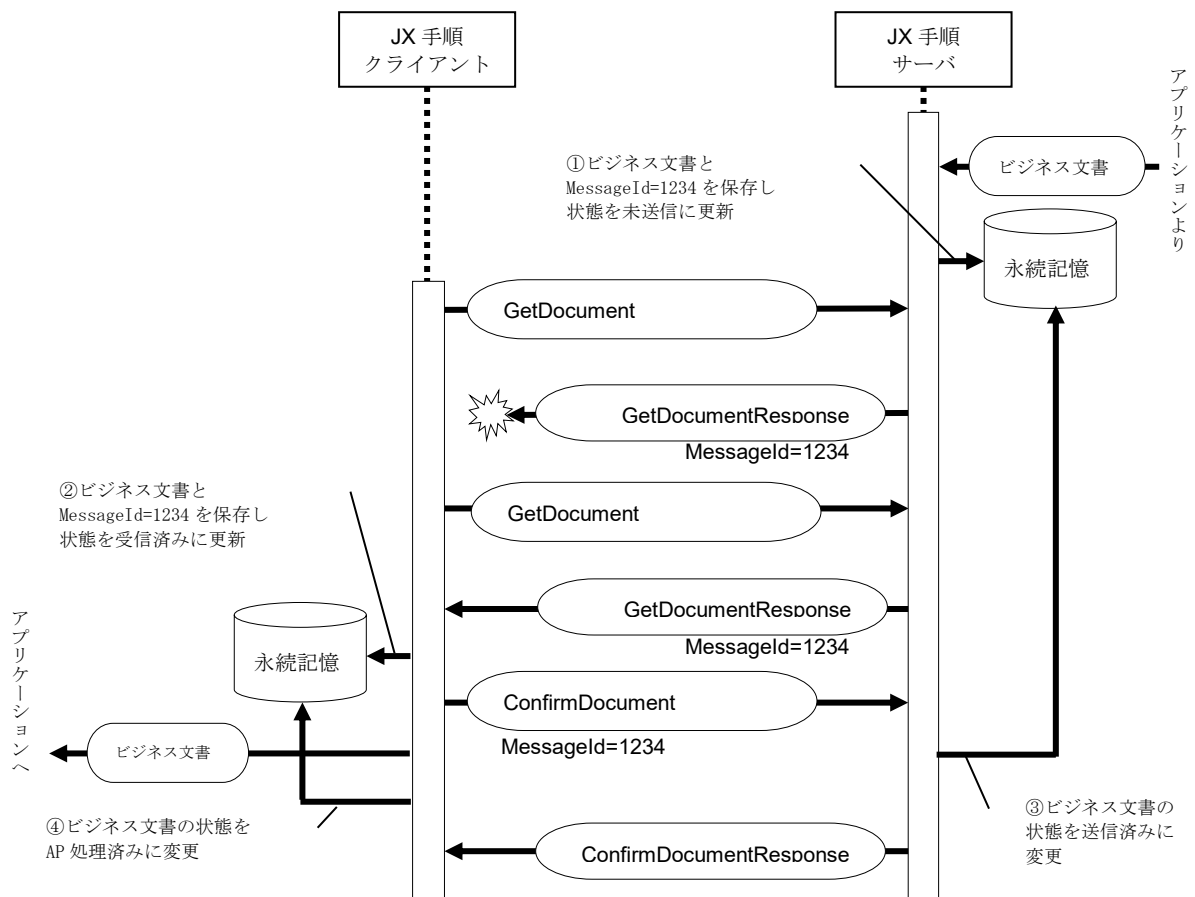


図表 18 ビジネス文書送信時の欠落防止

- ビジネス文書の受信

クライアント側は、**GetDocument** でビジネス文書の受信要求後、**GetDocumentResponse** がクライアントで定める所定時間を過ぎてても返って来なかった場合には、**GetDocument** で再受信要求しなければならない。尚、再受信要求はクライアントが定める所定回数リトライを行い、それでも受信できない場合はリトライオーバーとする。

サーバ側は、**GetDocumentResponse** 後 **ConfirmDocument** を受信する前に、**GetDocument** を受信した場合は前回のビジネス文書を **GetDocumentResponse** で返さなければならない。



図表 19 ビジネス文書受信時の欠落防止

(b) 重複破棄

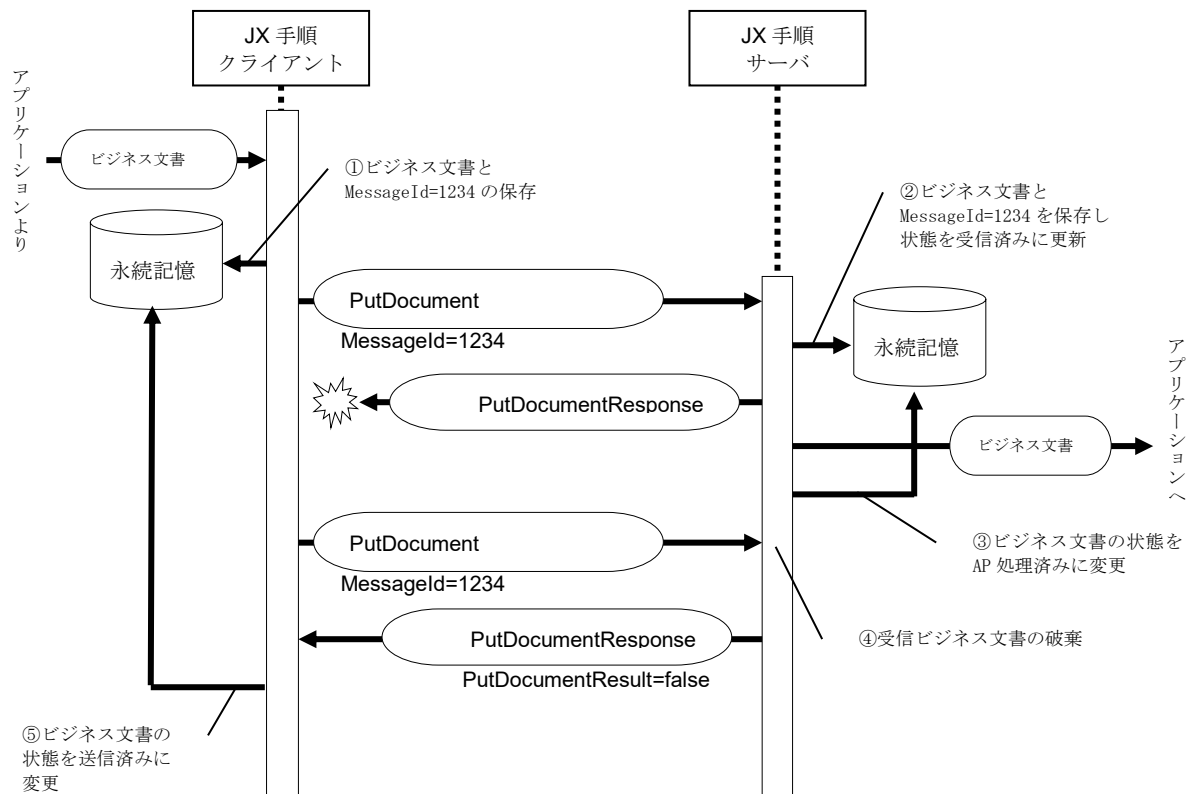
● ビジネス文書の送信

クライアント側は、PutDocument でビジネス文書を送信後、PutDocumentResponse がクライアントで定める所定時間を過ぎても返って来なかったため、PutDocument を再送信する。

サーバ側は 1 回目の PutDocument でビジネス文書を既に受信済みのため、2 回目の受信ビジネス文書を破棄し、PutDocumentResponse で PutDocumentResult=false を返さなければならない。ビジネス文書の重複チェックは PutDocument の MessageId を使用する。

クライアント側は、PutDocumentResponse が PutDocumentResult=false の場合、既に送信済みと見なし、ビジネス文書の状態を送信済みに変更する。

サーバ側での重複処理のための MessageId を保持する期間はサーバ企業が決定し、クライアント企業に通知しなければならない。



図表 20 ビジネス文書送信時の重複破棄

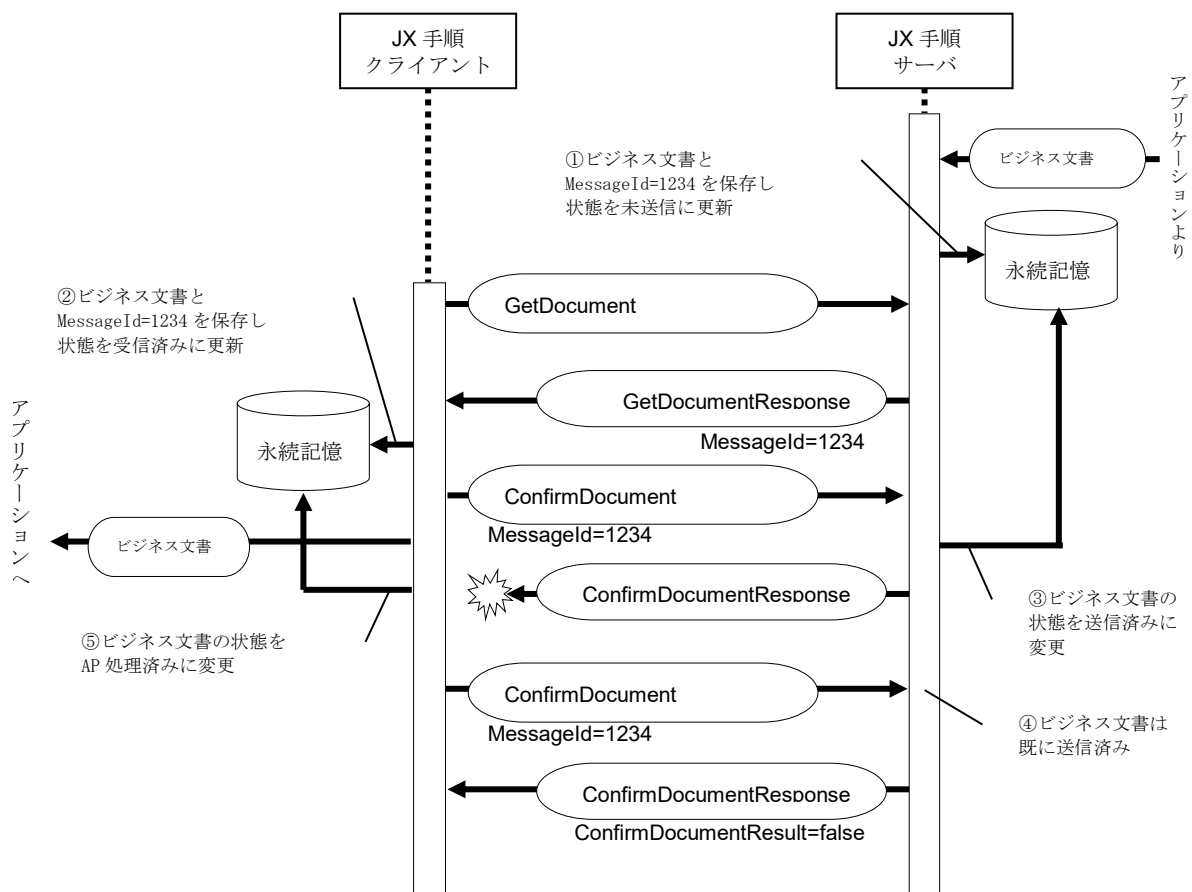
- ビジネス文書の受信

クライアント側は **ConfirmDocument** 送信後、サーバ側からの **ConfirmDocumentResponse** がクライアントで定める所定時間を過ぎても返って来なかったため、**ConfirmDocument** を再送信する。

サーバ側は 1 回目の **ConfirmDocument** でビジネス文書は既に送信済みのため、**ConfirmDocumentResponse** で **ConfirmDocumentResult=false** を返さなければならない。尚、**ConfirmDocument** で指定された **MessageId** が不明（サーバ側からの **GetDocumentResponse** で返した **MessageId** 以外）な場合は、**ConfirmDocumentResponse** の代わりに **SOAP Fault** を返す。

クライアント側は、**ConfirmDocumentResponse** が **ConfirmDocumentResult=false** の場合は、受信確定通知済みと見なし、**ConfirmDocumentResponse** が正常時と同様の処理を行う。

尚、クライアント側は、**GetDocumentResponse** により受信したビジネス文書を永続記憶に保存後、**ConfirmDocument** が完了する前に **GetDocument** を送信し、サーバ側から前回と同じビジネス文書が返された場合は、既に受信済みのため破棄しなければならない。



図表 21 受信確定通知の重複処理

### 2.1.2.3 メッセージ交換定義（WSDL）

WSDL（Web Services Description Language）は、Web サービスを記述するための、XML をベースとした言語仕様である。それぞれの Web サービスがどのような機能を持つのか、それを利用するためにはどのような要求をすればいいのか等を記述する方法が定義されている。

#### ・ JX 手順の WSDL

JX 手順（旧称、SOAP-RPC）の WSDL には、2004 年度版と 2007 年度版がある。2007 年度版は、ビジネス文書の形式と種別を限定した取得機能を 2004 年度版との互換性を考慮して作成した WSDL である。2004 年度版と 2007 年度版のそれぞれの JX 手順クライアントと JX 手順サーバの間の接続可否の関係を下表に示す。

	2004 年度版 WSDL の JX 手順サーバ	2007 年度版 WSDL の JX 手順サーバ
2004 年度版 WSDL の JX 手順クライアント	ビジネス文書の形式と種別を限定しない機能で接続できる	ビジネス文書の形式と種別を限定しない機能で接続できる
2007 年度版 WSDL の JX 手順クライアント	ビジネス文書の形式と種別を限定しない機能で接続できる (サーバ製品の仕様に依存)	ビジネス文書の形式と種別を限定した機能で接続できる。

2004 年度版：

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:s0="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
targetNamespace="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <s:element name="PutDocument">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="messageId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="data" type="s:base64Binary" />
            <s:element minOccurs="1" maxOccurs="1" name="senderId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="formatType" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </types>
```

```

        <s:element minOccurs="1" maxOccurs="1" name="documentType" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="compressType" type="s:string" />
    </s:sequence>
</s:complexType>
</s:element>
<s:element name="PutDocumentResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="PutDocumentResult" type="s:boolean" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="MessageHeader" type="s0:MessageHeader" />
<s:complexType name="MessageHeader">
    <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="From" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="To" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="MessageId" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Timestamp" type="s:string" />
    </s:sequence>
</s:complexType>
<s:element name="GetDocument">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetDocumentResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetDocumentResult" type="s:boolean" />
            <s:element minOccurs="1" maxOccurs="1" name="messageId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="data" type="s:base64Binary" />
            <s:element minOccurs="1" maxOccurs="1" name="senderId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="formatType" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="documentType" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="compressType" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="ConfirmDocument">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="messageId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="senderId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="ConfirmDocumentResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="ConfirmDocumentResult" type="s:boolean" />
        </s:sequence>
    </s:complexType>

```

```

        </s:element>
    </s:schema>
</types>
<message name="PutDocumentSoapIn">
    <part name="parameters" element="s0:PutDocument" />
</message>
<message name="PutDocumentSoapOut">
    <part name="parameters" element="s0:PutDocumentResponse" />
</message>
<message name="PutDocumentMessageHeader">
    <part name="MessageHeader" element="s0:MessageHeader" />
</message>
<message name="GetDocumentSoapIn">
    <part name="parameters" element="s0:GetDocument" />
</message>
<message name="GetDocumentSoapOut">
    <part name="parameters" element="s0:GetDocumentResponse" />
</message>
<message name="GetDocumentMessageHeader">
    <part name="MessageHeader" element="s0:MessageHeader" />
</message>
<message name="ConfirmDocumentSoapIn">
    <part name="parameters" element="s0:ConfirmDocument" />
</message>
<message name="ConfirmDocumentSoapOut">
    <part name="parameters" element="s0:ConfirmDocumentResponse" />
</message>
<message name="ConfirmDocumentMessageHeader">
    <part name="MessageHeader" element="s0:MessageHeader" />
</message>
<portType name="JXMSTransferSoap">
    <operation name="PutDocument">
        <documentation>ドキュメントの送信(Client To Server)</documentation>
        <input message="s0:PutDocumentSoapIn" />
        <output message="s0:PutDocumentSoapOut" />
    </operation>
    <operation name="GetDocument">
        <documentation>ドキュメントの受信(Client From Server)</documentation>
        <input message="s0:GetDocumentSoapIn" />
        <output message="s0:GetDocumentSoapOut" />
    </operation>
    <operation name="ConfirmDocument">
        <documentation>ドキュメントの受信確認</documentation>
        <input message="s0:ConfirmDocumentSoapIn" />
        <output message="s0:ConfirmDocumentSoapOut" />
    </operation>
</portType>
<binding name="JXMSTransferSoap" type="s0:JXMSTransferSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
    <operation name="PutDocument">
        <soap:operation soapAction="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/PutDocument"
style="document" />
        <input>
            <soap:body use="literal" />
            <soap:header message="s0:PutDocumentMessageHeader" part="MessageHeader" use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
            <soap:header message="s0:PutDocumentMessageHeader" part="MessageHeader" use="literal" />
        </output>
    </operation>
    <operation name="GetDocument">
        <soap:operation soapAction="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/GetDocument"
style="document" />
        <input>
            <soap:body use="literal" />
            <soap:header message="s0:GetDocumentMessageHeader" part="MessageHeader" use="literal" />

```



```

    </input>
    <output>
      <soap:body use="literal" />
      <soap:header message="s0:GetDocumentMessageHeader" part="MessageHeader" use="literal" />
    </output>
  </operation>
  <operation name="ConfirmDocument">
    <soap:operation soapAction="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/ConfirmDocument"
style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="s0:ConfirmDocumentMessageHeader" part="MessageHeader" use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
      <soap:header message="s0:ConfirmDocumentMessageHeader" part="MessageHeader" use="literal" />
    </output>
  </operation>
</binding>
<service name="JXMSTransfer">
  <documentation>SOAP-RPC メッセージ転送サービス</documentation>
  <port name="JXMSTransferSoap" binding="s0:JXMSTransferSoap">
    <soap:address location="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/JXMSTransfer" />
  </port>
</service>
</definitions>

```

※ 上記locationの下線部は、実際のサーバURIに置き換えて使用される。

2007 年度版 :

<?xml version="1.0" encoding="utf-8"?>

<!-- JX 手順 WSDL 2008 年 3 月版 -->

```
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:s0="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  targetNamespace="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
```

<types>

```
<s:schema elementFormDefault="qualified" targetNamespace="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
```

```
<s:element name="PutDocument">
```

```
<s:complexType>
```

```
<s:sequence>
```

```
<s:element minOccurs="1" maxOccurs="1" name="messageId"
  type="s:string" />
```

```
<s:element minOccurs="1" maxOccurs="1" name="data"
  type="s:base64Binary" />
```

```
<s:element minOccurs="1" maxOccurs="1" name="senderId" type="s:string" />
```

```
<s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
```

```
<s:element minOccurs="1" maxOccurs="1" name="formatType"
  type="s:string" />
```

```
<s:element minOccurs="1" maxOccurs="1" name="documentType" type="s:string" />
```

```
<s:element minOccurs="1" maxOccurs="1" name="compressType" type="s:string" />
```

```
</s:sequence>
```

```
</s:complexType>
```

```
</s:element>
```

```
<s:element name="PutDocumentResponse">
```

```
<s:complexType>
```

```
<s:sequence>
```

```
<s:element minOccurs="1" maxOccurs="1" name="PutDocumentResult" type="s:boolean" />
```

```
</s:sequence>
```

```
</s:complexType>
```

```
</s:element>
```

```
<s:element name="MessageHeader" type="s0:MessageHeader" />
```

```
<s:complexType name="MessageHeader">
```

```
<s:sequence>
```

```
<s:element minOccurs="1" maxOccurs="1" name="From" type="s:string" />
```

```
<s:element minOccurs="1" maxOccurs="1" name="To" type="s:string" />
```

```
<s:element minOccurs="1" maxOccurs="1" name="MessageId" type="s:string" />
```

```
<s:element minOccurs="1" maxOccurs="1" name="Timestamp" type="s:string" />
```

```
<s:element minOccurs="0" maxOccurs="1" name="OptionalFormatType"
  type="s:string" />
```

```
<s:element minOccurs="0" maxOccurs="1" name="OptionalDocumentType"
  type="s:string" />
```

```
</s:sequence>
```

```
</s:complexType>
```

```
<s:element name="GetDocument">
```

```

    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetDocumentResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="GetDocumentResult" type="s:boolean" />
        <s:element minOccurs="1" maxOccurs="1" name="messageId" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="data"
          type="s:base64Binary" />
        <s:element minOccurs="1" maxOccurs="1" name="senderId" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="formatType"
          type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="documentType" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="compressType" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="ConfirmDocument">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="messageId" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="senderId" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="ConfirmDocumentResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="ConfirmDocumentResult" type="s:boolean"
          />
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
</types>
<message name="PutDocumentSoapIn">
  <part name="parameters" element="s0:PutDocument" />
</message>
<message name="PutDocumentSoapOut">
  <part name="parameters" element="s0:PutDocumentResponse" />
</message>
<message name="PutDocumentMessageHeader">
  <part name="MessageHeader" element="s0:MessageHeader" />
</message>
<message name="GetDocumentSoapIn">
  <part name="parameters" element="s0:GetDocument" />
</message>

```

```

<message name="GetDocumentSoapOut">
  <part name="parameters" element="s0:GetDocumentResponse" />
</message>
<message name="GetDocumentMessageHeader">
  <part name="MessageHeader" element="s0:MessageHeader" />
</message>
<message name="ConfirmDocumentSoapIn">
  <part name="parameters" element="s0:ConfirmDocument" />
</message>
<message name="ConfirmDocumentSoapOut">
  <part name="parameters" element="s0:ConfirmDocumentResponse" />
</message>
<message name="ConfirmDocumentMessageHeader">
  <part name="MessageHeader" element="s0:MessageHeader" />
</message>
<portType name="JXMSTransferSoap">
  <operation name="PutDocument">
    <documentation>ドキュメントの送信(Client To Server)</documentation>
    <input message="s0:PutDocumentSoapIn" />
    <output message="s0:PutDocumentSoapOut" />
  </operation>
  <operation name="GetDocument">
    <documentation>ドキュメントの受信(Client From Server)</documentation>
    <input message="s0:GetDocumentSoapIn" />
    <output message="s0:GetDocumentSoapOut" />
  </operation>
  <operation name="ConfirmDocument">
    <documentation>ドキュメントの受信確認</documentation>
    <input message="s0:ConfirmDocumentSoapIn" />
    <output message="s0:ConfirmDocumentSoapOut" />
  </operation>
</portType>
<binding name="JXMSTransferSoap" type="s0:JXMSTransferSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <operation name="PutDocument">
    <soap:operation soapAction="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/PutDocument"
      style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="s0:PutDocumentMessageHeader" part="MessageHeader" use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
      <soap:header message="s0:PutDocumentMessageHeader" part="MessageHeader" use="literal" />
    </output>
  </operation>
  <operation name="GetDocument">
    <soap:operation soapAction="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/GetDocument"
      style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="s0:GetDocumentMessageHeader" part="MessageHeader" use="literal" />
    </input>
  </operation>
</binding>

```

```

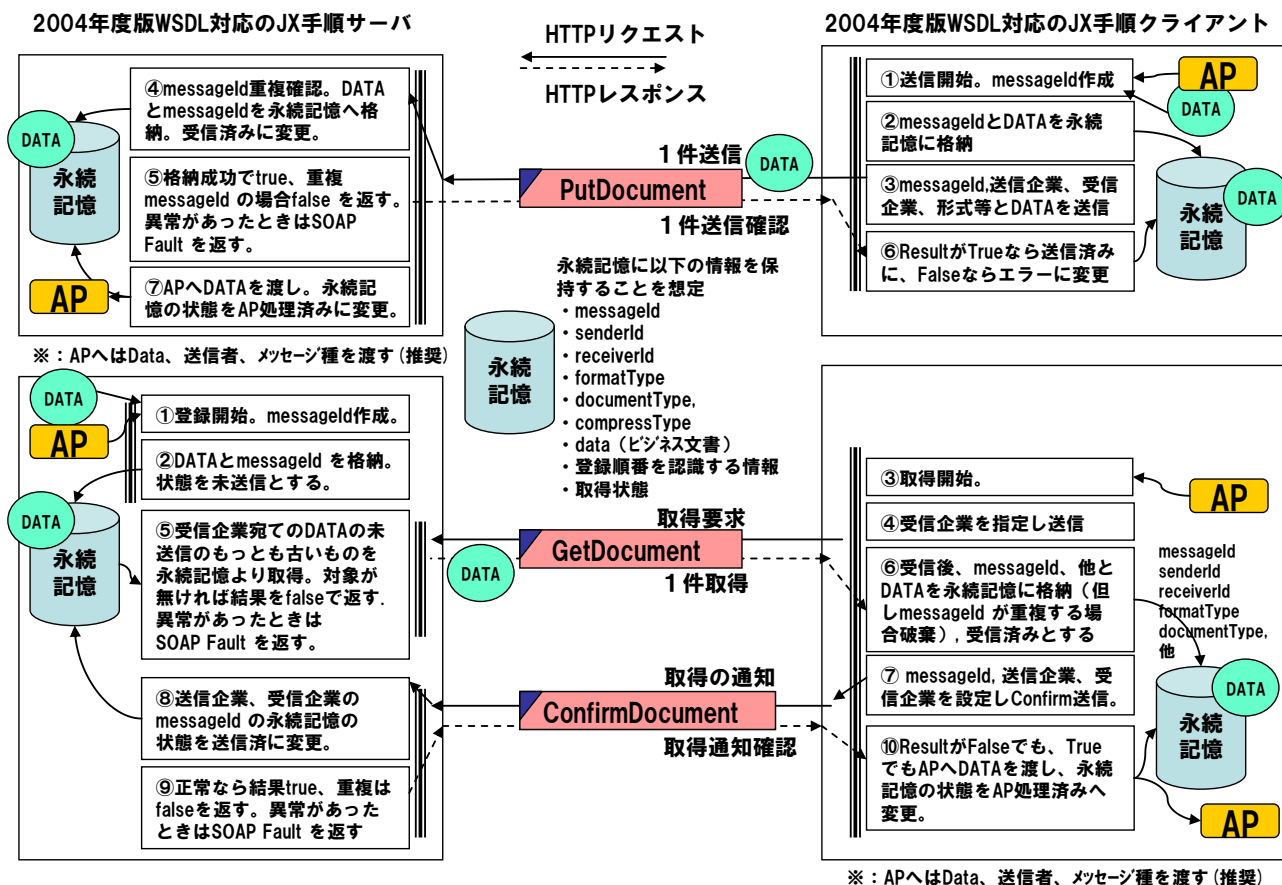
<output>
  <soap:body use="literal" />
  <soap:header message="s0:GetDocumentMessageHeader" part="MessageHeader" use="literal" />
</output>
</operation>
<operation name="ConfirmDocument">
  <soap:operation soapAction="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-
server/ConfirmDocument" style="document" />
  <input>
    <soap:body use="literal" />
    <soap:header message="s0:ConfirmDocumentMessageHeader" part="MessageHeader" use="literal"
    />
  </input>
  <output>
    <soap:body use="literal" />
    <soap:header message="s0:ConfirmDocumentMessageHeader" part="MessageHeader" use="literal"
    />
  </output>
</operation>
</binding>
<service name="JXMSTransfer">
  <documentation>JX 手順メッセージ転送サービス</documentation>
  <port name="JXMSTransferSoap" binding="s0:JXMSTransferSoap">
    <soap:address location="http://XXXX/XXX" />
  </port>
</service>
</definitions>

```

※上記の下線部が、2004 年度版 WSDL との差分である。location 属性の下線部分は、実際のサーバ URI に置き換えて使用される。

## 2. 1. 2. 4 JX 手順の動作の全体フロー

下図に、ビジネス文書の形式と種別を限定しない 2004 年度版 JX 手順の全体処理フローを示す。



※PutDocumentは同様のため省略

※OptionalFormatType / OptionalDocumentType: 種別指定

```

graph TD
    subgraph Server [2007年度版WSDL対応のJX手順サーバ]
        direction TB
        S1[①登録開始。messageld作成。]
        S2[②DATAとmessageld、他を格納。状態を未送信とする。]
        S3[⑤MSG種指定があった場合、受信者とビジネス文書種指定の文字列がサーバ管理情報にあるかを確認し、無ければSOAP Faultを返す。]
        S4[⑥受信企業宛てでかつビジネス文書種指定のDATAの未送信のもっとも古いものを永続記憶より取得し、対象が無ければfalseを返す]
        S5[ビジネス文書種指定がない場合、受信企業宛てのDATAの未送信のもっとも古いものを永続記憶より取得。対象が無ければ結果をfalseで返す。]
        S6[異常があったときはSOAP Faultを返す。]
        S7[⑧送信企業、受信企業のmessageldの永続記憶の状態を送信済に変更。]
        S8[⑨正常な結果true、重複はfalseを返す。messageldが不明や、異常があったときはSOAP Faultを返す]
    end

    subgraph Client [2007年度WSDL対応のJX手順クライアント]
        direction TB
        C1[③取得開始。]
        C2[④受信企業を指定し送信]
        C3[⑥受信後、messageld、他とDATAを永続記憶に格納(但しmessageldが重複する場合、破棄)、受信済みとする]
        C4[⑧送信したビジネス文書種指定と受信したビジネス文書種指定がことなる場合、エラー]
        C5[⑦messageld、送信企業、受信企業を設定しConfirm送信。]
        C6[⑩ResultがFalseでも、TrueでもAPへDATAを渡し、永続記憶の状態をAP処理済みへ変更。]
    end

    subgraph ServerManagement [サーバ管理の受信先と文書種リスト]
        SM[永続記憶]
    end

    subgraph ServerRequest [取得要求]
        SR[GetDocument]
    end

    subgraph ServerResponse [取得の通知]
        SRN[ConfirmDocument]
    end

    subgraph ClientRequest [取得の通知]
        CRN[GetDocument]
    end

    subgraph ClientResponse [取得通知確認]
        CCRN[ConfirmDocument]
    end

    S1 --> S2
    S2 --> S3
    S3 --> S4
    S4 --> S5
    S5 --> S6
    S6 --> S7
    S7 --> S8
    S8 --> S1

    C1 --> C2
    C2 --> C3
    C3 --> C4
    C4 --> C5
    C5 --> C6
    C6 --> C1

    SM --> S4
    SM --> S5
    SM --> C3
    SM --> C6

    SR --> C1
    C2 --> SR
    C3 --> SRN
    SRN --> CCRN
    CCRN --> C6
    C6 --> SM
  
```

2007年度版WSDL対応のJX手順サーバ

①登録開始。messageld作成。

②DATAとmessageld、他を格納。状態を未送信とする。

⑤MSG種指定があった場合、受信者とビジネス文書種指定の文字列がサーバ管理情報にあるかを確認し、無ければSOAP Faultを返す。

⑥受信企業宛てでかつビジネス文書種指定のDATAの未送信のもっとも古いものを永続記憶より取得し、対象が無ければfalseを返す

ビジネス文書種指定がない場合、受信企業宛てのDATAの未送信のもっとも古いものを永続記憶より取得。対象が無ければ結果をfalseで返す。

異常があったときはSOAP Faultを返す。

⑧送信企業、受信企業のmessageldの永続記憶の状態を送信済に変更。

⑨正常な結果true、重複はfalseを返す。messageldが不明や、異常があったときはSOAP Faultを返す

※PutDocumentは同様のため省略

※OptionalFormatType / OptionalDocumentType: 種別指定

クライアント管理の送信先と文書種リスト

③取得開始。

④受信企業を指定し送信

⑥受信後、messageld、他とDATAを永続記憶に格納(但しmessageldが重複する場合、破棄)、受信済みとする

⑧送信したビジネス文書種指定と受信したビジネス文書種指定がことなる場合、エラー

⑦messageld、送信企業、受信企業を設定しConfirm送信。

⑩ResultがFalseでも、TrueでもAPへDATAを渡し、永続記憶の状態をAP処理済みへ変更。

取得要求

GetDocument

1件取得

取得の通知

ConfirmDocument

取得通知確認

HTTPリクエスト

HTTPレスポンス

永続記憶

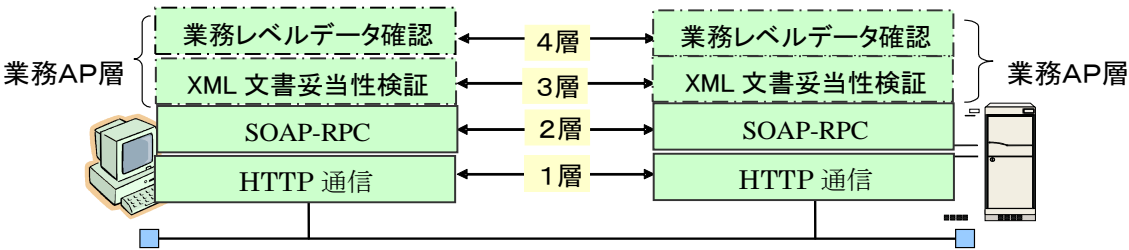
永続記憶

2007年度WSDL対応のJX手順クライアント

※ APへはData、送信者、メッセージ種を返す(推奨)

### 2.1.3 エラー通知

エラーの発生する状況は以下の4層に分けて考えることができる。



エラー発生階層	エラー検出のタイミング
1 層：HTTP 通信 レベル	HTTP プロトコルレベルでのメッセージ交換時
2 層：SOAP-RPC レベル	SOAP Envelope 内解析時
3 層：XML 文書妥当性検証	受信した XML ビジネス文書のスキーマ定義ファイルによる妥当性検証時（字句チェックエラー）
4 層：業務レベルデータ確認	受信側の業務 AP で XML ビジネス文書を処理中

図表 22 階層別エラー状況

#### （1） HTTP 通信レベルのエラー

HTTP では、サーバからの応答として 3 桁の数字によるステータスコードが返される。この値が 300 台、400 台、500 台の場合がエラーである。

図表 23 に、代表的な HTTP エラーを挙げる。

HTTP ステータスコード	説明
401	認証に失敗した。
404	接続先の URL に誤りがある。
500	サーバで何らかのエラーが発生した。
503	相手先のサーバが一時的に利用できない。

図表 23 代表的な HTTP ステータスコード

他の HTTP ステータスコードに関しては、HTTP の仕様（RFC2616）を参照されたい。



## (2) SOAP レベルのエラー

JX 手順では、SOAP におけるエラー通知手段である SOAP Fault を使用する。つまり、リクエスト処理中に SOAP エラーが発生した場合、サーバは HTTP レスポンス 500 "Internal Server Error"を発行すると同時に、そのレスポンスは、Body 要素に SOAP 処理エラーを示す Fault 要素を持つ SOAP メッセージを含まなければならない。

Fault 要素は SOAP 本体中に一度しか記述することができない。また、Fault 要素も、Envelope 要素、Header 要素、Body 要素と同じ名前空間に属するため、名前空間接頭辞 「soapenv」を用いて修飾する。

Fault 要素の記述ルールは次のとおりである。

- Fault 要素は、Body 要素中に 2 回以上現れてはいけない。
- Fault 要素は以下の子要素から構成される。

Fault 要素	用途・用法
faultcode (必須要素)	エラー内容をコード (SOAP フォールトコード値) で示す。
faultstring (必須要素)	エラー内容を説明する記述。エラーの性質についての何らかの説明が必要。
faultactor	エラーを検出したアプリケーションを示す情報を提供する。この要素は違反の発生元 URI が示される。
detail	Body 要素に関するアプリケーション固有のエラー情報を伝える。この要素は Body 要素の内容処理が正常終了しなかった場合には存在しなくてはならない。

図表 24 Fault 要素の子要素

Fault 要素の XML 文書記述例を次に示す。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>エラーの種類 (例、server, client, mustUnderstand, ...) </faultcode>
      <faultstring>エラーの内容を表す文字列</faultstring>
      <faultactor>誰がエラーを検出したか (例、URL) </faultactor>
      <detail>エラーの詳細情報</detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

また、仕様が定義されている違反を記述するときには、下記の faultcode の値が faultcode 要素の中で使われなければならない。

faultcode の値	説明
VersionMismatch	SOAP Envelope 要素に対して間違った名前空間を検出
MustUnderstand	値"1"の mustUnderstand 属性を含んだ SOAP Header 要素の子要素があり、それが理解出来ないか、意図したように処理されていない。
Client	メッセージが正しく構成されていないか、処理を進める上で適切な情報を含んでいないことを示す。
Server	直接メッセージの内容に起因する原因ではなくメッセージの処理過程に関わる理由でメッセージを処理することが出来なかったことを示す。

図表 25 SOAP エラー通知(faultcode)

尚、SOAP エラー通知の詳細については、SOAP 1.1 の仕様を参照されたい。

### (3) XML 文書妥当性検証のエラー／業務レベルデータ確認のエラー

JX 手順では、SOAP Fault 要素中の detail 要素の子要素として、exceptionInfo 要素を追加することにより業務アプリケーションレベルで検出したエラーに関する情報を交換できる。

尚、業務アプリケーションレベルのエラー情報を交換するには、サーバ側で PutDocument 受信後、業務アプリケーションを実行し、その結果（正常時：PutDocumentResponse、異常時：SOAP Fault）を返す必要がある。

```
<exceptionInfo xmlns="http://www.dsri-dcc.jp/ retailCollaboration/2001/11/b2b">
  <exceptionPhase>LEXICAL_CHECK</errorPhase>
  <errorInfo id="error0001">
    <errorLevel>ERROR_FATAL</errorLevel>
    <errorCode>ERROR_DATA_ALREADY_EXIST</errorCode>
    <errorDescription>既にデータが存在します</errorDescription>
    <errorLocation>
      xpointer(/ProductRegistration/MessageInfo/Sender/PartyId)
    </errorLocation>
  </errorInfo>
  <errorInfo id="error0002">
    ...
  </errorInfo>
</exceptionInfo>
```

上記の XML データ構造について説明する。

- 業務アプリケーションレベルで検出されたエラーの詳細情報を格納するデータ構造のルート要素のタグ名称は exceptionInfo である。
- exceptionInfo は 1 つの exceptionPhase 要素と 1 つ以上の errorInfo 要素で構成される。
- exceptionPhase 要素にはエラーの発生したタイミングを記述する。exceptionPhase に設定可能な文字列の例を次に示す。

exceptionPhase 要素	説明
LEXICAL_CHECK	字句チェックにおけるエラー発生
MEANING_CHECK	意味チェックにおけるエラー発生

図表 26 exceptionPhase 要素に設定可能な文字列

- errorInfo 要素は errorLevel 要素、errorCode 要素、errorDescription 要素、errorLocation 要素、および id 属性から構成される。
- id 属性は exceptionInfo コンテンツ内で一意の文字列を指定する。
- errorLevel 要素にはエラーの重大性を記述する。

errorLevel 要素に設定する文字列例は次のとおりである。

errorLevel 要素値	説明
WARNING	Validation 続行、処理続行
ERROR_NORMAL	Validation 続行、処理中止
ERROR_FATAL	Validation 中止、処理中止

図表 27 errorLevel 要素に設定可能な文字列

errorCode 要素にはエラーの内容を表すコードを記述する。

errorCode 要素に設定する文字列例を次に示す。

errorCode 要素	説明
ERROR_LEXICAL_TOKEN	妥当性(validation)検証でエラーを検出
ERROR_OUT_OF_BOUNDS	範囲外データ検出
ERROR_OUT_OF_SCOPES	データが複数存在する
ERROR_INVALID_DATA	無効データ検出
ERROR_EXPIRED_DATA	既に有効でないデータ検出
ERROR_DATA_ALREADY_EXIST	既にデータが存在
ERROR_OBJECT_NOT_FOUND	内部に必要なデータが見つからない
ERROR_OBJECT_NOT_FOUND_IN_PAYLOAD	ペイロードに必要なデータが見つからない
ERROR_INCONSISTENCY	内部に論理的な不整合が発生
ERROR_INCONSISTENCY_IN_PAYLOAD	ペイロードに論理的な不整合が発生
ERROR_ILLEGAL_CLASS	型の不整合
ERROR_UNSUPPORTED_OPERATION	サポートされていない要求
ERROR_CIRCULAR_REFERENCE	循環参照
ERROR_ILLEGAL_STATE	状態不整合(他のエラーコードに属さないエラー)

図表 28 errorCode 要素に設定可能な文字列

- errorDescription 要素にはエラーの内容を文字列で記述する。空要素は不可。できるだけ多くの情報を記述す

る。

- **errorLocation** 要素はエラーの発生箇所を XPointer (XPath) 形式で記述する。

**exceptionInfo** 要素は、SOAP の **fault** 要素中の **detail** 要素の子要素として利用することができる。

JX 手順における、WSDL の例を次に示す。

- **wsdl:message** 要素で **errorInfo** スキーマを指定する
- **wsdl:portType** 要素の子要素として、**input** 要素、**output** 要素に続いて、**fault** 要素を定義し、**wsdl:message** 要素で定義されたメッセージを指定する
- **wsdl:binding** 要素の子要素として、**input** 要素、**output** 要素に続いて、**fault** 要素を定義する。

```
<definitions .... >
  ...
  <wsdl:message name=" exceptionInfo ">
    <wsdl:part name="fault" type="s0:exceptionResponse"/>
  </wsdl:message>

  <wsdl:portType name="JXMSTransferSOAP">
    <wsdl:operation name="GetDocument">
      <wsdl:input message="s0:GetDocumentSoapIn" />
      <wsdl:output message=" s0:GetDocumentSoapOut" />
      <wsdl:fault message=" s0:exceptionInfo"/>
    ...
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding .... >
    <wsdl:operation .... >
      <wsdl:input>..</wsdl:input>
      <wsdl:output>..</wsdl:output>
      <wsdl:fault>
        <soap:fault use="literal " />
        <soap:header message=" s0:GetDocumentMessageHeader"
          part="MessageHeader" use="literal " />
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
</definitions>
```

## 2.1.4 セキュリティ仕様

### 2.1.4.1 JX 手順におけるセキュリティ技術

JX 手順のセキュリティ技術には、HTTP ベーシック認証、SSL/TLS、電子署名があり、それぞれの技術について以下に示す。

#### (1) HTTP ベーシック認証

HTTP ベーシック認証 (ID とパスワード) を用いたクライアントのアクセス制御を行う。

#### (2) SSL/TLS

本ガイドラインでは、通信プロトコルレベルのセキュリティ確保に関して以下を推奨する。

- SSL/TLS による暗号化通信
- SSL/TLS サーバ認証による、サービスプロバイダの成りすまし防止

#### (3) 電子署名

JX 手順の電子署名技術としては、WS-Security の署名(Signature)部分を利用することが可能である。

WS-Security は大きく次のような構成となっている。



図表 29 WS-Security の構成

電子署名部分は<ds:Signature>要素を利用し、メッセージの完全性を実現する。これは、XML-Signature 標準に対応している。

使用例として WS-Security 2004 仕様書に記載されている例を次に示す。

```
<?xml version="1.0" encoding="utf-8"?>
<S11:Envelope xmlns:S11="..." xmlns:wsse="..." xmlns:wsu="..." xmlns:ds="...">
  <S11:Header>
    <wsse:Security>
      <wsse:BinarySecurityToken ValueType="...#X509v3"
        EncodingType="...#Base64Binary" wsu:Id="X509Token">
        MIIeZzCCA9CgAwIBAgIQEmtJZc0rqrKh5i...
      </wsse:BinarySecurityToken>
      <ds:Signature>
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
          <ds:Reference URI="#myBody">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>EULddytSo1...</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>
          BL8jdfToEb1I/vXcMZNNjPOV...
        </ds:SignatureValue>
        <ds:KeyInfo>
          <wsse:SecurityTokenReference>
            <wsse:Reference URI="#X509Token" />
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>
      </ds:Signature>
    </wsse:Security>
  </S11:Header>
  <S11:Body wsu:Id="myBody">
    <tru:StockSymbol xmlns:tru="http://www.fabrikam123.com/payloads">
      QQQ
    </tru:StockSymbol>
  </S11:Body>
</S11:Envelope>
```

尚、WS-Security の詳細については、次の URL(OASIS の Web サイト)で公開されている仕様を参照されたい。

仕様公開 URL : <https://www.oasis-open.org/standards#wssv1.0>

#### 2.1.4.2 セキュリティ要件とセキュリティ技術の対応

セキュリティ要件を満たすためのセキュリティ技術の組み合わせを示す。

セキュリティ要件	セキュリティ技術		
	HTTP ベーシック認証	SSL/TLS	電子署名
機密性	×	○	×
完全性	×	×	○
認証	△ (ID とパスワード)	○	○ (発信者の認証のみ)
否認防止	×	×	○ (受信したメッセージの保存が必要)

図表 30 セキュリティ要件を満たす技術

尚、ビジネス文書の暗号化は、業務アプリケーション(レベル)で実施する。

## 2.1.5 メッセージサンプル

(1) ビジネス文書送信要求 (クライアント→サーバ)

POST /SOAP-RPC HTTP/1.1

Host: edi.seller.co.jp

Content-Length: 2410

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

SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/PutDocument"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <From>cliuri.co.jp</From>
      <To>svruri.co.jp</To>
      <MessageId>20040313123814@cliuri.co.jp</MessageId>
      <Timestamp>2004-03-13T12:38:14</Timestamp>
    </MessageHeader>
  </soap:Header>
  <soap:Body>
    <PutDocument xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <messageId>20040313123814@cliuri.co.jp</messageId>

      <data>PD94bWwgdMvyc2lvbj0iMS4wliBlbmNvZGluZz0iU2hpZnRfSkltIj8+DQo8IS0tIFRlc3QgUGF5..bG9hZCAtL
T4NCjxUZjXN0Um9vdD4NCgk8VGZvdEVsZW1lbnQ+DQoJCTxUZjXN0MSB0ZXN0PSlxli8+..DQoJCTxUZjXN0MiB0Z
XN0PSlyli8+DQoJCTxUZjXN0MyB0ZXN0PSlzlj4NCgkJCTxUZjXN0Q2hpbGQg..Y2hpbGQ9ImNoaWxkMy-
lvPg0KCQk8L1Rlc3QzPg0KCQk8VGZvdDQgdGVzdD0iNCivPg0KCTwvVGZz..dEVsZW1lbnQ+DQo8L1Rlc3RSb290
Pg==</data>
      <senderId>4912345000019</senderId>
      <receiverId>4969951110016</receiverId>
      <formatType>SecondGenEDI</formatType>
      <documentType>Order</documentType>
    </PutDocument>
  </soap:Body>
</soap:Envelope>
```



# ビジネス文書送信要求のSOAPメッセージ解説（クライアント→サーバ）

要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始（ルートタグ）	
	xmlns:soap	SOAPエンベロープの名前空間（固定）	http://schemas.xmlsoap.org/soap/envelope/
	xmlns:xsi	XMLスキーマインスタンスの名前空間（固定）	http://www.w3.org/2001/XMLSchema-instance
	xmlns:xsd	XML Schemaの名前空間（固定）	http://www.w3.org/2001/XMLSchema
soap:Header		SOAPヘッダの定義開始	
	MessageHeader	メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間（固定）（*）	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
	From	メッセージ送信ホスト（クライアント）の識別子	cliuri.co.jp
	To	メッセージ受信ホスト（サーバ）の識別子	svruri.co.jp
	MessageId	メッセージの識別番号（グローバルユニーク）	20040313123814@cliuri.co.jp
	Timestamp	メッセージヘッダが作成された日時	2004-03-13T12:38:14
soap:Body		SOAPボディの定義開始	
	PutDocument	ドキュメント送信のメソッド	
	xmlns	ドキュメント送信メソッドの名前空間（固定）（*）	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
	messageId	メッセージの識別番号（=MessageId）	20040313123814@cliuri.co.jp
	data	送信ドキュメントデータ（Base64Encoding）	PD94bWwgdmVyc2lvdj0iMS4wliBlbmNvZGluZz0iU2hpZnRfSkltTj8+DQo8IS0tIFRlc3QgUGF5
	senderId	ドキュメント送信者識別子	4912345000019
	receiverId	ドキュメント受信者識別子	4569951110016
	formatType	ドキュメント形式	SecondGenEDI
	documentType	ドキュメント種別	Order
	compressType	圧縮タイプ（Reserved）	
(*) 名前空間のプレフィックスなし			

(2) ビジネス文書送信応答 (サーバー→クライアント)

HTTP/1.1 200 OK

Content-Length: 771

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

SOAPAction: http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/PutDocument

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <From>svruri.co.jp</From>
      <To>cliuri.co.jp</To>
      <MessageId>20040313123817@svruri.co.jp</MessageId>
      <Timestamp>2004-03-13T12:38:17</Timestamp>
    </MessageHeader>
  </soap:Header>
  <soap:Body>
    <PutDocumentResponse xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <PutDocumentResult>true</PutDocumentResult>
    </PutDocumentResponse>
  </soap:Body>
</soap:Envelope>
```

ビジネス文書送信応答のSOAPメッセージ解説 (サーバークライアント)										
要素						属性		説明		サンプル値
<?xml version="1.0" encoding="UTF-8"?>										
soap:Envelope								SOAPエンベロープの定義開始(ルートタグ)		
						xmlns:soap		SOAPエンベロープの名前空間(固定)		http://schemas.xmlsoap.org/soap/envelope/
						xmlns:xsi		XMLスキーマインスタンスの名前空間(固定)		http://www.w3.org/2001/XMLSchema-instance
						xmlns:xsd		XML Schemaの名前空間(固定)		http://www.w3.org/2001/XMLSchema
soap:Header								SOAPヘッダの定義開始		
MessageHeader								メッセージヘッダの定義開始		
						xmlns		メッセージヘッダの名前空間(固定) (*)		http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
From								メッセージ送信ホスト(サーバ)の識別子		svruri.co.jp
To								メッセージ受信ホスト(クライアント)の識別子		cliuri.co.jp
MessageId								メッセージの識別番号(グローバルユニーク)		20040313123817@svruri.co.jp
Timestamp								メッセージヘッダが作成された日時		2004-03-13T12:38:17
soap:Body								SOAPボディーの定義開始		
PutDocumentResponse								ドキュメント送信メソッドの応答		
						xmlns		ドキュメント送信メソッドの名前空間(固定) (*)		http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
PutDocumentResult								ドキュメント送信メソッドの戻り値		true
(*) 名前空間のプレフィックスなし										

(3) ビジネス文書受信要求 (クライアント→サーバ)

POST /SOAP-RPC HTTP/1.1

Host: edi.seller.co.jp

Content-Length: 749

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

SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/GetDocument"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<soap:Header>

<MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">

<From>cliuri.co.jp</From>

<To>svruri.co.jp</To>

<MessageId>20040313123819@cliuri.co.jp</MessageId>

<Timestamp>2004-03-13T12:38:19</Timestamp>

</MessageHeader>

</soap:Header>

<soap:Body>

<GetDocument xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">

<receiverId>4912345000019</receiverId>

</GetDocument>

</soap:Body>

</soap:Envelope>

## ビジネス文書受信要求のSOAPメッセージ解説（クライアント→サーバ）

要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始（ルートタグ）	
	xmlns:soap	SOAPエンベロープの名前空間（固定）	http://schemas.xmlsoap.org/soap/envelope/
	xmlns:xsi	XMLスキーマインスタンスの名前空間（固定）	http://www.w3.org/2001/XMLSchema-instance
	xmlns:xsd	XML Schemaの名前空間（固定）	http://www.w3.org/2001/XMLSchema
soap:Header		SOAPヘッダの定義開始	
MessageHeader		メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間（固定）（*）	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
	From	メッセージ送信ホスト（クライアント）の識別子	cliuri.co.jp
	To	メッセージ受信ホスト（サーバ）の識別子	svruri.co.jp
	MessageId	メッセージの識別番号（グローバルユニーク）	20040313123819@cliuri.co.jp
	Timestamp	メッセージヘッダが作成された日時	2004-03-13T12:38:19
soap:Body		SOAPボディーの定義開始	
GetDocument		ドキュメント受信のメソッド	
	xmlns	ドキュメント受信メソッドの名前空間（固定）（*）	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
	receiverId	ドキュメント受信者識別子	4912345000019

（\*）名前空間のプレフィックスなし

(4) ビジネス文書の形式と種別の指定のあるビジネス文書受信要求 (クライアント→サーバ)

POST /SOAP-RPC HTTP/1.1

Host: edi.seller.co.jp

Content-Length: 749

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

SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/GetDocument"

<?xml version='1.0' encoding='UTF-8'?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Header>

<MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">

<From>wsretail.info</From>

<To>retail@svruri.co.jp</To>

<MessageId>20080130184952390.0001@wsretail.info</MessageId>

<Timestamp>2008-01-30T09:49:52</Timestamp>

<OptionalFormatType> SecondGenEDI </OptionalFormatType>

<OptionalDocumentType> Order</OptionalDocumentType>

</MessageHeader>

</soapenv:Header>

<soapenv:Body>

<GetDocument xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">

<receiverId>4912345000300</receiverId>

</GetDocument>

</soapenv:Body>

</soapenv:Envelope>

ビジネス文書の形式と種別の指定のあるビジネス文書受信要求のSOAPメッセージ解説（クライアント→サーバ）

要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始（ルートタグ）	
	xmlns:soap	SOAPエンベロープの名前空間（固定）	http://schemas.xmlsoap.org/soap/envelope/
soap:Header		SOAPヘッダの定義開始	
MessageHeader		メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間（固定）（*）	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
From		メッセージ送信ホスト（クライアント）の識別子	wsretail.info
To		メッセージ受信ホスト（サーバ）の識別子	retail@svruri.co.jp
MessageId		メッセージの識別番号（グローバルユニーク）	20080130184952390.0001@wsretail.info
Timestamp		メッセージヘッダが作成された日時	2008-01-30T09:49:52
OptionalFormatType		要求するビジネス文書の形式	SecondGenEDI
OptionalDocumentType		要求するビジネス文書の種別	Order
soap:Body		SOAPボディーの定義開始	
GetDocument		ドキュメント受信のメソッド	
	xmlns	ドキュメント受信メソッドの名前空間（固定）（*）	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
receiverId		ドキュメント受信者識別子	4912345000300

（\*） 名前空間のプレフィックスなし

(5) ビジネス文書受信応答 (サーバ→クライアント)

HTTP/1.1 200 OK

Content-Length: 2478

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

SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/GetDocument"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<soap:Header>

<MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">

<From>svruri.co.jp</From>

<To>cliuri.co.jp</To>

<MessageId>20040313123819@svruri.co.jp</MessageId>

<Timestamp>2004-03-13T12:38:19</Timestamp>

</MessageHeader>

</soap:Header>

<soap:Body>

<GetDocumentResponse xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">

<GetDocumentResult>true</GetDocumentResult>

<messageId>20040313123819@svruri.co.jp</messageId>

<data>PD94bWwgdmVyc2lvdj0iMS4wliBlbmNvZGluz0iU2hpZnRfSkltIj8+DQo8IS0tIFRlc3QgUGF5..bG9hZCAtL  
T4NCjxUZjN0Um9vdD4NCgk8VGZzdEVsZW1lbnQ+DQoJCTxUZjN0MSB0ZXN0PSixli8+..DQoJCTxUZjN0MiB0Z  
XN0PSlyli8+DQoJCTxUZjN0MyB0ZXN0PSlzlj4NCgkJCTxUZjN0Q2hpbGQg..Y2hpbGQ9ImNoaWxkMy-  
lvPg0KCQk8L1Rlc3QzPg0KCQk8VGZzdDQgdGVzdD0iNCI+DQoJPC9UZjN0..RWxlbWVudD4NCjwvVGZzdFJvb3  
Q+</data>

<senderId>4969951110016</senderId>

<receiverId>4912345000019</receiverId>

<formatType>SecondGenEDI</formatType>

<documentType>Order</documentType>

</GetDocumentResponse>

</soap:Body>

</soap:Envelope>



ビジネス文書受信応答のSOAPメッセージ解説 (サーバークライアント)

要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始(ルートタグ)	
	xmlns:soap	SOAPエンベロープの名前空間(固定)	http://schemas.xmlsoap.org/soap/envelope/
	xmlns:xsi	XMLスキーマインスタンスの名前空間(固定)	http://www.w3.org/2001/XMLSchema-instance
	xmlns:xsd	XML Schemaの名前空間(固定)	http://www.w3.org/2001/XMLSchema
soap:Header			
MessageHeader		メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間(固定) (*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
From		メッセージ送信ホスト(サーバ)の識別子	svruri.co.jp
	To	メッセージ受信ホスト(クライアント)の識別子	cliuri.co.jp
	MessageId	メッセージの識別番号(グローバルユニーク)	20040313123819@svruri.co.jp
	Timestamp	メッセージヘッダが作成された日時	2004-03-13T12:38:19
soap:Body			
GetDocumentResponse		ドキュメント送信メソッドの応答	
	xmlns	ドキュメント受信メソッドの名前空間(固定)(*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
GetDocumentResult		ドキュメント受信メソッドの戻り値	true
	messageId	メッセージの識別番号(=MessageId)	20040313123819@svruri.co.jp
	data	送信ドキュメントデータ(Base64Encoding)	PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iU2hpZnRfSkITj8+DQo8IS0tIFRlc3QgUGF5
	senderId	ドキュメント送信者識別子	4969951110016
	receiverId	ドキュメント受信者識別子	4912345000019
	formatType	ドキュメント形式	SecondGenEDI
	documentType	ドキュメント種別	Order
	compressType	圧縮タイプ(Reserved)	

(\*) 名前空間のプレフィックスなし

(6) ビジネス文書受信確定要求 (クライアント→サーバ)

POST /SOAP-RPC HTTP/1.1

Host: edi.seller.co.jp

Content-Length: 865

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

SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/ConfirmDocument"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <From>cliuri.co.jp</From>
      <To>svruri.co.jp</To>
      <MessageId>20040313123826@cliuri.co.jp</MessageId>
      <Timestamp>2004-03-13T12:38:26</Timestamp>
    </MessageHeader>
  </soap:Header>
  <soap:Body>
    <ConfirmDocument xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <messageId>20040313123819@svruri.co.jp</messageId>
      <senderId>4969951110016</senderId>
      <receiverId>4912345000019</receiverId>
    </ConfirmDocument>
  </soap:Body>
</soap:Envelope>
```

ビジネス文書受信確定要求のSOAPメッセージ解説（クライアント→サーバ）									
要素				属性		説明		サンプル値	
<?xml version="1.0" encoding="UTF-8"?>									
soap:Envelope						SOAPエンベロープの定義開始（ルートタグ）			
				xmlns:soap		SOAPエンベロープの名前空間（固定）		http://schemas.xmlsoap.org/soap/envelope/	
				xmlns:xsi		XMLスキーマインスタンスの名前空間（固定）		http://www.w3.org/2001/XMLSchema-instance	
				xmlns:xsd		XML Schemaの名前空間（固定）		http://www.w3.org/2001/XMLSchema	
soap:Header						SOAPヘッダの定義開始			
MessageHeader						メッセージヘッダの定義開始			
				xmlns		メッセージヘッダの名前空間（固定）（*）		http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server	
From						メッセージ送信ホスト（クライアント）の識別子		cliuri.co.jp	
To						メッセージ受信ホスト（サーバ）の識別子		svruri.co.jp	
MessageId						メッセージの識別番号（グローバルユニーク）		20040307074036@cliuri.co.jp	
Timestamp						メッセージヘッダが作成された日時		2004-03-07T07:40:36	
soap:Body						SOAPボディーの定義開始			
ConfirmDocument						ドキュメント受信確定のメソッド名			
				xmlns		ドキュメント受信確定メソッドの名前空間（固定）（*）		http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server	
messageId						受信確定するメッセージの識別番号		20040313123826@cliuri.co.jp	
senderId						ドキュメント送信者識別子		4969951110016	
receiverId						ドキュメント受信者識別子		4912345000019	
(*) 名前空間のプレフィックスなし									

(7) ビジネス文書受信確定応答 (サーバ→クライアント)

HTTP/1.1 200 OK

Content-Length: 787

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

SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/ConfirmDocument"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <From>svruri.co.jp</From>
      <To>cliuri.co.jp</To>
      <MessageId>20040313123826@svruri.co.jp</MessageId>
      <Timestamp>2004-03-13T12:38:26</Timestamp>
    </MessageHeader>
  </soap:Header>
  <soap:Body>
    <ConfirmDocumentResponse xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <ConfirmDocumentResult>true</ConfirmDocumentResult>
    </ConfirmDocumentResponse>
  </soap:Body>
</soap:Envelope>
```

ビジネス文書受信確定応答のSOAPメッセージ解説（サーバークライアント）											
要素						属性		説明		サンプル値	
<?xml version="1.0" encoding="UTF-8"?>											
soap:Envelope								SOAPエンベロープの定義開始（ルートタグ）			
						xmlns:soap		SOAPエンベロープの名前空間（固定）		http://schemas.xmlsoap.org/soap/envelope/	
						xmlns:xsi		XMLスキーマインスタンスの名前空間（固定）		http://www.w3.org/2001/XMLSchema-instance	
						xmlns:xsd		XML Schemaの名前空間（固定）		http://www.w3.org/2001/XMLSchema	
soap:Header								SOAPヘッダの定義開始			
MessageHeader								メッセージヘッダの定義開始			
						xmlns		メッセージヘッダの名前空間（固定）（*）		http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server	
From								メッセージ送信ホスト（サーバ）の識別子		svruri.co.jp	
To								メッセージ受信ホスト（クライアント）の識別子		cliuri.co.jp	
MessageId								メッセージの識別番号（グローバルユニーク）		20040313123826@svruri.co.jp	
Timestamp								メッセージヘッダが作成された日時		2004-03-13T12:38:26	
soap:Body								SOAPボディーの定義開始			
ConfirmDocumentResponse								ドキュメント受信確定メソッドの応答			
						xmlns		ドキュメント受信確定メソッドの名前空間（固定）（*）		http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server	
ConfirmDocumentResult								ドキュメント受信確定メソッドの戻り値		true	
（*） 名前空間のプレフィックスなし											

## 2.2 推奨通信プロトコルパラメータセット

電子商取引を実施するに当たり、通信プロトコル（通信手順）は接続の第一歩である。尚、電子商取引を実施するに当たっては、別途商取引上の取り決めをする必要がある。

今回、本ガイドラインは通信手順に関する設定パラメータを出来るだけ固定値化することによって、相対で通信パラメータを試行錯誤しながらセットする手間を極力少なくすることを目的に作成した。

次章の「2.2.1 EDI 通信パラメータ参考情報（JX 手順用）」に、通信するために必要な設定をまとめた。

## 2.2.1 EDI 通信パラメータ参考情報（JX 手順用）

JiEDIA 推奨パラメータ値を記載した参考情報である。各参考情報のパラメータ設定値欄に記載されている値は、調整欄が◎の場合はサンプルであり、調整欄が空欄の場合は、推奨値となっている。また、備考欄に設定値の形式等の情報が記載されている。

1 基本情報		パラメータ設定値	↓ 当事者間で調整が必要な項目	
			調整	備考
1 通信プロトコル情報ID		1234567890123-3210987654321-001-ptc	◎	"サーバ側識別情報" - "クライアント側識別情報" - 連番3桁 - ptc
2 サーバ側情報	1 企業(サイト)の識別情報	1234567890123	◎	senderID/receiverIDに設定する情報
	2 マシンの識別情報	____@____.co.jp	◎	To/Fromに設定する情報
3 クライアント側情報	1 企業(サイト)の識別情報	3210987654321	◎	senderID/receiverIDに設定する情報
	2 マシンの識別情報	____@____.co.jp	◎	To/Fromに設定する情報

2 トランスポート層情報 (Transport)				
1 通信プロトコル情報		HTTP1.1		HTTPなどのトランスポートプロトコル名、バージョン
2 セキュリティプロトコル情報	1 プロトコル名・バージョン	TLS1.0~TLS1.2 (JiEDIA推奨 TLS1.2)	◎	TLSなどのセキュリティプロトコル名、バージョン
	2 サーバ認証	あり		サーバ認証の有無
	3 クライアント認証	あり／なし	◎	クライアント認証の有無
	4 ページック認証情報	あり／なし	◎	ありの場合、ID/パスワードを別途通知
3 フォーマットタイプ		異界毎に指定 (JiEDIA推奨 文字型64桁)	◎	SOAPメッセージのformatTypeタグの値
4 ドキュメントタイプ		異界毎に指定 (JiEDIA推奨 文字型64桁)	◎	SOAPメッセージのdocumentTypeタグの値
5 エンドポイントURI	1 URI	https://____.co.jp:443/____	◎	サーバのアクセスURI
	2 IPアドレス(オプション)	xxx.xxx.xxx.xxx:PortNo	◎	クライアント環境のIPアドレス (サーバ側でアクセス制限している場合)
	3 サブネットマスク(オプション)	255.255.255.252	◎	接続可能性のあるクライアントのアドレス空間 (ネットマスク)

3 EDI関連通信仕様情報 (DeliveryChannel)				
1 同期/非同期応答モード		同期応答		同期モードで動作
2 応答要求		あり		受領確認を要求するか
3 応答への署名		なし		受領確認メッセージに署名するか
4 重複検出		あり		同じメッセージIDを持つメッセージを登録できない

4 ビジネスメッセージ特性 (BusinessTransactionCharacteristics)				
1 圧縮	1 メッセージの圧縮	あり／なし	◎	メッセージ種別ごとに、圧縮あり／なしを調整
	2 圧縮形式	ZIP		ZIPを推奨

5 電子証明書情報				
1 証明書の種類	1 サーバ側	SSL/TLSサーバ証明書		
	2 クライアント側	SSL/TLSクライアント証明書		

6 通信可能なデータサイズ				
1 1通信あたりの最大データサイズ		異界毎に指定 (JiEDIA推奨 10MB)	◎	

◎ 相対で要調整

## ◆【参考資料】 JX 手順のエラー情報リスト

本リストはJX手順で発生・検出するエラー情報を整理した一覧です。  
 JX手順は、エラーに関する規定が詳細に定められていないため、通信における問題発生時に、接続先同士が互いにエラーを解決することが困難な状況になる場合があります。  
 本リストを利用して、導入したJX手順製品のエラー情報を整理・対応付けすることによって、エラー原因の特定や接続先との現象確認に活用してください。

サーバで発生・検出するエラー情報	エラーの説明	エラー 確認手段	エラー出力結果：ログメッセージの内容や画面表示の内容
<b>接続フェーズのエラー</b>			
コネクション切断（認証完了前）	接続要求に対する認証処理中に、サーバ以外の要因によりコネクションが切断された。 ・クライアントによるサーバ認証エラー ・接続経路のネットワーク機器障害 ・クライアント側の環境問題 など		
HTTPヘッダエラー	HTTPリクエストメッセージに不正がある		
<b>Basic認証エラー</b>			
認証情報（Authorization行）なし	HTTPリクエストメッセージにBasic認証情報のヘッダ行が存在しない		
該当IDなし	指定されたIDがサーバ環境に登録していない		
パスワード不一致	指定されたパスワードが登録情報と一致しない		
<b>TLSクライアント認証エラー</b>			
	クライアント証明書の検証に失敗した		
	サーバ環境に中間証明書およびルート証明書が登録されていない		
	証明書の検証エラー		
	証明書の期限切れ		
	証明書情報が登録されていない		
	その他		
<b>メッセージ処理中のエラー</b>			
コネクション切断（認証完了後）	接続要求を受けて通信が終了する前に、サーバソフトウェア以外の要因によりコネクションが切断された。 接続経路のネットワーク機器障害 クライアント側の環境問題 など		
<b>SOAPメッセージエラー</b>			
SOAP Action（メソッド）不正	メソッドが以下に該当していない PutDocument、GetDocument、ConfirmDocument		
SOAPメッセージ構造不正	WSDLに定義されているメッセージ構造に違反している		
SOAPメッセージ項目不正	WSDLに定義されているメッセージ項目に違反している		
From及びToの値不一致	SOAPヘッダーに指定されたクライアント情報が一致しない		
SenderId及びReceiverId	SOAPボディに指定されたクライアント情報が一致しない		
送信メッセージ重複	PutDocument要素において、同一MessageIdが既に受信済みのため、受信要求を拒否する		
メッセージ種の不一致	DocumentType（及びFormatType）の値が該当接続先の通信情報（メッセージ種）として登録されていない		
受信待ちメッセージなし	GetDocument要素に対して、未受信メッセージが1件もなかった		
受信完了時のメッセージID不一致	ConfirmDocument要素で指定されたMessageIdに該当するものがなかった		
<b>サーバ内部のエラー</b>			
	リソース等起因する問題		
	ディスクやDBなど環境に依存する問題		
	製品の稼働設定に依存する問題		
	製品固有機能による制限事項		
	その他		



---

## 改訂の要約

- ・ V1.0.0（2020 年 1 月発行）作成

インターネットを利用した  
通信プロトコル利用ガイドライン  
JX 手順版

---

2020 年 1 月 発行

インターネット EDI 普及推進協議会  
Japan internet EDI Association (JiEDIA)

本資料に関する問い合わせは、下記までお願いします。

JiEDIA 事務局：一般社団法人 情報サービス産業協会  
<https://www.jisa.or.jp/tabid/2821/Default.aspx>

〒101-0047

東京都千代田区内神田 2-3-4

S-GATE 大手町北 6F

TEL：03-5289-7651（代表）

FAX：03-5289-7653