資　料

ＣＩＩシンタックスルール
1.１１及び1.５１

平成５年３月

(財)日本情報処理開発協会

産業情報化推進センター

この資料は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて、平成4年度に実施した「ビジネスプロトコル等に関する調査研究」の一環としてとりまとめたものであります。

はじめに

昭和６２年から、ＥＤＩの業界標準化作業を開始した㈳日本電子機械工業会（ＥＩＡＪ）は、㈶日本情報処理開発協会　産業情報化推進センター（ＣＩＩ）と共同で、昭和６３年に新しいＥＤＩ標準の開発を行った。このＥＤＩ標準では、流通業界に比べてデータ項目の多い電子機器業界特有のメッセージ構造を考慮し、既に米国で採用されていた可変長フォーマットを採用することになった。様々な検討の結果、シンタックスルール，標準メッセージおよび標準データ項目で構成するのが合理的であるとの結論になったが、このようなフォーマットの構成方法は、米国のＡＮＳＩ　Ｘ．１２や当時審議中であったＥＤＩＦＡＣＴ（ＩＳＯ９７３５）と同一である。

シンタックスルールは新たに開発することになり、ＣＩＩが提案したデータタグ方式シンタックスルールを、ＥＩＡＪの改良提案にもとづいて、データ項目の区切りをデリミターからレングスタグに変更して設計された。このシンタックスルールは、当初から漢字データの使用や他業界への適用も考慮されていたが、当面の単一業界（電子機器業界）への適用を前提に、一部の拡張機能は保留されることになり、電子機器業界での限定使用になることから、ＥＩＡＪシンタックスルールと命名された。ＥＩＡＪでは、ＥＩＡＪシンタックスルールの使用を前提にした標準メッセージと標準データ項目の開発を進めたが、これらは、ＥＩＡＪ標準メッセージ、ＥＩＡＪ標準データ項目と呼ばれるようになった。

昭和６３年秋に、実用性を確認するトライアル（試行）が行われ、好結果を得たため、平成元年４月に、ＥＩＡＪシンタックスルール，ＥＩＡＪ標準メッセージおよびＥＩＡＪ標準データ項目をまとめて、『ＥＩＡＪ取引情報化対応標準１Ａ』として、電子機器業界のＥＤＩ標準とした。これが、ＥＩＡＪ標準である。

ＥＩＡＪ標準は、その後順調に電子機器業界への普及が進んだが、電子機器業界での成果を他業界でも注目するようになり、ＥＩＡＪ標準を使用したいという電子機器以外の業界が増加してきた。しかし、ＥＩＡＪ標準のベースになっているＥＩＡＪシンタックスルールは、当面の電子機器業界への適用を前提に、一部の機能が保留されていたため、他業界では使用できないという問題が生じてきた。保留されているために障害となった機能とは、使用可能データ項目数を２４０種以上に拡張する機能である（ＥＩＡＪ標準では、最大２３９種のデータ項目を使用できる）。

電子機器業界単独での使用であれば、２３９種のデータ項目でもメッセージの設計は不可能ではないが、例えば、製造業界全体での使用になった場合には、２３９種ではとうてい足りない状況になる。そこで、当初保留された拡張機能を復活することになったが、予想を遙に越えるＥＩＡＪ標準の普及を考慮し、既に表面化してきた不具合点も含めて、ＥＩＡＪシンタックスルールを改良することになり、その検討は平成２年度から開始された。

この検討は、ＥＩＡＪシンタックスルールの原提案者であるＣＩＩにおいて行われ、平成２年

末から平成３年にかけて、ＥＩＡＪとの調整が行われた。ＥＩＡＪシンタックスルールを改良した新しい標準は、『ＣＩＩシンタックスルール』と呼ぶことになったが、従来のＥＩＡＪ標準との互換性の確保（上方互換）について特に留意された。さらに、国際標準として制定されたＥＤＩＦＡＣＴ（ＩＳＯ９７３５）との互換性についても真剣な検討が行われたが、メッセージの内部構造も含めた完全な互換性を確保するのは難しいため、メッセージ・グループの外部構造についての互換性を確保したオプションを、当面の対策として追加している。こうして、平成３年４月１日に、『ＣＩＩシンタックスルール試作仕様１．００』がまとめられ、さらに、ＣＩＩシンタックスルールの採用を決めた建設業界等からの要望などを追加し、平成３年８月２８日に『ＣＩＩシンタックスルール試作仕様１．０２』が発表された。平成３年度～４年度にかけて、トランスレーターの開発も行われ、実用化テストも実施された。平成４年７月１日からは、一般ユーザーに対してトランスレーターの供給も行われるようになり、平成４年８月２８日付けで『ＣＩＩシンタックスルール１．１０』として公開することになった。

ＣＩＩシンタックスルールは、このような経過で誕生した、わが国の国内取引に用いるＥＤＩのための標準である。最大の特徴は、漢字の使用と効率の向上である。長期に渡って使用可能な標準であり、業界間や業際の取引にも適用可能である。今後、ＣＩＩシンタックスルールをベースにした標準メッセージの開発が各業界で行われ、わが国のＥＤＩ標準化の進行が期待される。

今後、以下に示すドキュメントを発行する予定である。

『ＣＩＩトランスレーター導入ガイドライン』

『－－ＣＩＩシンタックスルールを用いた－－メッセージ設計ガイドライン』

平成４年８月２８日

産業情報化推進センター

『ＣＩＩシンタックスルール１．１１及び１．５１』

１９９３．３．３１
（産業情報化推進センター）

ＣＩＩシンタックスルールは、我が国のＥＤＩの標準化を促進するために、新たに開発された構文規則である。このシンタックスルールは、㈳日本電子機械工業会で開発されたＥＩＡＪシンタックスルール（電子機器業界のＥＩＡＪ標準）を、他の業界にも適用できるように拡張したシンタックスルールで、ＥＩＡＪシンタックスルールに対して上方互換の機能設定になっている。また、今後予想される国際標準ＵＮ／ＥＤＩＦＡＣＴにおけるＩＳＯ９７３５の普及に対して、ＥＤＩネットワーク上での共存が図れる機能も設定されている。

本仕様は、実用仕様として公開した「ＣＩＩシンタックスルール１．１０（１９９２．８．２８）」の改訂版で、『ＣＩＩシンタックスルール１．１１』と『ＣＩＩシンタックスルール１．５１』の２種類で構成されている。『ＣＩＩシンタックスルール１．１１』は、同１．１０にＹ属性を追加した規格であり、それ以外は同一である。『ＣＩＩシンタックスルール１．５１』は同１．１１にさらに設計画像データ伝送機能を追加した規格で、バージョン１．５１がバージョン１．１１を包含する形になっている。

バージョン１．１１とバージョン１．１０は、ほとんどの場合互換性をとることができ、通常は一緒に使うことができる（詳細は、付属資料の第７項を参照）。バージョン１．５１についても、設計画像データ伝送機能を使う時には、専用のトランスレーターが必要になるが、そうでなければ、バージョン１．１０や１．１１と一緒に使うことができる。

本仕様の管理は、今後とも当センターである㈶日本情報処理開発協会　産業情報化推進センターで行う。また、著作権を含む工業所有権は当センターに帰属する。しかしながら、我が国のＥＤＩの普及と標準化推進のため、当センターは本仕様の仕様を広く公開するとともに、無償での使用を認める。同時に、本仕様の一部を変更したような紛らわしいシンタックスルールの仕様は、標準化の乱れにつながる類似規格の発生を防止するため、排除の対象になることに注意されたい。

さらに、本仕様によってベンダーで開発されたトランスレーターについては、ＣＩＩシンタックスルール準拠表示を、製品に行うことを認める。但し、粗悪品等の発生があった場合、ＣＩＩシンタックスルール準拠表示の中止を求めることがある。当センターでは、このような事態を防止するため、当センターの示す特別な手続きを実施したトランスレーターに対して、『産業情報化推進センター推奨』表示を許可することとしている。

本仕様に関する問い合わせ先を、下記に示す。

〒１０５　東京都港区芝公園３－５－８　機械振興会館内
(財)日本情報処理開発協会　産業情報化推進センター
ＴＥＬ　０３－３４３２－９３８６　　ＦＡＸ　０３－３４３２－９３８９

目　　次

Ⅰ　概　　要

『ＣＩＩシンタックスルールの概要』

1．特　　徴

ＣＩＩシンタックスルールは、我が国のＥＤＩに適合するよう使用可能文字と使用通信システムを特に考慮した体系である。以下に主な特徴を述べる。

① 製造業界など、項目数の多いメッセージのコーディングに適した、可変長フォーマットの採用

② 英数字、カタカナ、ひらがな、漢字など、我が国のＥＤＩで必要とする文字をサポート

③ 複雑な業務処理に対応できる多機能なメッセージ構造（繰返明細の９重のネスティングが可能）

④ 最小のメッセージ長を実現する効率的なメッセージ・コントロール構造（データタグ／制御タグ方式）

⑤ 我が国のＥＤＩの標準化に十分な、最大６１４３９種のデータ項目を使用できる。

⑥ 通常のＥＤＩには十分な、最大３２７６７文字（漢字の場合は、１６３８３文字）のデータ項目が取り扱える。

⑦ オプション機能の活用により、ＯＳＩ新手順（Ｆ手順およびM手順）をはじめ、全銀手順、ＪＣＡ手順など、あらゆる通信システムに対応できる（図０－１参照）。

⑧ 将来普及が予想されるＩＳＯ９７３５との並行使用のためのオプションを最初から持っている。このオプションでは、ＩＳＯ９７３５（ＵＮ／ＥＤＩＦＡＣＴ）に合致する伝送形態になり、一つのネットワーク上で両方のメッセージを取り扱うことが可能。

⑨ 現在のＥＩＡＪシンタックスルールと互換性がある。シンタックスレベルで、上方互換になっている。

⑩ 以上の特徴を十分に発揮させるトランスレーターのサポート（ＣＩＩトランスレーターは、様々なコンピュータに対して供給される）。

尚、ベンダーから実際に開発提供されるトランスレーターでは、取り扱い可能な最大データ項目長などは、上記の値よりも小さい場合があるので、ユーザーが導入する際には、注意が必要である。詳細は、『トランスレーター導入ガイドライン（産業情報化推進センター編）』を参照されたい。

2．基本構造

ＣＩＩシンタックスルールは、ファイル転送で実現するＥＤＩのために設計されており、その基本形態は、ＴＹＰＥ１２と呼ばれる一つのメッセージグループ・ヘッダー、複数のメッセージ及び一つのメッセージグループ・トレーラで構成されるファイルである（図０－２の上段

を参照）。メッセージグループ・ヘッダー、メッセージ及びメッセージグループ・トレーラは、それぞれ一つのレコードに収容される（一つのメッセージを一つのレコードに収容）。この構造を基本形式として、２つのバリエーションがある。一つは、ＴＹＰＥ１２の分割モードと称する一つのメッセージを複数の固定長レコードに収容した形式（図０－２の中段を参照）であり、もう一つは、ＴＹＰＥ－Ｅと称するＩＳＯ９７３５構造を応用した形である（図０－２の下段を参照）。以下、順に概要を述べる。

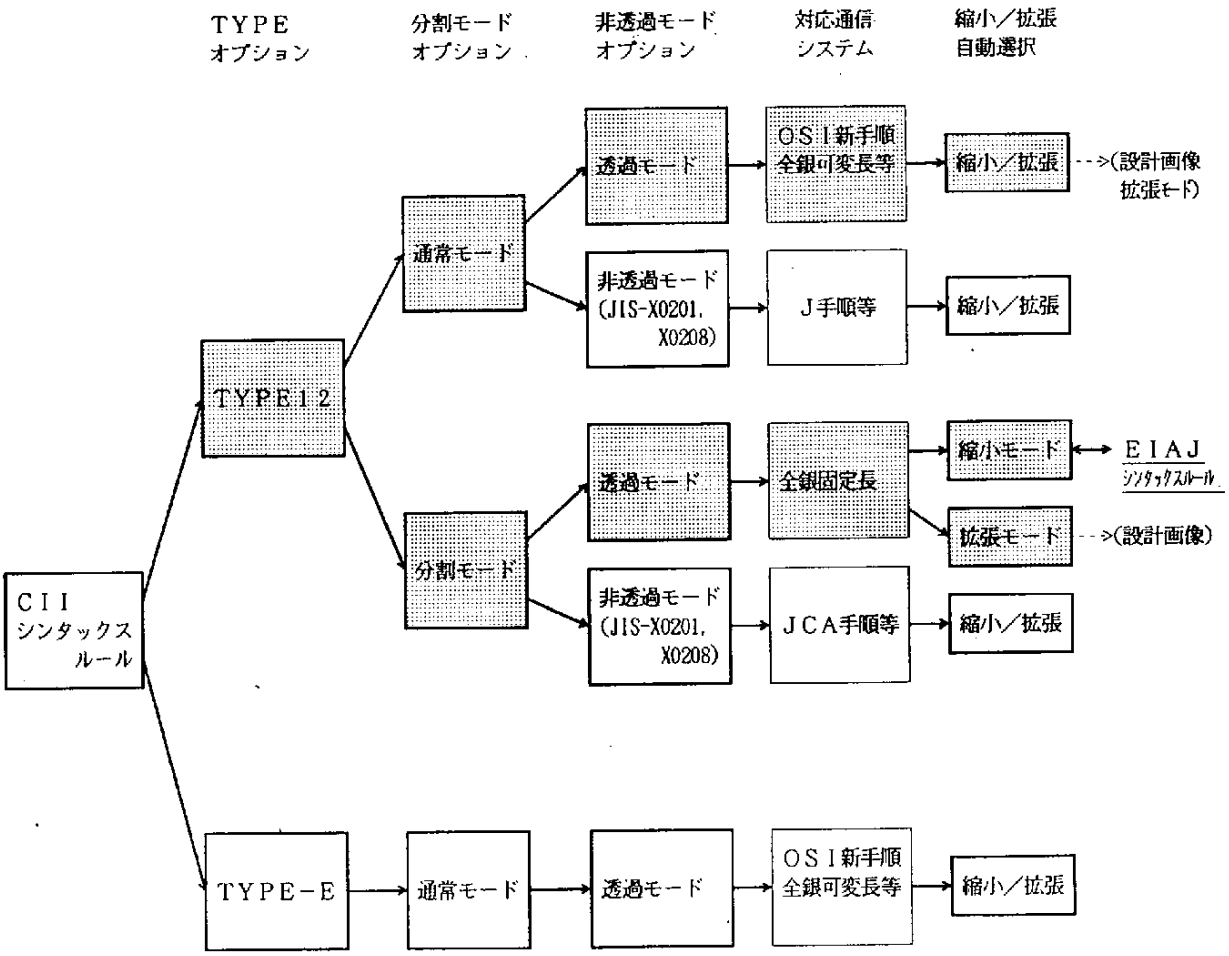


図０－１　ＣＩＩシンタックスルールのオプションと通信システムとの関係
注）網かけ内がＣＩＩトランスレーターの標準的サポート範囲

2－1　ＴＹＰＥ１２（通常モード）

図０－２の上段の構造で、ＣＩＩシンタックスルールの基本形である。その交換の階層は、図０－３で、示される。メッセージグループ・ヘッダーとメッセージグループ・トレーラは、それぞれ一つの２５１byteの固定長レコードに収容され、一つのメッセージが一つの可変長レコードに収容される。

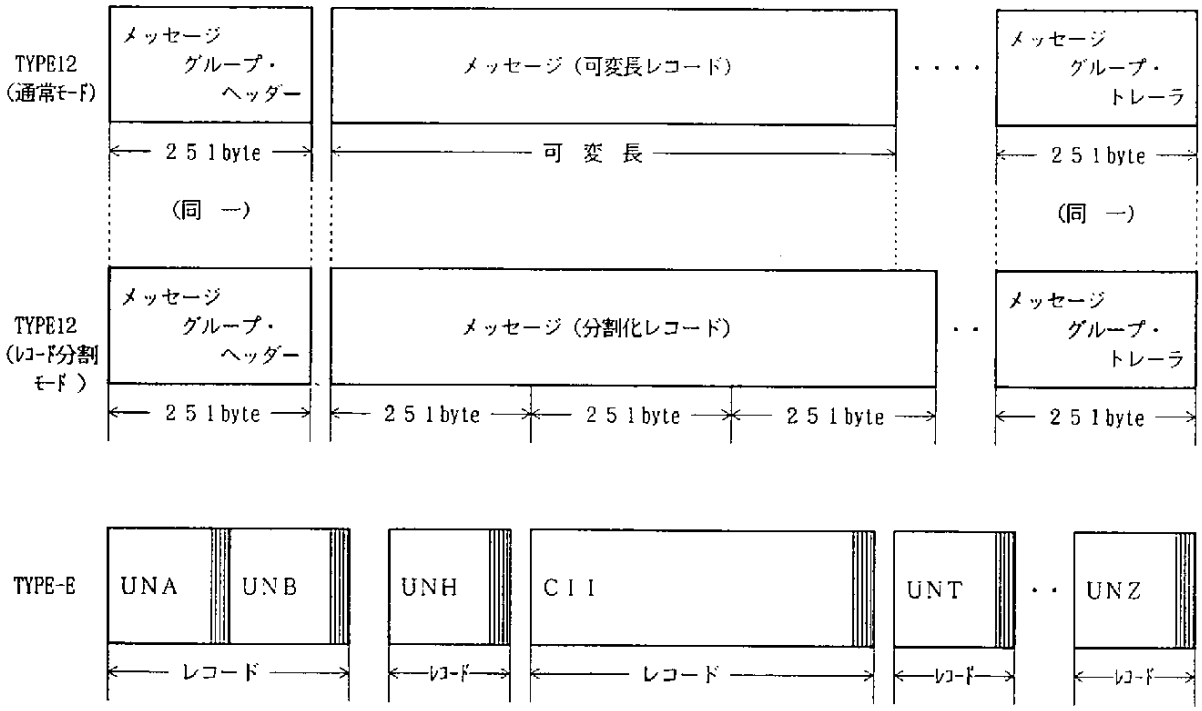


図０－２　ＴＹＰＥ１２（基本形、分割モード）及びＴＹＰＥ－Ｅの伝送形態

2－2　ＴＹＰＥ１２（分割モード）

図０－２の中段の構造であり、基本形式（通常モード）と基本的には同一で、交換の階層も通常モードと同じく図０－３で表せられる。しかし可変長レコードが取り扱えない通信システムに適合するよう、一つのメッセージは複数の２５１byteの固定長レコードに収容される。このモードの縮小モード（後述）の時は、ＥＩＡＪシンタックスルールと互換性（同一）がある。

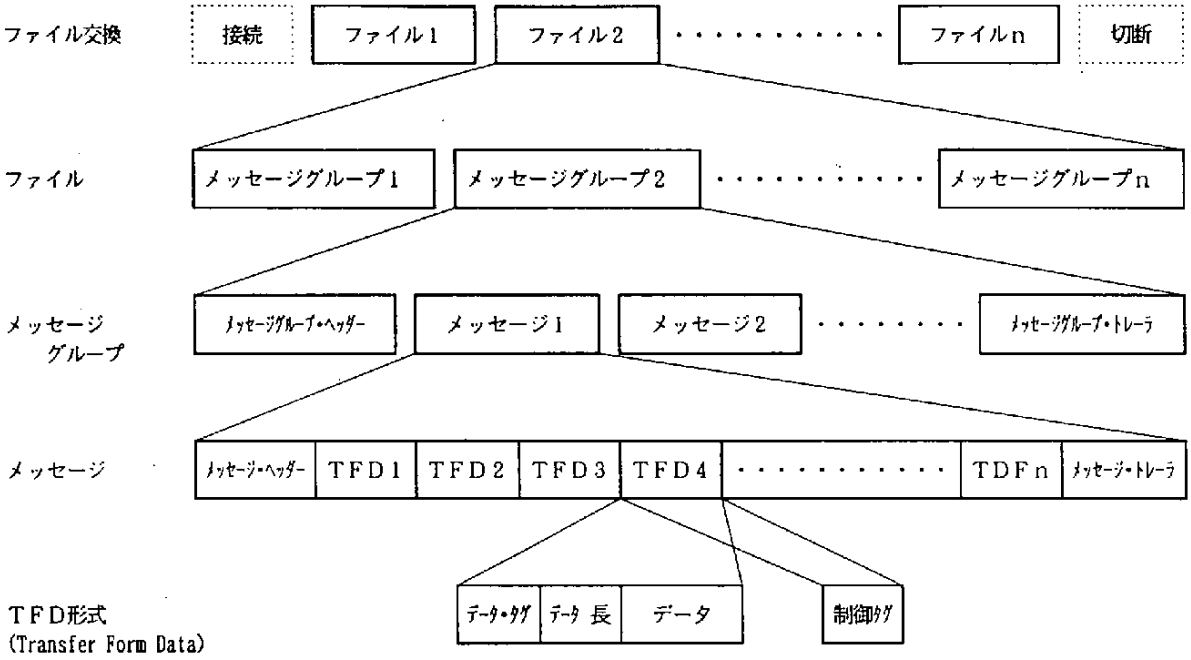


図０－３　交換の階層（ＴＹＰＥ１２）

2－3　ＴＹＰＥ－Ｅ

図０－２の下段の構造である。基本形式のメッセージグループ・ヘッダーが、ＩＳＯ９７３５のＵＮＡ及びＵＮＢセグメントに置き換えられ、メッセージグループ・トレーラが、ＵＮＺセグメントに置き換えられている。そして、メッセージは、メッセージ・ヘッダーがＵＮＨセグメントに置き換えられ、メッセージ本体はセグメント・タグ（ＣＩＩ）がセットされる以外はＴＹＰＥ１２のメッセージがそのまま用いられ、ＵＮＴセグメントが追加される（図０－４参照）。このことにより、ＵＮ／ＥＤＩＦＡＣＴ電文（ＩＳＯ９７３５でコーディングされている）と同一のネットワークでＥＤＩを実施することができる。ＴＹＰＥ－Ｅの交換の階層は、ＩＳＯ９７３５の交換の階層と同一である。ＵＮＡセグメントとＵＮＢセグメントは一つの可変長レコードに格納され、他のセグメントは、一つのセグメントが一つの可変長レコードに格納される。

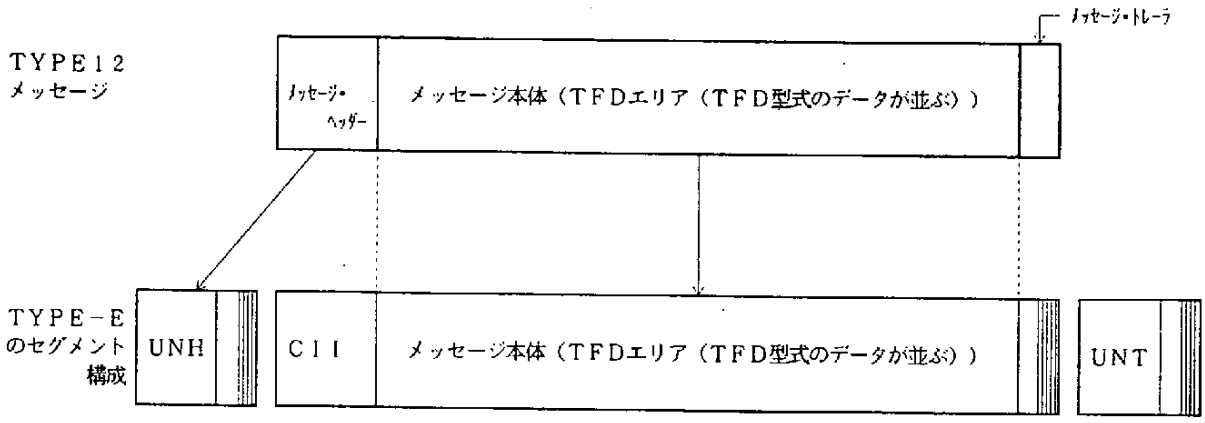


図0－4　TYPE12メッセージのセグメント化

3．縮小モードと拡張モード

ＣＩＩシンタックスルールでは、データエレメントはＴＦＤ(Transfer Form Data)形式で、伝送される。ＴＦＤは、データエレメントに、データタグ及びデータ・レングスタグ（レングスタグ）が追加されたもので、図０－５のフォーマットである。一つのメッセージは、このＴＦＤを必要数集めて構成され、例えば、１通の注文書／請求書を表す。

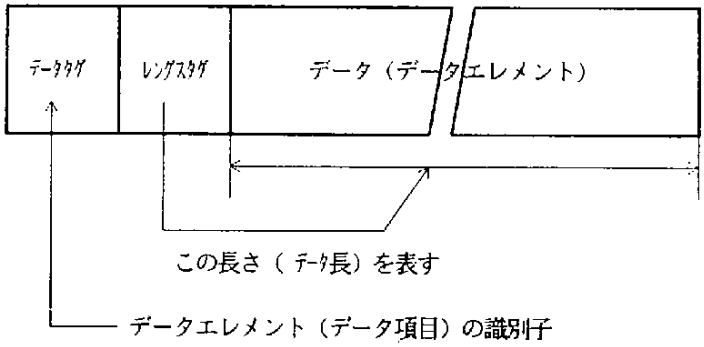


図0－5　ＴＦＤの構造

『データタグ』とは、データエレメントの意味や属性を表すＩＤであり、その値は、標準メッセージ及びデータエレメント・ディレクトリー（データ項目一覧表）上で、項目№（整理番号）として表現される。データタグには縮小形式（２３９種のデータエレメントを表示できる）と拡張形式（６１４３９種のデータエレメントを表示できる）があり、ＣＩＩシンタックスルールでは、メッセージ長を短縮するために最適な使い分けが行われる。この使い分けをコントロールする制御子を、拡張モード指示子と呼ぶ。

ＴＦＤエリア中の拡張モード指示子（Ｘ‘Ｆ０’）の左側を縮小モード領域、右側を拡張モード領域という（図０－６参照）。縮小モード領域中のＴＦＤには、縮小モードのデータタグが使用され、拡張モード領域中のＴＦＤには拡張モードのデータタグが使用される。ＴＦＤエリア中に拡張モード指示子がない時は、すべて縮小モード領域となり、ＴＦＤエリアはＥＩＡＪシンタックスルールと同一になり、互換性の確保にも活用される。

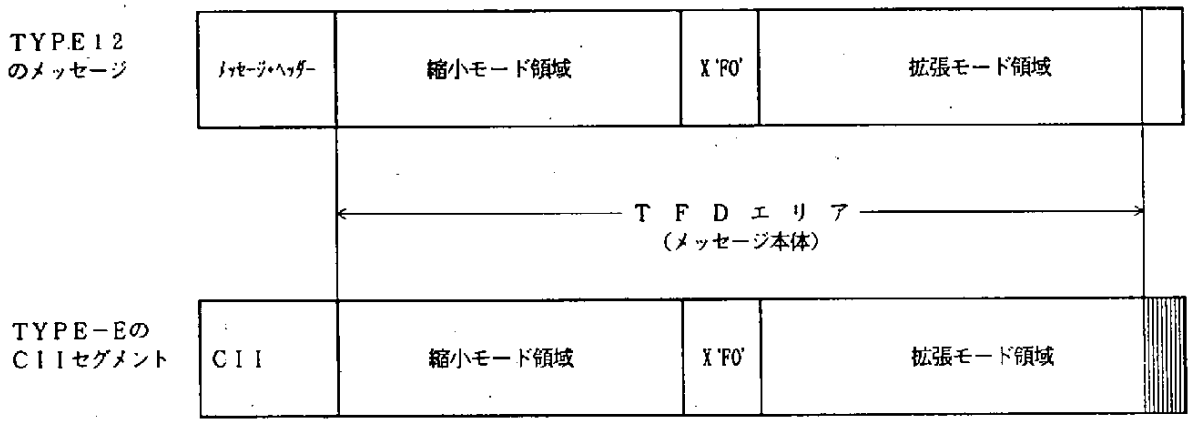


図０－６　縮小モードと拡張モード（メッセージ本体）

４．非透過モードオプション

このオプションは、Ｊ手順等の非透過モードの通信回線を使用する時に用いる。ＣＩＩシンタックスルールは、透過モードの通信回線用に設計されているため、非透過モードの通信回線を使用すると一部のデータが通信制御のキャラクターと衝突しデータ伝送ができない。非透過モードオプションは通信制御キャラクターとの衝突を避けるための対策を追加するオプションである。但し、このオプションを設定するとデータ長が長くなる。したがって、透過モードの通信回線を用いる時は、このオプションを使用しない。

５．文字コード

ＣＩＩシンタックスルールでは、ＪＩＳ－Ｘ０２０１及びＪＩＳ－Ｘ０２０８を標準の文字

コードとする。特に、メッセージグループ・ヘッダーとメッセージグループ・トレーラーでは、英数字とブランク以外の文字の使用は禁止され、文字コードもＪＩＳ－Ｘ０２０１の使用が義務づけられる。ＴＹＰＥ１２のメッセージについては、ローカル協定に基づくローカル文字の使用が可能であるが、推奨できない。ＪＩＳ－Ｘ０２０８（ＪＩＳ第１水準、第２水準漢字）は、メッセージ中のＴＦＤのデータ部（データエレメント）でのみ使用できる。

また、ＴＹＰＥ－Ｅでは、メッセージ（ＣＩＩセグメント）も含めてすべてのセグメントについて、ＪＩＳ－Ｘ０２０１，ＪＩＳ－Ｘ０２０８の文字コードを使用しなければならない。ＪＩＳ－Ｘ０２０８（ＪＩＳ第１水準、第２水準漢字）は、ＣＩＩセグメント内でのみ使用できる。

６．トランスレーター

６－１　トランスレーターのアウトライン

送信用のトランスレーターは、オプションの設定で、ＴＹＰＥ１２，ＴＹＰＥ－Ｅのどのメッセージも作成することができる。また、次の条件の内のどれか１つ（拡張モード）を検出するまで、縮小モードでメッセージを作成する。

① ２４０以上の値を持つデータ・タグの検出

② 明細番号付マルチ明細の検出

すなわち、縮小／拡張モードを自動的に設定する。一つのメッセージ中で以上の条件が成立しない時は、メッセージ全体が縮小モードで作成される。この時、ＥＩＡＪシンタックスルールと完全に互換性のあるメッセージになる（ＴＹＰＥ１２の時）。

受信用のトランスレーターは、ＴＹＰＥ１２及びＴＹＰＥ－Ｅのメッセージに加えてＥＩＡＪシンタックスルールのメッセージでも、すべて自動的に検出し、同一の変換テーブルを用いてメッセージの解釈を完全にできる。

また、トランスレーターには文字コード変換機能が含まれ、漢字のＪＩＳ第１，第２水準をカバーする。文字コードの変換は、必要に応じてバイパスできるオプションが設けられる。

トランスレーターに関しては、別途資料『トランスレーター導入ガイドライン』を参照されたい。

６－２　モデル・トランスレーター

トランスレーターは、ＥＤＩシステムにとって、最も重要な要素である。ＣＩＩシンタックスルールの大きな特徴の一つは、トランスレーターをあらかじめ考慮して設計されていることで、本シンタックスルールが期待するトランスレーターの構造や機能があらかじめ決められている。この標準トランスレーターを、『モデル・トランスレーター』と呼ぶ。

トランスレーターの仕様を標準化する場合に、最も困難な事項は、一般的なＡＰＩ（ｱﾌﾟﾘｹｰｼｮﾝ・ｲﾝﾀﾌｪｰｽ）の標準化である。ＡＰＩは、ユーザーのコンピュータ・システムの構成やＥＤＩシステムの構成に応じて様々な形態があり、通常、汎用的な仕様を定義することは難しい。そこで、業務処理システムとのデータの受渡しをすべてファイルで行い、トランスレーターへの起動／停止などの指令（コマンド）を手動で与える、実質的にＡＰＩのないトランスレーターを設定する。このトランスレーターは、いわば架空のトランスレーターであり、単なる設計ガイドであるが、ユーザー環境に応じた機能を追加することで（ＡＰＩの変更等）、実用的なトランスレーターになる。従って、モデル・トランスレーターは、様々なベンダーがＣＩＩトランスレーターを開発する時見本となるものである。

モデル・トランスレーターを参考にしてベンダーがトランスレーターを開発することで、異なるベンダーが開発したトランスレーターの組合せでも問題なくＥＤＩのデータ交換が可能になる。

モデル・トランスレーターは、コンピュータの機種によらず同一であるのが原則であるが、メンフレームに実装されるトランスレーターとパーソナル・コンピュータ（パソコン）に実装されるトランスレーターとが、同一の性能／機能を保持するというわけにはいかない。そこで、ＥＤＩ本来の目的であるデータ交換に問題が生じない範囲で、パソコン用トランスレーターでは、機能の簡略化が行われる。

従って、モデル・トランスレーターは４種存在する。すなわち、

① メンフレーム用モデル・トランスレーター

　a. データ送信用

　b. データ受信用

② パソコン用モデル・トランスレーター

　a. データ送信用

　b. データ受信用

である。図０－７は、メンフレーム用モデル・トランスレーター（送信用）の構成例である。

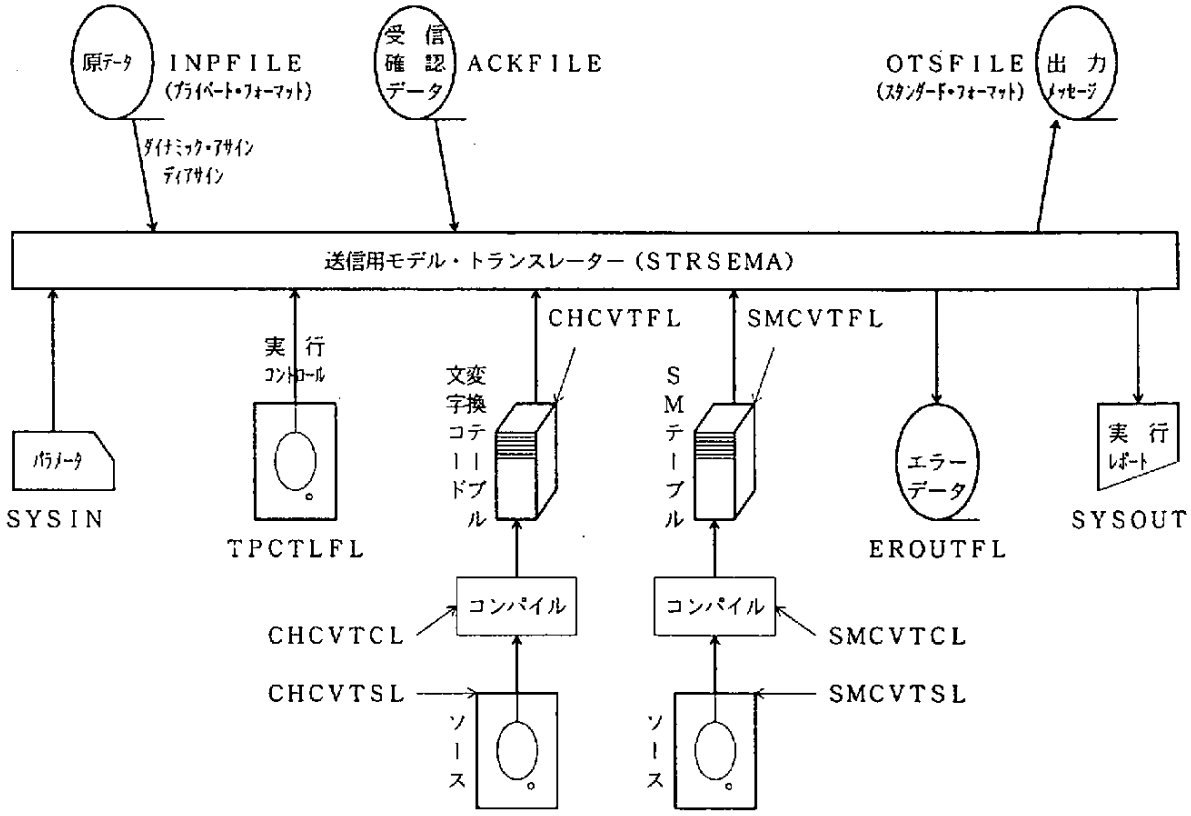


図０－７　送信用モデル・トランスレーターの構成（メインフレーム用）

７．ＣＩＩシンタックスルールによるシステムの構築

ＣＩＩシンタックスルールは、ＥＩＡＪシンタックスルールの上位互換の規格であるため、ＥＩＡＪ標準によるシステムとの共存を図るための特別な工夫は、通常、不要である。以下では、この状況について説明を行う。この説明のために、一つの仮定を行う。それは、ＥＩＡＪ標準メッセージには、当分の間、項目№２４０以上の項目と、Ｎ，Ｋ属性のデータエレメントの導入が行われないということである。もしこの仮定が成り立たない時は、ＥＩＡＪシンタックスルールそのものがＣＩＩシンタックスルールに改訂される時であるから、特別な工夫は、当然、不要になる。

図０－８の状況を設定する。ＥＩＡＪ標準を用いている業界（Ｅ業界）の他にＡ，Ｂの二つの業界がある。Ｅ業界は、早くからＥＩＡＪ標準による業界内ＥＤＩを実施している。Ａ業界が新たにＥ業界とＥＤＩを実施することになったが、そのための標準メッセージ（ＳＭ）は、項目№２４０以上の項目と、Ｎ，Ｋ属性のデータエレメントは導入しないで作成された。Ｅ業界側の企業は、既に使っているＥＩＡＪトランスレーターを使用することになったが、Ａ業界側の企業は、新規のシステムなので、ＣＩＩトランスレーターを使用することになった。

その後さらに、B業界も新たにE業界とEDIを実施することになったが、こちらのSMは、B業界の強い要望もあり、K属性（漢字）のデータエレメントを導入することになった。幸いなことに、B業界とEDIを行うE業界側の企業は比較的少数の大企業だったので、既に使用していたEIAJトランスレーターに加えてCIIトランスレーターを追加することにした。ごく一部の企業では、すべてCIIトランスレーターに変更した。B業界はCIIトランスレーターを導入した。

このようなEDIの進展が、さらに、A業界とB業界のEDIも構築させることになったが、両業界ともCIIトランスレータを導入済であり、スムースにEDIが構築された。

以上の過程は、CIIシンタックスルールの上位互換性が完全にカバーしており、ユーザー（各企業）は、シンタックスルールの違いに関する考慮は不要である。

尚、EIAJトランスレーターとの互換性等については、別途資料『トランスレーター導入ガイドライン』を参照されたい。

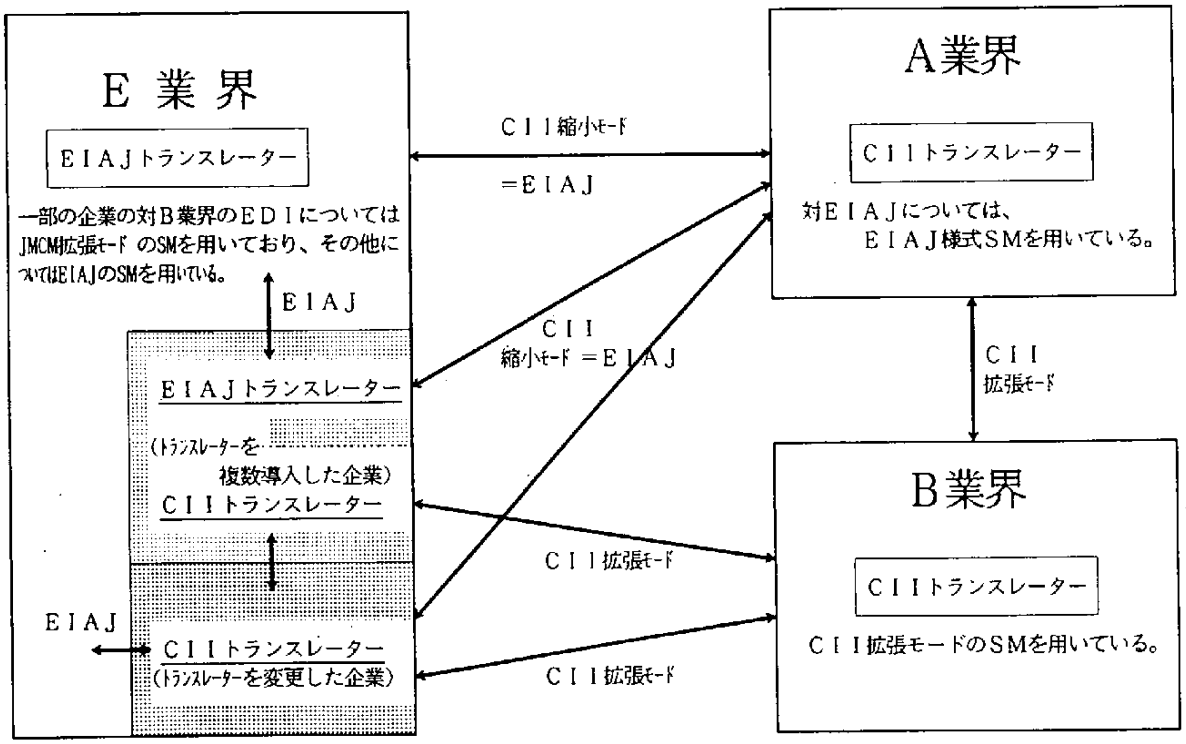


図0－8　EIAJシンタックスルールとCIIシンタックスルールの共存

8　TYPE－Eの応用

TYPE－Eは、最近増加してきたUN／EDIFACTをサポートするEDIネットワーク（海外に多い）を活用するための形式である。但し、TYPE－EはISO9735に準拠したメッセージではあるが、UN／EDIFACTの標準メッセージ（UNSM）には準拠し

ていない。ＴＹＰＥ－ＥからＵＮ／ＥＤＩＦＡＣＴのメッセージを生成するために、ＣＩＩ－ＵＮ／ＥＤＩＦＡＣＴコンバーターが必要である。当分の間、現実の業務ではそのような必要性はないと考えられるが、もし必要であれば技術的に可能であり、ＵＮＳＭの第１バージョンが確定し日本国内サブセット制定後に、ＣＩＩ（ＴＹＰＥ－Ｅ）－ＵＮ／ＥＤＩＦＡＣＴコンバーターが開発されることになろう。但し、この日本国内サブセットは、漢字項目を始めとする多くのローカル規定が含まれた、通常のＵＮＳＭサブセットとはかなり異なるものとなることが予想される。

Ⅱ　詳　　細

『ＣＩＩシンタックスルールの詳細』

1．キャラクターセットと文字コード

1－1　キャラクターセット

ＣＩＩシンタックスルールでは、8bit 文字と16bit 文字（漢字）が使用できる。表１－１は、ＣＩＩシンタックスルールで使用可能な8bit 文字の一覧表である。これ以外の文字は原則として使用できないが、ローカル協定にローカル文字の使用も不可能ではない。

16bit 文字は、別途資料『ＣＩＩシンタックスルール漢字コード表』を参照のこと。

1－2　文字コード

文字コードは、原則としてＪＩＳ－Ｘ０２０１（8bit 文字，表１－１にコードを示す）とＪＩＳ－Ｘ０２０８（16bit 文字）を用いる。しかしながら、ローカル協定により、ローカルな文字コードを用いることも不可能ではない。

表１－１　8bit文字コード表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2	SP	!	”	#	$	%	&	’	(	)	*	+	,	−	.	／
3	0	1	2	3	4	5	6	7	8	9	:	;	<	＝	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	￥	]	^	＿
6	’	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	￣	
8																
9																
A		。	「	」	、	・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
B	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
C	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	゛	゜
E																
F																

（注）コードは、Hex表示，縦軸は，上位4bit，横軸は、下位4bitである。
網掛けの部分は使用禁止である。

1－3　メッセージグループ・ヘッダーとメッセージグループ・トレーラのキャラクターセットと文字コード

メッセージグループ・ヘッダーとメッセージグループ・トレーラでは、表1－2に示す8bit文字のみ使用可能で、かつ、文字コードはＪＩＳ－Ｘ０２０１を用いなければならない。ローカル協定によるローカルコードの使用は許されない。

この処置は、オープンなＥＤＩネットワークにおいて、確実なデータ交換（行き先のコントロール）を行うために必要である。

表1－2メッセージグループ・ヘッダー及びメッセージグループ・トレーラの文字コード表（8bit文字コード）

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2	SP															
3	0	1	2	3	4	5	6	7	8	9						
4		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z					

（注）コードは、Hex表示，縦軸は，上位4bit，横軸は、下位4bit
網掛けの部分は使用禁止

2．ＣＩＩシンタックスルールの詳細（ＴＹＰＥ１２）

以下の記述では、8bit を1byteと記述し、X'1A' は、Hex 表示を表すものとする。又、誤解を避けるために、例えば、キャラクターＣはＣ(=X'43')のようにHex 表示を併用する。又、図示されたフォーマット中の値は、すべてHex 表示とする。さらに、特に断らないかぎり、バイナリーとは、符号無２進数とする。

2－1　定　　義

2－1－1　固定長データエレメント

右側のブランクの省略を行わない、データ長が常に固定されている文字列データエレメントである。ＣＩＩシンタックスルールでは、メッセージグループ・ヘッダー，メッセージグループ・トレーラ及びシステム運用電文のみで用いられ、使用可能な文字セットと文字コードは、表２－１で示される。

2－1－2　可変長データエレメント

数値データの左側のゼロあるいは文字列データの右側のブランクの省略を行うデータエレメントである。このデータエレメントは、メッセージ内のＴＦＤエリア中で、ＴＦＤ形式データ構成要素として用いられる。以下の４種がある。

(1)　数値データエレメント（９属性）

最大３０桁以下の数字列で、『０』～『９』までの数字だけで構成される数値データエレメントである。後述するＴＦＤ形式では、有効桁より左側にある『０』を省略することができる。すなわち『００１２３』と『１２３』は同一の意味とする。小数点は暗示的に示され、小数点以下の桁は、整数部に有効桁がある時は省略できない。

例えば、小数点以下が４桁である場合、『００１２３（＝０．０１２３）』は『１２３』と同一の意味になるが、『１００００』は『１』と同一の意味にはならない。本データエレメントでは8bit 文字を用いる。

(2)　数値データエレメント（N属性）

最大３０桁以下の数字列で、『０』～『９』までの数字、正負符号（『＋』及び『－』）及び小数点（『．』）で構成される数値データエレメントである。正負符号及び小数点は桁数に含めない。

正負符号は、常に数字列の先頭（左側）になければならない。『＋』符号は省略することができる。正負符号と最も左側にある有効桁との間にある『０』を省略することができる。すなわち、『－００１２３』と『－１２３』は同一の意味とする。小数点は、『．』で明示される。小数点以下の桁については、最も右側にある有効桁よりもさらに右側にある『０』を省略することができる。従って、『１２．２１００』と『１２．２１』は同一の意味とす

る。小数点が省略された時は、整数となる。『．１２３』や『－．０１２』等の表現も許される。本データエレメントでは8bit 文字を用いる。

尚、受信用トランスレーターでは、ブランクとゼロを同等に扱うケースがあるが、これは、ローカルな拡張オプションである。

(3) 8bit 文字列（X属性）

最大３２７６７文字の8bit 文字で構成される文字列データエレメントである。後述するＴＦＤ形式では、最も右側にあるブランク以外の文字よりもさらに右側にあるブランクを省略することができる。例えば、『△△ＡＢＣ△△△△』と『△△ＡＢＣ』は同一の意味である。

(4) 16bit 文字列（K属性）

最大１６３８３文字（３２７６６byte）の16bit 文字で構成される文字列データエレメントである。本文字列内に、8bit 文字を含めることはできない。本文字列では、最も右側にあるブランク（16bit 文字のブランク、以下同じ）以外の文字よりもさらに右側にあるブランクを省略することができる。

例えば、『△△構文規則△△△』と『△△構文規則』は同一の意味である。

(5) 年月日（6桁または8桁の数字列：Y属性）

西暦年月日を表す数値である。本規格は、ＣＩＩシンタックスルール１．１１をサポートしているトランスレーターで用いることが可能であるが、ＣＩＩシンタックスルール１．１０をサポートするトランスレーターでも、Y属性のかわりに9属性（変換テーブル上での定義）を用いることで、西暦１９５１～１９９９年までは、正常に動作する。２０００年以後は、ＣＩＩシンタックスルール１．１１をサポートしているトランスレーターでなければ、誤動作する可能性が大きい。

① 6桁表現と8桁表現がある。

② 6桁表現は Y(6) と表記し、１９５１年～２０５０年までの年月日を表す最大6桁のYYMMDD形式8bit 数字列である（小数点記号や正負記号を含まない）。伝送時に左側のゼロを省略することができる。

③ 8桁表現は Y(8) と表記し、０１００年～９９９９年までの年月日を表す最大8桁のYYYYMMDD形式8bit 数字列である（小数点記号や正負記号を含まない）。伝送時に左側のゼロを省略することができる。

尚、００００年～００５０年までは２０００年～２０５０年と解釈され、００５１年～００９９年までは１９５１年～１９９９年と解釈される。すなわち、００２５年＝２０２５年とされ、００９１年＝１９９１年とされる。

④　属性 Y(6) と Y(8) は、数字列であり、左側のゼロを省略することができる。すなわち、ＴＦＤ形式においては、 Y(6) 属性のデータ Y(8) 属性のデータ共に、最大長がそれぞれ６byte、８byteの可変長である。

⑤　Y(6) と Y(8) は、１９５１年～２０５０年までの間を表現する時は、完全に互換性があり、同一の形式として扱う事ができる。

(6)　補足１

標準メッセージ上では、(1)から(4)までのデータエレメントの属性と桁数の表示を、表２－１のように行う。

(7)　補足２

数値データ内でのブランクは原則として使用禁止であるが、受信用トランスレーターでは、たとえ数値の途中に存在しても、エラー扱いにせず『０』と見なす。

表２－１　ＣＩＩシンタックスルールのデータエレメントの標準メッセージ上での表現方法

データ・タイプ			標準ﾒｯｾｰｼﾞ,ﾃﾞｰﾀｴﾚﾒﾝﾄ・ﾃﾞｨﾚｸﾄﾘｰでの表現	データ例	備考
文字データ	8 bit文字列 (JIS X0201)	X属性	X（n） n:最大長（byte数）	例）X（8）の時 ABCDEFGH	長さは、byte数で表す。 ＥＩＡＪシンタックスルールと同一である。
	16 bit文字列 (JIS X0208)	K属性	K（n） n:最大長（byte数）	例）K（10）の時 産業と情報	長さは、byte数で表す。 漢字文字数の２倍になる。
数値データ	固定小数点 正　数 (JIS X0201)	9属性	9（n）， 9（n）V（m） n:整数部の桁数 m:小数部の桁数	例）9（5）の時 23456 例）9（3）V（2）の時 23456（小数点は、４と５の間）	9(5)V(0)と9(5)は同一の意味である。 ＥＩＡＪシンタックスルールと同一である。
	浮動小数点数 (JIS X0201)	N属性	N（n）， N（n）V（m） n:整数部の桁数 m:小数部の桁数	例）N（5）の時 -23456 例）N（4）V（2）の時 -2345.6	N(5)V(0)とN(5)は同一の意味である。
	日　付 (JIS X0201)	Y属性	Y（n） nは６又は８	例）Y（6）の時 YYMMDD タイプ 930331 例）Y（8）の時 YYYYMMDD タイプ 19930331	西暦日付である。

2－1－3　ＴＦＤ形式データ１（データエレメント）

メッセージ内の可変長データエレメントは、ＴＦＤ(Transfer Form Data)形式で表現しなければならない。

データタグ（データ識別ＩＤ）、データエレメント長（データ長）及び可変長データエレメントでＴＦＤ形式データを構成する（図２－１参照）。

① データタグは、メッセージ内でのデータエレメントを識別する。通常、標準メッセージ及びデータエレメント・ディレクトリーの整理番号が用いられる。

② データエレメント長は、直後のデータエレメントの長さをbyte数で表す。

③ データエレメントには、２－１－１で示した省略ルールが適用できる。

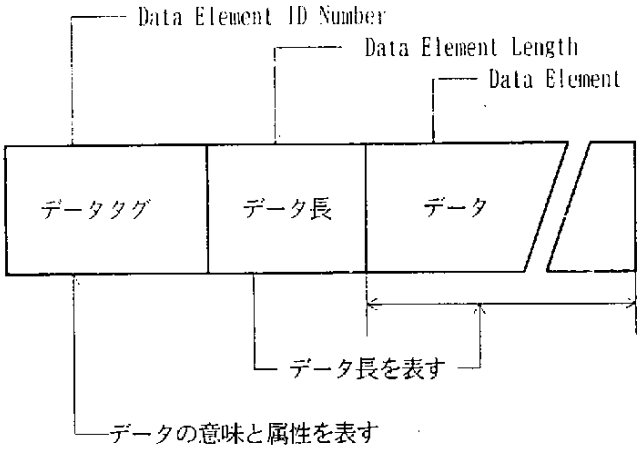


図２－１　ＴＦＤ形式データ

(1) データタグ

データタグには、縮小モードと拡張モードがある。メッセージ内には、ＴＦＤ形式のデータを格納するＴＦＤエリアがあり、ＴＦＤエリア内の拡張モード指示子の左側が縮小モード、右側が拡張モードになる。ＴＦＤエリア内に拡張モード指示子がない時は、ＴＦＤエリア全体が縮小モードとなる（図２－２を参照）。縮小モードでは、データタグは１byteの符号無バイナリーの数値となり、１～２３９の範囲の数値でデータエレメントの識別子を表す。拡張モードでは、データタグは２byteの符号無バイナリーの数値となり、１～６１４３９の範囲の数値でデータエレメントの識別子を表す。この数値（データタグの値）には、通常、標準メッセージやデータエレメント・ディレクトリーの整理番号（項目No.）を用いる。

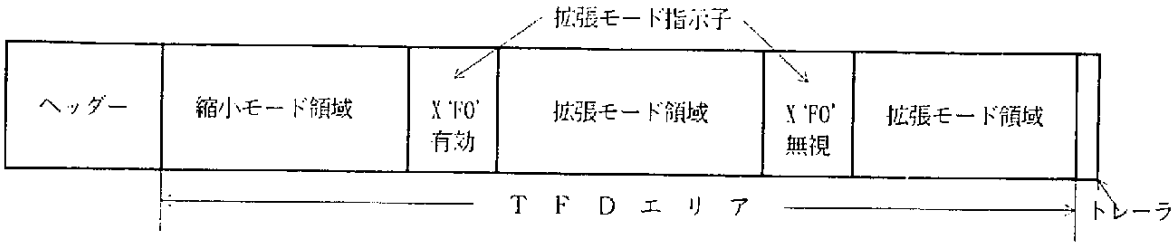


図２－２　メッセージ内のＴＦＤエリアにおける拡張モード指示子の機能

(2)　データ長（データエレメント長）

データ長は、直後のデータエレメントのデータ長をbyte数で表す。１～２３９byteまでのデータエレメントの長さは、１byteの符号無バイナリー数値で表す。２４０～３２７６７byteまでのデータエレメントの長さは、データ長拡張子と２byteの符号無バイナリー数値を用いて表す（図２－３を参照）。

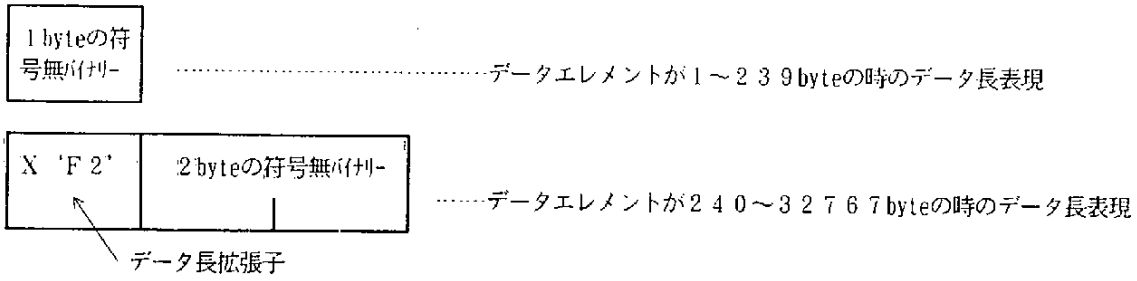


図２－３　データ長（データエレメント長）の形式

(3)　ＴＦＤ形式データ１のまとめ

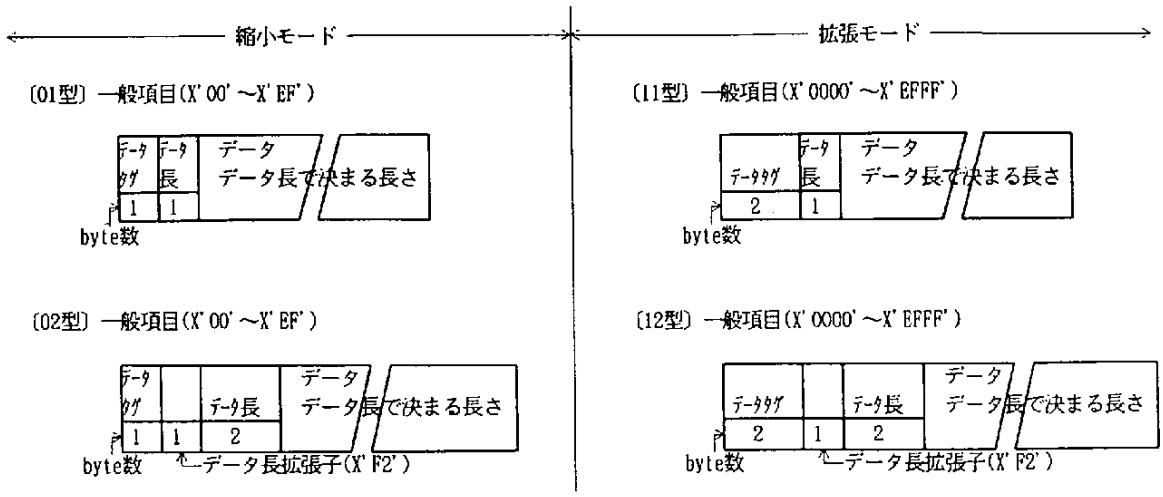
ＴＦＤ形式データ１は、データタグが縮小モードと拡張モードの２種類あり、データ長も通常形式と拡張形式の２種類があるので、両者を組み合わせることにより、合計４種類ある。

①　縮小モードデータタグ ＋ 通常形式データ長……（０１型ＴＦＤである(図２－４の０１型))

②　縮小モードデータタグ ＋ 拡張形式データ長……（０２型ＴＦＤである(図２－４の０２型))

③　拡張モードデータタグ ＋ 通常形式データ長……（１１型ＴＦＤである（図２－４の１１型))

④　拡張モードデータタグ ＋ 拡張形式データ長……（１２型ＴＦＤである（図２－４の１２型))



注）（）内は、データタグ（項目No.）の値の範囲を示す。

図２－４　ＴＦＤ型式データ１の構造

２－１－４　ＴＦＤ形式データ２（ＴＦＤ制御子）

ＴＦＤ制御子は、メッセージのＴＦＤエリア内の各種モードをコントロールする。制御子は、常に１byteの符号無バイナリーの数値（データタグ：制御タグとも言う）で構成され、制御データを持つ場合と持たない場合がある。例えば、拡張モードにおけるマルチ明細ヘッダーは、制御データとして１byteの明細番号を持つ（図２－５及び２－３－５を参照）。

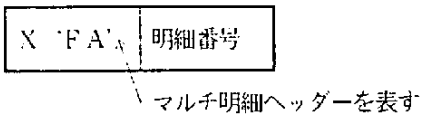


図２－５　　マルチ明細ヘッダーの構造（拡張モード）

２－１－５　ＴＦＤエリア

一つ又は複数のＴＦＤ形式データ１（データエレメント）と一つ又は複数のＴＦＤ形式データ２（ＴＦＤ制御子）で、ＴＦＤエリアを構成する。ＴＦＤエリアは可変長データエレメントの集合体である。

ＴＦＤエリアの最後部（ＴＦＤエリアの終わり）には、ＴＦＤ－ＥＮＤ指示制御子『Ｘ‘ＦＥ’』を置かなければならない。ＴＦＤ制御子『Ｘ‘ＦＥ’』は、メッセージ・トレーラを兼ねる。

2－1－6　メッセージ

一つのメッセージ・ヘッダー、一つのＴＦＤエリア及びメッセージ・トレーラでメッセージを構成する。メッセージ・ヘッダーは、常にメッセージの先頭に置き、次いで、ＴＦＤエリアを置かなければならない。メッセージ・トレーラはメッセージの最後部に置かれ、ＴＦＤエリアの最後部に置かれるＴＦＤ制御子『Ｘ‘ＦＥ’』で代用される（兼用される）。

2－1－7　メッセージグループ・ヘッダー

メッセージグループ・ヘッダーは、固定長データエレメントで構成される全長２５１byteの固定フォーマットの文字列である。

2－1－8　メッセージグループ・トレーラ

メッセージグループ・トレーラは、固定長データエレメントで構成される全長２５１byteの固定フォーマットの文字列である。

2－1－9　メッセージグループ

一つのメッセージグループ・ヘッダー、一つ又は複数のメッセージ及び一つのメッセージグループ・トレーラで、メッセージグループを構成する。メッセージグループ・ヘッダーは、常に、メッセージグループの先頭に置き、次いでメッセージを置き、最後部にメッセージグループ・トレーラを置かなければならない。

2－1－10　交　　換

一つ又は複数のメッセージグループで、一つの交換を構成する。一つの交換とは、一回の通信システム上の接続に相当し、一つの交換に含まれるすべてのメッセージグループは、その交換では、通信システム上の一つの発信地から一つの着信地に送信される。

2－1－11　レコードとファイル

コンピュータ・システムにおける補助記憶装置の論理的管理単位をファイルと呼び、個々のファイル内の論理的かつ機械的分割単位をレコードと呼ぶ。

但し、ここで言うレコードは、一般的なコンピュータのＯＳにおけるレコードの概念とは、必ずしも一致しない。ＯＳによっては、レコードが存在しないこともある。

ＣＩＩシンタックスルールでは、ファイル内の論理的分割単位をレコードとしており、コンピュータのＯＳにおけるレコードの概念と一致する場合もある。しかしながら、一致しない場合でも、処理上の問題点があるわけではなく、ＯＳにおけるレコードとは切り離したプロセスを構築することが可能である。

2－1－12　格納構造

メッセージグループのファイル／レコードへの格納方法を、格納構造と呼ぶ。

(1)　交換単位の格納構造

一つの交換は、一つのファイルに格納する。

(2)　メッセージグループの格納構造

一つのメッセージグループを一つのファイルに格納してもよいし、複数のメッセージグループを一つのファイルに格納してもよい。しかし、一つのメッセージグループを複数のファイルに格納してはいけない。

(3)　メッセージグループ・ヘッダーとメッセージグループ・トレーラの格納構造

一つのメッセージグループ・ヘッダー及び一つのメッセージグループ・トレーラは、それぞれ一つの２５１byteのレコードに格納しなければならない。

(4)　メッセージの格納構造

一つのメッセージは、一つの可変長レコードに格納するか、一つ又は複数の２５１byteの固定長レコードに格納しなければならない。前者を通常モード、後者を分割モードと呼ぶ。分割モードにおいて、メッセージの長さとレコードの長さが合わない時レコード上の余りの部分は、ブランクで満たさなければならない。

2－2　メッセージグループ・ヘッダー

メッセージグループ・ヘッダーは、ＴＹＰＥ１２で用いられるヘッダーで、図２－６の２５１byte長の固定フォーマットである。メッセージグループ・ヘッダーは、１－３項で示すキャラクターセットと文字コードで表現しなければならない。

一つのメッセージグループ・ヘッダーは、一つの２５１byteのレコードに格納しなければならない。

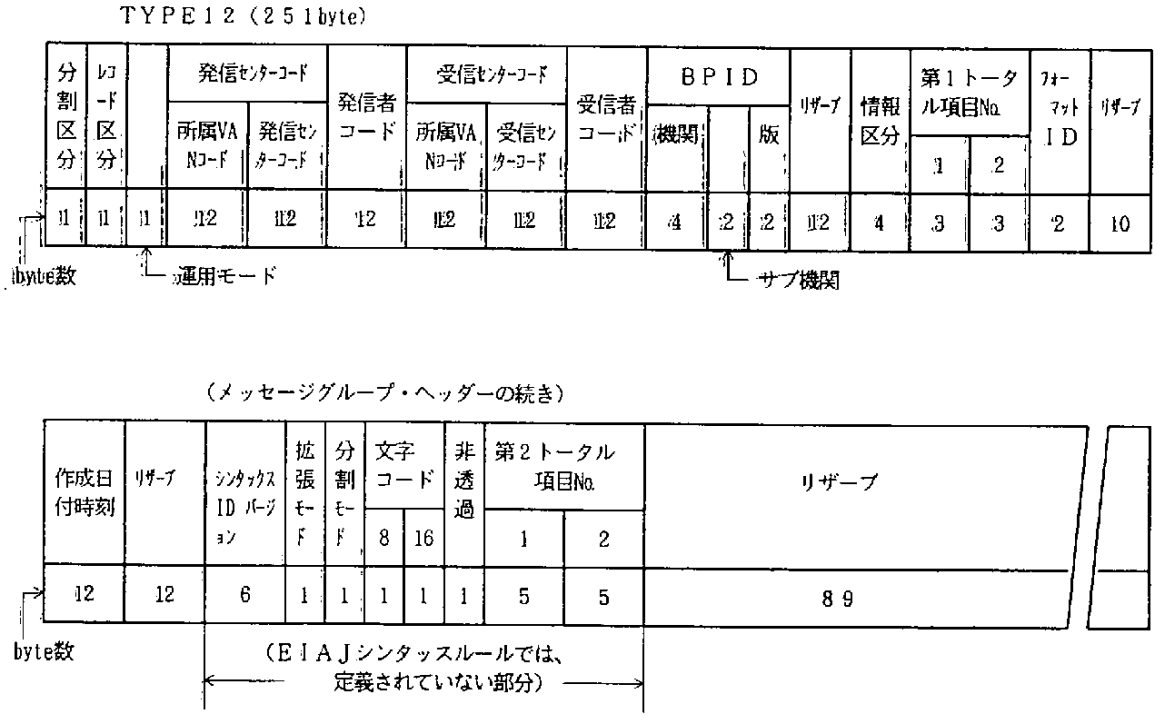


図２－６　メッセージグループ・ヘッダーのフォーマット

① 分割区分　　（文字１byte：０（＝X‘30’））

ＴＹＰＥ１２のヘッダーかトレーラーであること示す。

② レコード区分　　（文字１byte：Ｃ（＝X‘43’））

ＴＹＰＥ１２のメッセージグループ・ヘッダーであること示す。

③ 運用モード　　（文字１byte）

１（=X‘31’）の時テスト・データを表す。０（=X‘30’）または△（=X‘20’）の時通常のデータを表す。

④ 発信センターコード（ｷｬﾗｸﾀｰｾｯﾄ と文字ｺｰﾄﾞ は、表１－２を参照）

ａ．所属ＶＡＮコード　（文字12byte）

ｂ．発信センターコード（文字12byte）

⑤ 発信者コード　　（文字12byte）

ｷｬﾗｸﾀｰｾｯﾄ と文字ｺｰﾄﾞ は、表１－２を参照。

⑥ 受信センターコード（ｷｬﾗｸﾀｰｾｯﾄ と文字ｺｰﾄﾞ は、表１－２を参照）

a．所属ＶＡＮコード　（文字12byte）

b．受信センターコード（文字12byte）

⑦ 受信者コード　　（文字12byte）

ｷｬﾗｸﾀｰｾｯﾄ と文字ｺｰﾄﾞ は、表１－２を参照。

⑧ ＢＰＩＤ（ｷｬﾗｸﾀｰｾｯﾄ と文字ｺｰﾄﾞ は、表１－２を参照）

a．機　　関（文字４byte）

ユニークな、標準メッセージを管理する機関を表すコード

b．サブ機関（文字２byte）

大分類管理コード（標準メッセージ管理機関が管理するコード）

c．　版　　（文字２byte）

バージョン番号（標準メッセージ管理機関が管理するコード）

⑨ リザーブ　　（文字12byte）

将来の拡張エリア（all ﾌﾞﾗﾝｸ(=X'20'))

⑩ 情報区分コード　　（文字４byte）

情報の種類を表す。ｷｬﾗｸﾀｰｾｯﾄ と文字ｺｰﾄﾞ は、表１－２を参照。

⑪ 第１ﾄｰﾀﾙ項目ﾅﾝﾊﾞｰ　　（文字３byte＊２）

後述⑰の『拡張モード』に縮小がセットされている時ハッシュ・トータルに使う項目を指定する。項目No.（ﾃﾞｰﾀ･ﾀｸﾞ値）が２３９以下の９属性またはＮ属性の項目を指定する。２つの項目を指定できる。ﾌﾞﾗﾝｸを指定すると、ハッシュ・トータルを行なわない。ｷｬﾗｸﾀｰｾｯﾄ と文字ｺｰﾄﾞ は、表１－２を参照。

⑫ フォーマットＩＤ　　（文字２byte）

ＥＩＡＪシンタックスルールとの互換を保つため、以下の値とする。

a．ＴＹＰＥ１２通常モード（非分割モード）…… １０(=X'3130')

b．ＴＹＰＥ１２分割モード …… １１(=X'3131')

c．ＴＹＰＥ１２受信確認電文 …… ２０(=X'3230')

尚、透過／非透過モードの区別は本コードの値に影響しない。分割モードについては、⑱を参照

⑬ リザーブ　　（文字10byte）

将来の拡張エリア（all ﾌﾞﾗﾝｸ(=X'20'))

⑭ 作成日付時刻　　（文字12byte）

ﾒｯｾｰｼﾞｸﾞﾙｰﾌﾟの作成日付時刻　YYMMDDHHMMSS。文字コードは、表１－２を参照。

⑮　リザーブ　　（文字12byte）

将来の拡張エリア（all ﾌﾞﾗﾝｸ =X'20'）

⑯　ｼﾝﾀｯｸｽIDﾊﾞｰｼﾞｮﾝ 番号　（文字６byte）

シンタックスルール管理機関およびリリース・バージョン番号を表す。ｷｬﾗｸﾀｰｾｯﾄと文字ｺｰﾄﾞは、表１－２を参照。

⑰　拡張モード　　（文字１byte）

ﾌﾞﾗﾝｸ(=X'20')またはＳ(=X'53')の時、縮小モード（ＥＩＡＪシンタックスルール互換モード）を表し、Ｅ(=X'45')の時、拡張モードを表す。文字ｺｰﾄﾞは、表１－２を参照。

このエリアが縮小にセットされている時はメッセージが縮小モード（ＥＩＡＪシンタックスルール互換モード）であることを示し、拡張にセットされている時は、メッセージが縮小モードか拡張モードのどちらかであることを示す。

通常、ＥＩＡＪシンタックスルール互換モード（縮小モード）とする時はﾌﾞﾗﾝｸ(=X'20')とし、それ以外の時はＥ(=X'45')にする。

⑱　分割モード　　（文字１byte）

ﾌﾞﾗﾝｸ(=X'20')またはＭ(=X'4D')の時、分割モードを表し、Ｓ(=X'53')の時、通常モード（レコード分割無し）を表す。文字ｺｰﾄﾞは、表１－２を参照。

通常、ＥＩＡＪシンタックスルール互換モード（分割モード）とする時はﾌﾞﾗﾝｸ(=X'20')とし、非分割モードにする時はＳ(=X'53')とする。

⑲　文字コード

メッセージをコーディングしている文字コードの種類を表す。文字コードは、表１－２を参照。

ａ．８ｂｉｔ系　　（文字１byte）

ﾌﾞﾗﾝｸ(=X'20')またはＳ(=X'53')の時８ｂｉｔ系文字がJIS-X0201であり、Ｐ（=X'50'）の時、それ以外を表す。

ｂ．16ｂｉｔ系　　（文字１byte）

ﾌﾞﾗﾝｸ(=X'20')またはＳ(=X'53')の時16ｂｉｔ系文字がJIS-X0208であり、Ｐ（=X'50'）の時それ以外を表す。

尚、ＥＩＡＪシンタックスルールではこのエリアは定義されていないので、いかなる値を設定しても、通常のＥＩＡＪトランスレーターに対して効果を持たないが、JIS-X0201, JIS-X0208を指定する時にはﾌﾞﾗﾝｸ(=X'20')を用いる方が望ましい。

⑳　非透過モード　　（文字１byte）

ﾌﾞﾗﾝｸ(=X'20')またはＳ(=X'53')の時、透過モードを表し、Ｍ(=X'4D')の時、非透過モー

ドを表す。文字コードは、表１－２を参照。非透過モードについては、２－６を参照。

尚、ＥＩＡＪシンタックスルール互換モード（透過モード）とする時は、ﾌﾞﾗﾝｸ(=X'20')にする必要がある。

㉑　第２ﾄｰﾀﾙ項目ﾅﾝﾊﾞｰ　　（文字５byte＊２）

⑰の『拡張モード』に拡張がセットされている時、ハッシュ・トータルに使う項目を指定する。９属性またはN属性の項目を指定する。２つの項目を指定できる。ﾌﾞﾗﾝｸを指定すると、ハッシュ・トータルを行なわない。また、⑪の第１ﾄｰﾀﾙ項目ﾅﾝﾊﾞｰは指定があっても、無視する。文字コードは、表１－２を参照。

尚、⑰の『拡張モード』に縮小がセットされている時は、このエリアをブランクにしなければならない。

㉒　リザーブ　　（文字89byte）

将来の拡張エリア（all ﾌﾞﾗﾝｸ(=X'20')）

2－3　メッセージ（可変長）の構造（ＴＹＰＥ１２）

一つのメッセージは、メッセージ・ヘッダー，ＴＦＤエリア，メッセージ・トレーラで構成され、一つのトランザクション、例えば１通の発注書に相当する。又、通常モードでは、一つのメッセージは一つの可変長レコードに格納し、分割モードでは、 250byte（分割識別子を除く）づつ単純に分解して、これに分割識別子を付加した 251byteの固定長レコード（ 250byte未満の時はブランクを右側に追加して 250byteの１レコードにする）に格納する。レコードへの格納方法を除けば、通常モードと分割モードは同一の構造である。以下では、通常モードのメッセージの構造を説明する。

2－3－1　メッセージの全体構造

図２－７の構造である。ＴＦＤエリア（Transfer Form Data Area ）は可変長であり、この中にＴＦＤ型式のデータが収容される。ＴＦＤ型式データの構造は、２－１－３及び２－１－４を参照。

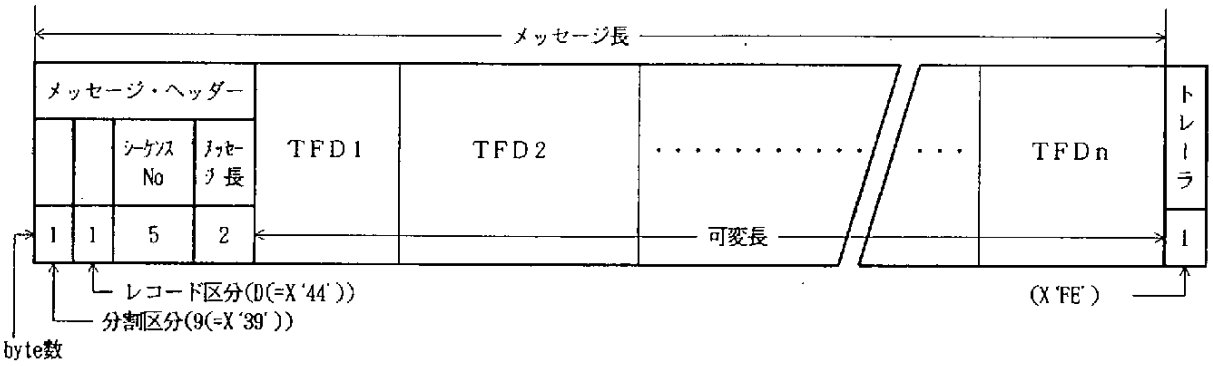


図２－７　メッセージの全体構造

2－3－2　メッセージ・ヘッダーの詳細

① 分割区分　(9(=X'39'))

メッセージを収容したレコードの分割識別子

② レコード区分(D(=X'44'))

メッセージを収容したレコードであることを表す。

③ シーケンスNo.（数値５桁ｱﾝﾊﾟｯｸ 型式）

メッセージを収容したレコードの昇順の５桁のシーケンス番号である。メッセージグループ・ヘッダー・レコード直後のレコードの値は１で、以後１づつ増加する。文字コードは表２－１を参照。

④　メッセージ長（数値２byteﾊﾞｲﾅﾘ型式）

メッセージを収容したレコードの先頭から、トレーラの手前までの全長を表す。トレーラはこの長さに含まれないことに注意する。

メッセージグループ・ヘッダーで透過モードが設定されている時は、数値の最大値は、32767 である。これより長いメッセージの時は、このエリアの値を 32896(X'8080')とし、『２－３－５』で示すメッセージ長指示子によって、メッセージ長を表す。メッセージ長指示子を用いる場合は、メッセージ長指示子は、図２－７のＴＦＤ１の位置におかなければならない。

メッセージグループ・ヘッダーで非透過モードが設定されている時は、このエリアの値を 32896(X'8080')とし、『２－３－５』で示すメッセージ長指示子によって、メッセージ長を表す。メッセージ長指示子を用いる場合は、メッセージ長指示子は、図２－７のＴＦＤ１の位置におかなければならない。

２－３－３　ＴＦＤエリア（Transfer Form Data Area ）の詳細

ＴＦＤエリアは、ＴＦＤ形式データ１（２－１－３参照）及びＴＦＤ形式データ２（２－１－４参照）で構成する。

ＴＦＤエリアは常に縮小モードで始まり、拡張モード指示子が出現した時点で拡張モードに切り換る。この拡張モードは、同一メッセージグループ内の次のメッセージのＴＦＤエリアに影響しない。拡張モードで終わったメッセージの次のメッセージのＴＦＤエリアは、再び縮小モードで始まる。

拡張モード指示子がＴＦＤエリアにない時は、ＴＦＤエリア全体が縮小モードであり、さらに、メッセージグループ内のすべてのメッセージのＴＦＤエリア全体が縮小モードの時メッセージグループ・ヘッダーの『拡張モード』が『Ｓ』になる。メッセージグループ内にひとつでも拡張モードのメッセージが存在する場合は、メッセージグループ・ヘッダーの『拡張モード』は『Ｅ』にしなければならない。通常のＣＩＩトランスレーター（送信用）では、『拡張モード』の初期値を『Ｓ』とし、拡張モードのメッセージを作成した時点で、『拡張モード』を『Ｅ』に変更する。

ＴＦＤ形式データは、常にタグ（データタグ又は制御タグ）で始まる。縮小モードの時、データタグは１byteのバイナリー数値であり、拡張モードの時、データタグは２byteのバイナリー数値である。制御タグは、縮小／拡張モードによらず常に１byteのバイナリーの数値である。データタグか制御タグかは、タグの最初の１byteの値によって決定される。この１byteの値が 240(X'F0')より小さい時データタグであり、 240(X'F0')以上の時制御タグである。

制御タグは、ＴＦＤエリアのモードや繰り返し構造のコントロールを行う。そこでＴＦＤ制御子とも呼ぶ。すなわち、データタグや制御タグ（ＴＦＤ制御子）は、ＣＩＩシンタックスルールの最も重要な要素であり、**このため、ＣＩＩシンタックスルールはタグ方式と呼ぶ。**

ＴＦＤエリアを参照する時は、必ずエリアの先頭（左端）から参照しなければならない。エリアの左端には、必ず、データタグか制御タグが存在する。

２－３－４　ＴＦＤの省略

一つのメッセージに含めなければならないＴＦＤ（データ項目）は、標準メッセージによって規定される。しかしながら、実際の業務で必要とするＴＦＤ（データ項目）が、標準メッセージによって規定されるＴＦＤ（データ項目）よりも少ないことがある。このような時、不要なＴＦＤ（データ項目）を省略することができる。ＴＦＤ形式データ１（一般データ項目）の省略については何らの制限もない。さらに、数値がゼロの数値ＴＦＤや内容が allブランクの文字列ＴＦＤは、一般的に省略可能である。

ＴＦＤ形式データ２（ＴＦＤ制御子）については、一般に省略できない。しかしながら、マルチ明細制御子（マルチ明細ヘッダー、改行マーク及びマルチ明細トレーラ）は、条件により省略できることがある（２－３－７参照）。

２－３－５　ＴＦＤ制御子の機能

ＴＦＤ形式データ２は、ＴＦＤ制御子と呼ばれ、ＴＦＤエリアのコントールを行う。ＴＦＤ制御子は、１byteのバイナリーの数値であるが、ある種のＴＦＤ制御子は制御データを保持する。制御データは固定長である（図２－８を参照）。

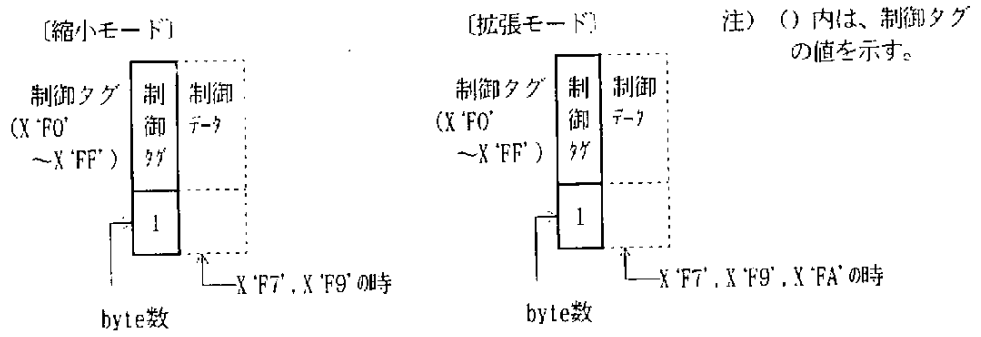


図２－８　ＴＦＤ制御子の構造

(1) 拡張モード指示子(240=X'F0')

ＴＦＤエリアに、この制御タグが出現すると、それより右側のＴＦＤエリアは、拡張モードになり、拡張モードのＴＦＤが存在すると見なす。拡張モード指示子が同一のＴＦＤエリアに複数個存在する場合は一番左側の拡張モード指示子が有効で、それ以外は無視する（図２－９を参照）。

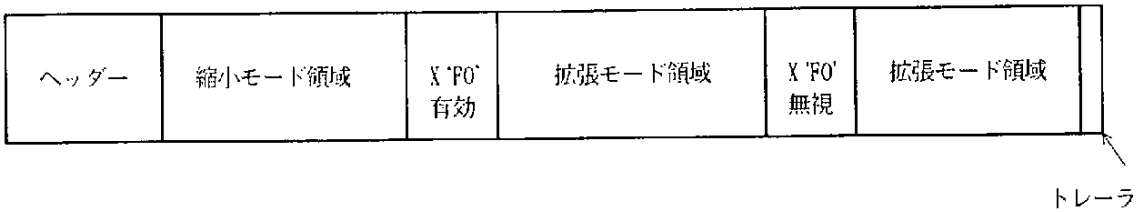


図２－９　ＴＦＤエリアにおける拡張モード指示子の機能

(2) データ長拡張子(242=X'F2')

データ長が２４０byte以上の時、これを付加することで、データ長が２byteで表現されていることを示す。この拡張子は、常に、データタグの直後でデータ長エリアの直前になければならない。

データ長の直前にデータ長拡張子(X'F2')が付加することで、データ長の最大長が、３２７６７byteまで拡張される。通常のトランスレーターは、データ長が２４０byte以上になると自動的にデータ長拡張子(X'F2')を付加し、データ長を２byteで表現する。

尚、単一項目のデータ長が２４０byte以上の時は、縮小モードでも、ＥＩＡＪシンタックスルールとは互換性がない。

(3) メッセージ長指示子(247=X'F7')

メッセージ長指示子はメッセージ長を表す制御タグであり、制御データとして７byteのメッセージ長を保持する。この制御タグは、ＴＦＤエリアの先頭（左端：図２－７のＴＦＤ１の位置）にある時のみ有効で、その他の場所に存在する時はエラーである。７byteのメッセージ長は、メッセージ・トレーラ（データセグメント・セパレーター（ＴＹＰＥ－Ｅの時））を除く、メッセージ長を表す。メッセージ長は、十進キャラクター表現である。

メッセージ長指示子を用いる時は、メッセージ・ヘッダーのメッセージ長エリアをX'8080'にするとともに、メッセージ長指示子の長さ８byte分メッセージ長を増加させなければならない。メッセージ長指示子は、以下の時に用いる

① 透過モードの時（ＴＹＰＥ１２）

メッセージ長が 32776（byte）～ 9999999（byte）（メッセージ長指示子の長さ８byteを含む）の時、図２－１０に示すように用いる（メッセージ長が 9（byte）～ 32767（byte）の時は、メッセーシ・ヘッダーのメッセージ長で表す）。

② 非透過モードの時（ＴＹＰＥ１２）

メッセージ・ヘッダーのメッセージ長エリアを、X'8080' にし、図２－１０に示すように用いる。メッセージ長は17（byte）～ 9999999（byte）（メッセージ長指示子の長さ８byteを含む）が許される。

メッセージ長指示子を含むＴＹＰＥ１２メッセージは、縮小モードであっても、ＥＩＡＪトランスレーターでは解読不能になる。

③ ＴＹＰＥ－Ｅの時

メッセージ長が 32776（byte）～ 9999999（byte）（メッセージ長指示子の長さ８byteを含む）の時、メッセージ長エリアを 99999として、メッセージ長指示子により、データ長を表す（詳細は、３－３－２を参照）。

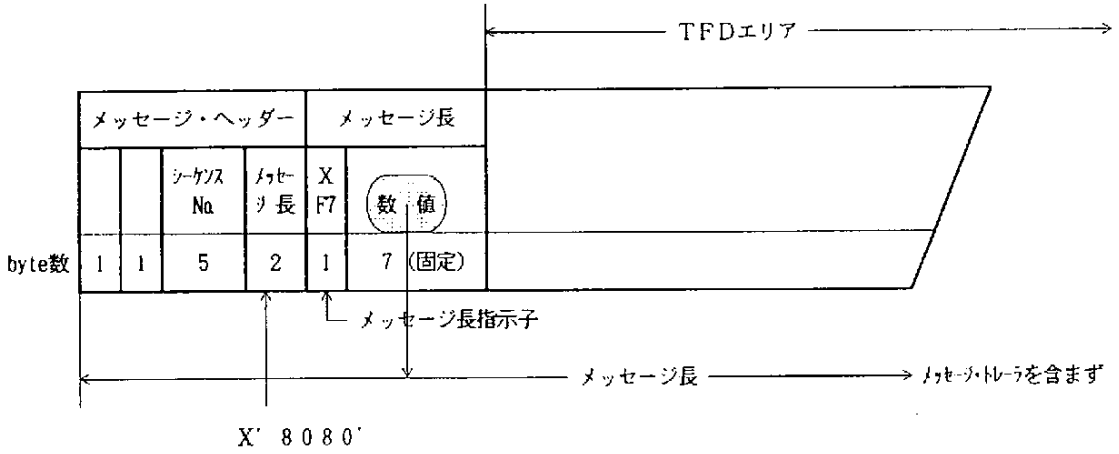


図２－１０　ＴＹＰＥ１２におけるメッセージ長指示子の使い方

(4) バイナリーデータ補正子(248=X'F8')

ＴＹＰＥ１２の非透過モードの時、この補正子の直後の１byteのバイナリーデータが、補正されていることを示す。補正の方法は、２－６を参照。バイナリーデータ補正子は、以下のケースの時のみ有効である。

① 縮小モードで、データタグ（１byte）の直前にある時

② 拡張モードで、データタグ（２byte）の直前、データタグ第２byteの直前にある時

③ 縮小／拡張モードで、レングスタグ（1 byte）の直前、データ長拡張子付レングスタグ（2 byte）の直前と第2 byteの直前にある時

(5) インターナルセグメント指示子(249=X'F9')

メッセージ本体内のＴＦＤエリア内の任意のＴＦＤの集合を、『インターナルセグメント』と定義する。インターナルセグメント指示子は、インターナルセグメントの区切りを示す（３－６も参照）。

① インターナルセグメントは、インターナルセグメント指示子（X'F9'），マルチ明細ヘッダー（X'FA'），改行マーク（X'FB'）およびマルチ明細トレーラ（X'FC'）で区切られるＴＦＤの集合である。

② マルチ明細におけるセグメント

マルチ明細のセグメントは、インターナルセグメントであり、一つの明細内にインターナルセグメント・セパレーターを設けることで、二つのインターナルセグメントに分割することも可能である。

この制御タグは、1 byteの明細番号を持つ。明細番号は、１～９とＡ～Ｚまでの３５種類であるが、オプションとして１５６種類にすることもできる（３－６を参照）。すなわち、一つのメッセージ内に最大３５種類（もしくは１５６種類）のインターナルセグメントを設定できる（３－３－２を参照）。インターナルセグメントはオプションであり、トランスレーターの処理になんらの影響も与えない。送信用トランスレーターではオプション指定によって、インターナルセグメント指示子を挿入する。一方、受信用トランスレーターでは、インターナルセグメント指示子を、単純に読み飛ばす（無視する）。

標準メッセージ上でインターナルセグメントを設定し、このセグメントとＵＮ／ＥＤＩＦＡＣＴにおけるセグメントの構造を一致させることにより、ＴＹＰＥ－ＥとＵＮ／ＥＤＩＦＡＣＴとの間の相互変換が可能になる。

尚、インターナルセグメント指示子を含むＴＹＰＥ１２メッセージは、縮小モードであってもＥＩＡＪトランスレーターでは解読不能になる。

(6) マルチ明細ヘッダー(250=X'FA')

このＴＦＤは、マルチ明細ヘッダーを表し、マルチ明細の始まりを示す（２－３－７を参照）。

拡張モードの時は、このＴＦＤ制御子は1 byteの制御データを保持し（図２－８）、マルチ明細の番号を明示する。一つの標準メッセージ内に２つ以上のマルチ明細がある時、この番号を用いて、明示的に、それぞれのマルチ明細を区別する。明細番号は、1(=X'31'), 2(=

X'32'), ・・・・, 9(=X'39'), A(=X'41'), B(=X'42'), ・・・・, Z(=X'5A')までの35種である。

尚、縮小モードの時はマルチ明細の番号の明示はなく、明細番号は常に『0』として扱う。例えば、縮小モードの時にマルチ明細が始まり、その明細が終わる前に拡張モードへ移行した時は、ゼロ番のマルチ明細として認識する。

通常の送信用トランスレーターは、明示的明細番号付マルチ明細をＴＦＤエリアに組み込む時、拡張モードに移行させる。

(7) マルチ明細改行マーク(251=X'FB')

マルチ明細の１行の終わりを示す（２－３－７を参照）。

(8) マルチ明細トレーラ(252=X'FC')

マルチ明細の終わりを示す（２－３－７を参照）。

(9) トレーラ（メッセージ・トレーラ）(254=X'FE')

ＴＦＤエリアの終りを示す。メッセージの終りでもある。メッセージ・トレーラを兼ねる。

(10) そ の 他

ＴＦＤ形式データ１のデータエレメントのデータ長は、縮小形式の時、最小１byte, 最大２３９byteであり、拡張形式の時、最小２４０byte, 最大３２７６７byteである。いづれの場合も、０byteは許されない。

２－３－６　データタグと項目No.

標準メッセージやデータエレメント・ディレクトリーでは、設計やメンテナンスの合理化のために、データエレメントに整理番号を付け、項目No.とする。ＣＩＩシンタックスルールでは、この項目No.とＴＦＤ形式データ１のデータタグの値を一致させることにより、トランスレータの変換テーブルの運用を効率化させる。従って、新しい項目を標準メッセージやデータエレメント・ディレクトリーに追加する時は、ある規則に従って、その追加項目の項目No.を定める。この規則を、項目No.割当計画と呼ぶ。表２－３は、産業情報化推進センターが推奨する割当計画である。

表2－3　　項目No.割当計画

データタグの値						データ自身の長さ	項目No.の割当（意味及び機能）
拡張モード			縮小モード				
Hex	Dec	長さ	Hex	Dec	長さ		
0000	0	2	00	000	1	1	使用禁止
0001 ～ 00EF	1 ～ 239	2	01 ～ EF	001 ～ 239	1	1 又は 2	一般項目（8 bit文字，16 bit文字）の項目Noを表す。（9，X属性のデータのみ使用し、データ長エリアが1 byte（項目データ長が239byte以下）の時、ＥＩＡＪ標準との互換性領域）
00F0 ～ EA5F	240 ～ 59999	2	無効	無効	無効	1 又は 2	一般項目（8 bit文字，16 bit文字）の項目Noを表す。
EE48 ～ EFFF	61000 ～ 61439	2	無効	無効	無効	1 又は 2	使用禁止（リザーブ，但し、X'EF00'(=61184)～X'EF0F'(=61199)までは、設計画像データ識別子）

2－3－7　データの繰り返し（繰り返しルール：表型式データ）

同一メッセージ内のＴＦＤエリアでは、同一のデータタグ値を持つＴＦＤ形式データ１のＴＦＤ（データ項目）を２個以上含めることは原則としてできない。同一のデータタグ値を持つＴＦＤ形式データ１を複数個含めるためには、繰り返しルールを用いる。

(1)　単独項目の暗示的繰り返し

同一のデータタグ値のＴＦＤを単純に複数個並べることで、同一メッセージ上で、同一項目を複数個伝送できる。この場合、同一項目（ＴＦＤ）を隣合うように並べ、送信側と受信側でメッセージ内での項目の順番の管理を、厳密に行わなければならない。この方式は運用ミスを起こしやすく、推奨できない。なるべく、セグメント化する。

(2)　マルチ明細

いくつかのＴＦＤ（データタグ値は、それぞれ異なる）を集めてセグメントとし、セグメント単位で明示的に繰り返しを行うもので、表型式データの伝送に適用する。セグメントは一つのＴＦＤで構成してもよい。

図２－11のように、最初のセグメントの先頭にマルチ明細ヘッダー，セグメントとセグメントの間に、改行マーク，最後のセグメントの終わりに、マルチ明細トレーラーをセットする。また、必要に応じて各セグメント内の任意のＴＦＤを省略できる。同一セグメント内のすべてのＴＦＤが省略された時は、改行マークだけが残される（図２－12）。

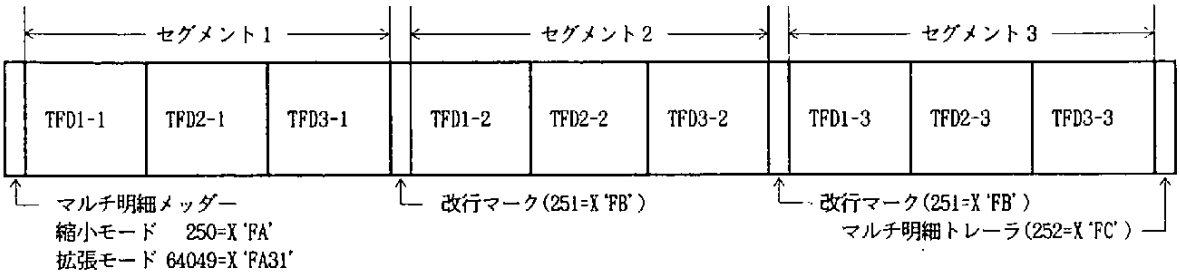


図２－１１　マルチ明細

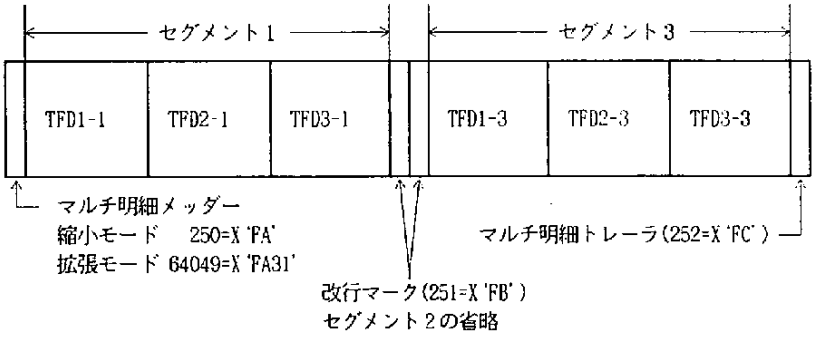
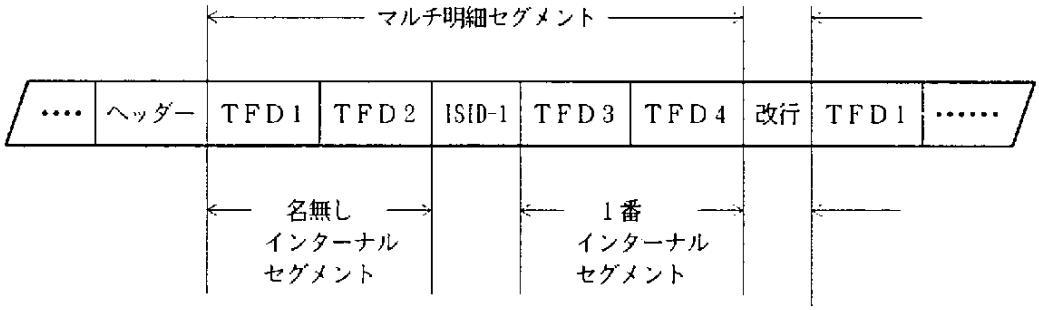


図２－１２　マルチ明細（図２－11　のＴＦＤ１－２，ＴＦＤ２－２，ＴＦＤ３－２が省略された時）

マルチ明細のセグメントの中にインターナルセグメント指示子を含めることもできる（図２－13）。インターナルセグメント指示子は、マルチ明細のコントロールには何らの影響も与えない。



注１）ISID-1：インターナルセグメント識別子（明細番号「１」）

注２）図の名無しインターナルセグメントに名前を付けるためには、
ＴＦＤ１の左側にインターナルセグメント識別子を挿入する。

図２－１３　マルチ明細内のインターナルセグメント

(3) マルチ明細の複数化

縮小モードでは、暗示的にマルチ明細の複数化ができる。この場合、図２－14に示すように、左側のマルチ明細の省略を行う時は、少なくともマルチ明細ヘッダーとマルチ明細トレーラーは残すことが望ましい。

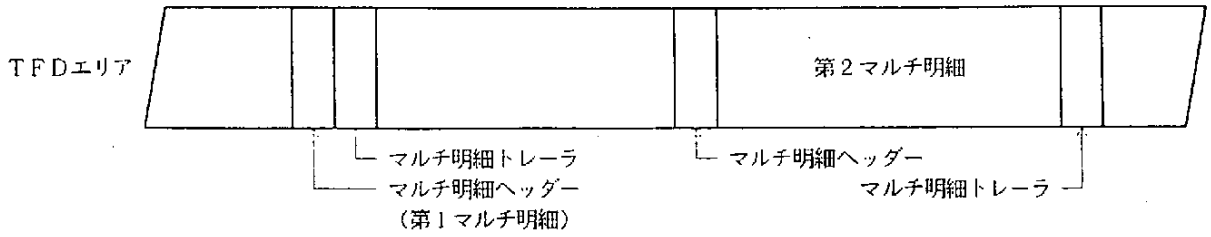


図２－１４　縮小モードにおけるマルチ明細の複数化

拡張モードでは、マルチ明細の番号によって明示的にマルチ明細の複数化を行う。この場合、任意のマルチ明細が省略可能である（図２－15参照）。

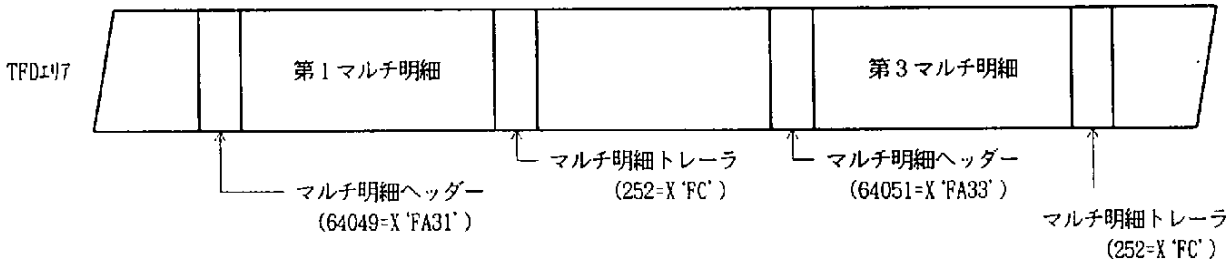


図２－１５　拡張モードにおけるマルチ明細の複数化（第２マルチ明細の省略）

(4) マルチ明細のネスト化（拡張モード）

拡張モードでは、マルチ明細のネスト化ができる。図２－16に示すように、３次元以上の表型式データで用いる。各レベルでのマルチ明細ヘッダーの明細番号は、異なる番号を用いる。また、任意のＴＦＤ，セグメントおよびマルチ明細の全体を省略できる。しかしながら、レベル２のセグメントが省略されない場合、レベル１のセグメントのマルチ明細ヘッダーとマルチ明細トレーラは省略しないことも可能であるが、このことによって誤動作する受信用トランスレータもある。一般的にｎ＜ｍの時、レベルｍのセグメントが省略されない場合、レベルｎのマルチ明細ヘッダーとマルチ明細トレーラを省略しない時は、注意を要する。

縮小モードでのマルチ明細のネスト化は、禁止する。多くの送信用トランスレーターは、明細番号無のマルチ明細によるネスト化を検出した場合、自動的にエラー処理を行う。

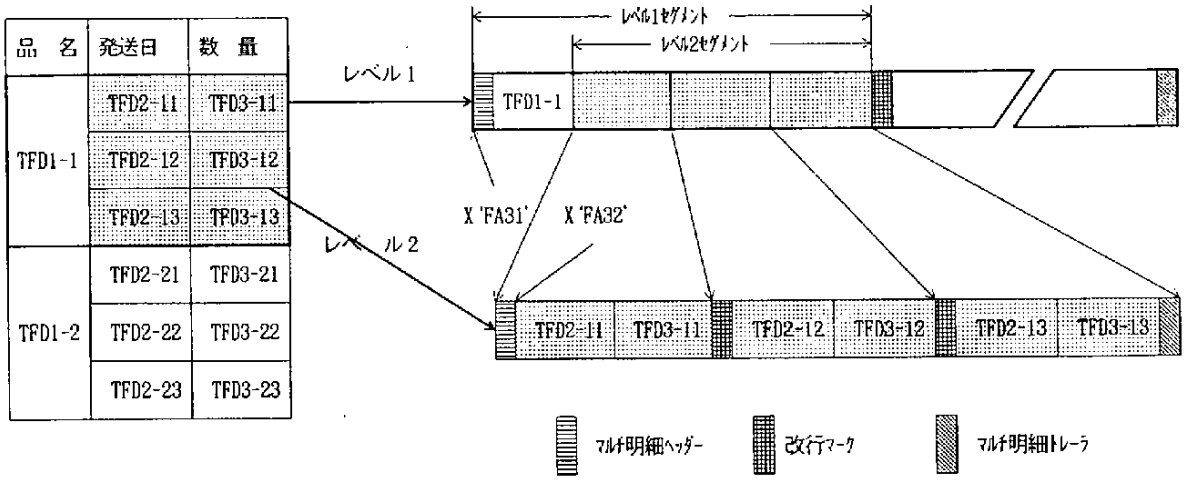


図２－１６　マルチ明細のネスティング

２－３－８　縮小／拡張モードとマルチ明細の関係

(1)　マルチ明細中での拡張モードへの移行

縮小モード中の明細番号無マルチ明細の途中で、拡張モードへ移行した時はそのマルチ明細は、『０』番の明細番号付マルチ明細と見なす。

(2)　拡張モード中の明細番号無マルチ明細

拡張モード中では、明細番号無マルチ明細ヘッダーの使用は許されない。従って、拡張モード中に明細番号無マルチ明細が出現した時（トランスレーターの変換テーブルでこのような指定のケースがある）は、強制的に『０』番の明細番号を付加する。従って、拡張モードの『０』番マルチ明細と縮小モードにおけるマルチ明細は、同一明細として取り扱う。

２－３－９　メッセージ・トレーラ

ＴＦＤエリアの終わりを示すＴＦＤ制御子（１byteの数値（254=X'FE'））で、一つのメッセージの終わりをも示す。メッセージ・トレーラを兼ねる。

2－4　メッセージの格納構造（ＴＹＰＥ１２分割モード）

ＴＹＰＥ１２の通常モードでは一つの可変長のメッセージを一つの可変長レコードに格納する。一方、分割モードでは一つの可変長メッセージを２５０byteづつに分割して、１個あるいはｎ個の２５１byteの固定長レコードに収容する。すなわち、分割モードの時はメッセージの内容自体は同一で、論理レコードへの格納方法が異なる。

レコードの分割は、以下のように行われる。

メッセージ長（分割区分を除く）	1 byte ～ 250byte	……1 レコード	物理レコード数
	251byte ～ 500byte	……2 レコード	
	501byte ～ 750byte	……3 レコード	
	・	・	
	250*(n-1)+1byte ～ 250*n byte	……n レコード	

分割されたメッセージ(250byte) の左先頭に、分割区分（１byte）を追加した251byte のデータが分割化レコードとする。分割区分は、レコードの順番チェックと最終レコードの検出のために、次のように付番する。文字コードは、原則としてJIS-X0201である。

1(=X'31') → 2(=X'32')→・・・・→ 8(=X'38')

→ 1(=X'31')→2(=X'32') →・・・・→ 5(=X'35')→ 9(=X'39')（最終レコード）

つまり、１，２，３，・・・・の順で８まで行き、ふたたび１から８まで番号を付ける。１～８までの繰り返しである。そして、最終レコードは、常に９を付ける。図２－17は、４つに分割された例を示す。最終レコードの右余白には、ブランク(=X'20')を満たす。

尚、メッセージ長が２５０byte以下の時（分割区分を除く）は、１レコードに格納され、分割区分は、9=X'39' である。

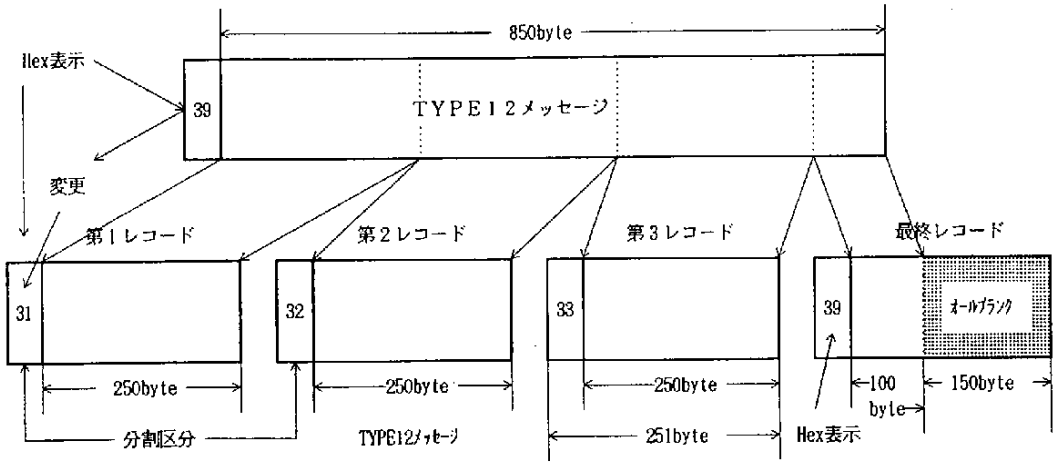


図2－17　ＴＹＰＥ１２分割化固定長レコード（４分割の例）

2－5　メッセージグループ・トレーラ

メッセージグループ・トレーラは、ＴＹＰＥ１２で用いられるトレーラで、図２－18で示す２５１byteの固定フォーマットである。一つのメッセージグループ・トレーラは、一つの２５１byte長のレコードに格納される。

ＴＹＰＥ１２（２５１byte）

		最終ｼｰｹﾝｽ番号	トータル項目No.		リザーブ
			項目合計１	項目合計２	
1	1	5	15	15	２１４

byte数
分割区分(0(=X'30'))
レコード区分(E(=X'45'))

図２－１８　メッセージグループ・トレーラ（ＴＹＰＥ１２）

２－５－１　メッセージグループ・トレーラの詳細

①　分割区分　(0(=X'30'))

ヘッダーもしくはトレーラー・レコードであることを示す。

②　レコード区分(E(=X'45'))

メッセージグループ・トレーラであることを示す。

③　最終シーケンスNo.（数値15桁ｱﾝﾊﾟｯｸ 型式）

直前のメッセージのシーケンスNo.である。すなわち、メッセージの数を表す。メッセージが０件の時は、０になる。文字コードはJIS-X0201 を用いる。

④　トータル項目No.

ａ．項目合計１（数値15桁ｱﾝﾊﾟｯｸ 型式）

チェック用トータル数値１、文字コードは、JIS-X0201 を用いる。

ｂ．項目合計２（数値15桁ｱﾝﾊﾟｯｸ 型式）

チェック用トータル数値２、文字コードは、JIS-X0201 を用いる。

⑤　リザーブ　　（文字 214byte）

将来の拡張エリア(allﾌﾞﾗﾝｸ=X'20')

２－５－２　チェック用トータル数値の算出方法

メッセージグループ・ヘッダーで指定された項目No.（ﾃﾞｰﾀ･ﾀｸﾞ値）の数値データをメッセージ中から取り出し、全体の桁数が15桁より小さい時は、左側に０を必要な数だけ追加し全体の桁数が16桁以上の時は時は、16桁以上をカットして15桁にして、加算する。和が16桁以上になる

時は、その都度16桁以上をカットして15桁にする。一つのメッセージグループ内にある指定されたＴＦＤの数値（データ部）の総計を、チェック用トータル数値とする。項目１と項目２それぞれ独立に計算する。

メッセージグループ・ヘッダーの拡張モード指示子で、縮小モード（ﾌﾞﾗﾝｸ又はＳ）が設定されている場合には、第１トータル項目ナンバー・エリアに設定された３桁の項目No.（データタグ）で示されるＴＦＤのデータ部の数値を用いてチェック値を計算し、拡張モード（Ｅ）が設定されている場合には、第２トータル項目ナンバー・エリアに設定された５桁の項目No.（データタグ）で示されるＴＦＤのデータ部の数値を用いてチェック値を計算する。

計算に用いる数値は、ＴＦＤ形式における数値データをそのまま用いる。Ｎ形式の数値データの時は、正負の符号と小数点を無視して用いる。特に、Ｎ形式において送信用のトランスレーターの不具合により、省略可能な右側のＬＳＢにゼロが存在する場合がある（図２－19）。この場合はゼロを無視しないで計算を行う。

ブランク（=X'20'）または０（=X'30'）の項目No.が指定された場合には、項目合計については、チェック用トータル数値を算出せず 0(=X'30')とする。

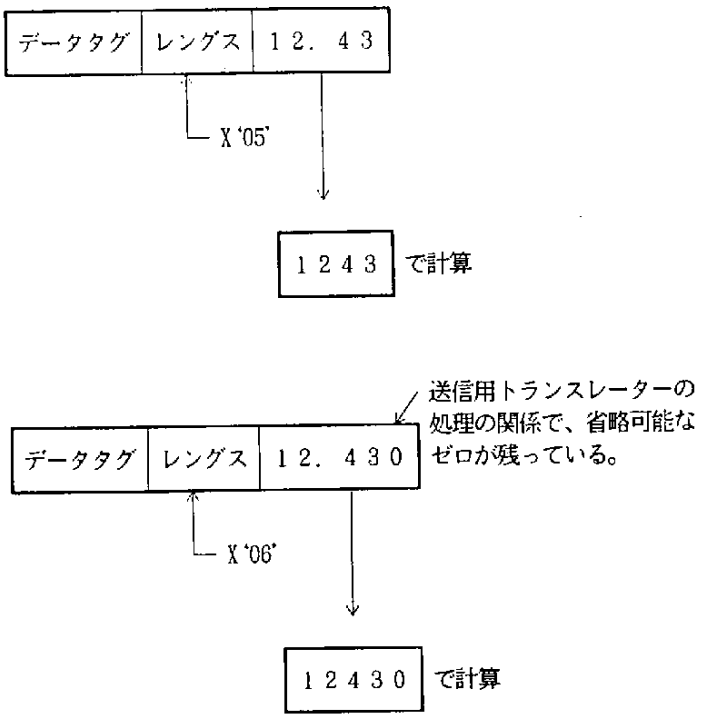


図２－１９　Ｎ形式データにおけるトータルチェック値計算用の数値

ＣＩＩシンタックスルールにおけるチェック用トータル数値の算出方法は、ＥＩＡＪシンタックスルールにおけるそれとは異なるので、縮小モードにおいて互換性を持たせるためには、９属性で小数点以下の桁数が３桁で定義されているデータ項目を、第１トータル項目ナンバー・エリアに設定しなければならない。

２－６　非透過モード時の文字コードの変換およびバイナリーデータの補正

メッセージグループ・ヘッダーの非透過モード指示子で、非透過モード（＝M）が設定された場合、以下のように、文字コードの変換およびバイナリーデータの補正が行われる。尚、非透過モードでは、ＪＩＳ－Ｘ０２０１とＪＩＳ－Ｘ０２０８の文字コードが使用されることを前提に、非透過モード通信系による不具合を回避する対策を行うので、これ以外の文字コードのデータを用いることは許されない。

２－６－１　メッセージグループ・ヘッダーとメッセージグループ・トレーラでの対策

① 分割識別子

メッセージグループ・ヘッダーとメッセージグループ・トレーラの分割識別子は、特に変換をしない（0(=X'30')とする）。これは、受信用トランスレーターが分割識別子によって、メッセージグループ・ヘッダーを識別するために必要な処置である。

② 分割識別子以外

X'20' ～X'3F' を、X'80' ～X'9F' に変換し、伝送する。この変換は、トランスレーターで行われる（表２－４参照）。

受信用トランスレーターでは、透過モードか非透過モードかを判別してメッセージグループを分析しなければならない。メッセージグループ・ヘッダーとメッセージグループ・トレーラについては、次のようにして行う。メッセージグループ・ヘッダーの非透過モード表示は、透過モードの時はﾌﾞﾗﾝｸ(=X'20')かＳ(=X'53')となり、非透過モードの時はM（=X'4D'：非透過モードへの変換後もM(=X'4D')）となる。従って、M(=X'4D')であるかどうかを判定して、非透過モードの時は、メッセージグループ・ヘッダーとメッセージグループ・トレーラの分割識別子を除いて、表２－４に従って逆変換を行ってから分析を行う。

２－６－２　メッセージの対策

① メッセージ・ヘッダー

メッセージ・ヘッダーの分割識別子，レコード区分，シーケンスNo.は、表２－４に従って変換を行う。そして、メッセージ長は、X'8080' とし、メッセージ長指示子（２－３－５の(3)参照）によって、メッセージ長を表す。但し、メッセージ長を表す文字データの文字コードの X'30'～X'39' は表２－４に従って、 X'90'～X'99' に変換して表さなければならない。さらに、メッセージ長が増加（８byte）するので注意する。

受信用トランスレーターでは、メッセージグループ・ヘッダーのチェックにより非透過モードを検出した時は、メッセージ・ヘッダーの分割識別子を表２－４に従って逆変換して透過モードの時と同様な方法でメッセージを認識するか、X'91'（但し、分割モード時はX'91' ～ X'98'で最終レコードはX'99' のシーケンス）を検出することで、メッセージを

認識する。

② ＴＦＤ形式データ１のバイナリーデータ（データタグおよびレングスタグ）の補正

表２－５に従って、補正を行う。補正を行うバイナリーデータの直前には非透過モード・バイナリーデータ補正指示子（X'F8'）を追加する。表２－５で、下線が付いているデータは必ず補正しなければならない（例えば、X'01' → X'F881'）。一方（）が付いているデータの補正は任意である。例えば、X'14' は、無補正でもX'14' → X'F894'のように補正しても、どちらでもよい。X'40' のように無補正が指示されているデータは、補正してはいけない。

例）X'00' ⟶ X'F880' ， X'F8' ⟶ X'F8C8'

X'40' ⟶ X'40' ， X'FF' ⟶ X'F8CF'

バイナリーデータの補正を行った時は、メッセージ長が増加する。非透過モード・バイナリーデータ補正指示子（X'F8'）を一つ使うごとに、１byteづつ増加する。

③ ＴＦＤ形式データ１のデータ部の変換

表２－４に従って変換する。この変換は、変換前の文字コードがＪＩＳ－Ｘ０２０１かＪＩＳ－Ｘ０２０８になっていることが前提になっている。すなわち、非透過モードではこれ以外の文字コードを使うことはできない。

④ ＴＦＤ形式データ２（ＴＦＤ制御子の変換）

制御タグは、バイナリーデータであるので表２－５に従って補正するが、X'F0' ～ X'FF'（X'F8'を除く）の範囲にあり、 X'FF'をX'CF' に補正する以外は、無補正である。但し、現在制御タグ X'FF'は定義されていないので、実質的に補正無しとしてよい。X'F8' は、この文字がＴＦＤ制御子（バイナリーデータ補正子）として使用される時は無補正であり、バイナリーデータ（データタグまたはレングスタグ）として使用される時は、表２－５に従って、X'F8' を追加してX'F8C8' にしなければならない(メッセージ長も１増加する)。

ＴＦＤ制御子の制御データについては、以下のようになる。

a．データ長拡張子(X'F2')

この制御タグの制御データは、２byteのバイナリーデータ（レングスタグ）であり、前述②と同じく、表２－５に従って補正を行う。

b．メッセージ長指示子(X'F7')

この制御タグの制御データ（メッセージ長）は表２－４に従って変換する（前述①）。

c．インターナルセグメント指示子(X'F9')

この制御タグの制御データ（インターナルセグメント番号）は表２－４に従って変換する。

d．マルチ明細ヘッダー(X'FA')

この制御タグの制御データ（明細番号）は表２－４に従って変換する。

⑤ メッセージ・トレーラ(X'FE')

変換も補正も特に必要としない（制御タグと見なす）。

表２－４　非透過モード時の文字コードの変換

数値	変換値	数値	変換値	数値	変換値
20	→ 80	30	→ 90	40	変換無
21	→ 81	31	→ 91	↓	
22	→ 82	32	→ 92	↓	
23	→ 83	33	→ 93	↓	
24	→ 84	34	→ 94	↓	
25	→ 85	35	→ 95	↓	
26	→ 86	36	→ 96	↓	
27	→ 87	37	→ 97	7F	
28	→ 88	38	→ 98	A0	変換無
29	→ 89	39	→ 99	↓	
2A	→ 8A	3A	→ 9A	↓	
2B	→ 8B	3B	→ 9B	↓	
2C	→ 8C	3C	→ 9C	↓	
2D	→ 8D	3D	→ 9D	↓	
2E	→ 8E	3E	→ 9E	↓	
2F	→ 8F	3F	→ 9F	DF	

（注）　ﾒｯｾｰｼﾞｸﾞﾙｰﾌﾟ･ﾍｯﾀﾞｰとﾄﾚｰﾗの分割識別子
及びバイナリーデータ以外に適用，
表中の値は、すべてHex 表示

表２－５　非透過モード時のバイナリーデータの補正

数値	補性値	数値	補性値	数値	補性値	数値	補性値
00	→ 80	10	→ 90	(20 | → A0)	(30 | → B0)	40	補正無
01	→ 81	(11	→ 91)	(21	→ A1)	(31	→ B1)	↓	
02	→ 82	(12 | → 92)	(22 | → A2)	32	→ B2	↓	
03	→ 83	(13	→ 93)	(23	→ A3)	(33	→ B3)	↓	
04	→ 84	(14	→ 94)	(24	→ A4)	(34	→ B4)	↓	
05	→ 85	15	→ 95	(25 | → A5)	(35 | → B5)	↓	
06	→ 86	16	→ 96	26	→ A6	(36 | → B6)	↓	
(07 | → 87)	17	→ 97	(27 | → A7)	37	→ B7	↓	
(08	→ 88)	(18	→ 98)	(28	→ A8)	(38	→ B8)	F7	
(09	→ 89)	(19	→ 99)	(29	→ A9)	(39	→ B9)	F8	→ C8
(0A	→ 8A)	(1A	→ 9A)	(2A	→ AA)	(3A	→ BA)	F9	補正無
(0B	→ 8B)	(1B	→ 9B)	(2B	→ AB)	(3B	→ BB)	↓	
(0C	→ 8C)	(1C	→ 9C)	(2C	→ AC)	(3C	→ BC)	↓	
(0D | → 8D)	(1D | → 9D)	2D	→ AD	3D	→ BD	↓	
(0E	→ 8E)	(1E	→ 9E)	2E	→ AE	(3E	→ BE)	FE	
(0F	→ 8F)	(1F	→ 9F)	(2F	→ AF)	(3F	→ BF)	FF	→ CF

（注）　データタグ、制御タグ及びレングスタグのバイナリーデータに適用
表中の値は、すべてHex 表示

3．ＣＩＩシンタックスルールの詳細（ＴＹＰＥ－Ｅ）

ＴＹＰＥ－Ｅは、将来、ＵＮ／ＥＤＩＦＡＣＴと並行使用する時に備えて用意されるオプションで、ヘッダーとトレーラにＩＳＯ９７３５のシンタックスルールを取り入れ、電文全体がＩＳＯ９７３５と同一のセグメント構成ルールに従って、コーディングされる。文字コードは、JIS-X0201(8bit 文字）とJIS-X0208(16bit 文字）の使用を義務づける。

ＣＩＩシンタックスルールのＴＹＰＥ１２は、メッセージグループを構成するメッセージグループ・ヘッダー、メッセージそしてメッセージグループ・トレーラ等の構成要素の外形が、ＩＳＯ９７３５とは異なるため、ＩＳＯ９７３５をサポートするＥＤＩネットワークを利用することができない。ＴＹＰＥ－Ｅは、構成要素の外形がＩＳＯ９７３５と同一になっているので、ＩＳＯ９７３５をサポートするＥＤＩネットワークを利用することができる。但し、メッセージ内部の論理構造はＵＮ／ＥＤＩＦＡＣＴとは異なるため、ＵＮ／ＥＤＩＦＡＣＴメッセージとして用いることはできない。ＣＩＩシンタックスルールＴＹＰＥ－ＥメッセージとＵＮ／ＥＤＩＦＡＣＴメッセージとを相互に変換し、アプリケーションレベルでの相互利用を可能とするコンバーターが、将来サポートされよう。

以下の記述では、8bit を1byteと記述し、X'1A'は、 Hex表示を表すものとする。又、誤解を避けるために、例えば、キャラクターＣは、C(=X'43)のように Hex表示を併用する。又、図示されたフォーマット中の値は、すべて Hex表示とする。さらに、特に断らないかぎり、バイナリーとは、符号無２進数とする。

３－１　交換の階層

ＴＹＰＥ－Ｅの交換の階層は、ＩＳＯ９７３５と同一である。詳細は、ＩＳＯ９７３５の規格書を参照されたい。尚、機能グループは用いない。従ってＵＮＧセグメントとＵＮＥセグメントは用いない。

３－２　ＵＮＡセグメントとＵＮＢセグメント

ＵＮＡ，ＵＮＢセグメントの構造は、ＩＳＯ９７３５と同一で、ＴＹＰＥ１２におけるメッセージグループ・ヘッダーに相当する。ＵＮＡセグメントはオプションであるが、使用を義務づける。

(1)　ＵＮＡセグメント

① コンポーネント・データエレメント・セパレータ …… =X'FC'

② データエレメント・セパレータ …… =X'FF'

③ 小数点記号文字 …… =X'2E'

④ リリース記号 …… =X'FD'

⑤ セグメント・セパレータ …… =X'FE'

(2) ＵＮＢセグメント

以下に、ＴＹＰＥ－ＥとＴＹＰＥ１２のメッセージグループ・ヘッダーとの対応を示す。文字コードは、JIS-X0201 を使用する。尚、ＴＹＰＥ－Ｅでは、発信者コード，受信者コード，トータル項目No.およびフォーマットＩＤは使用しない。

S001	ｼﾝﾀｯｸｽ識別記号		
0001	ｼﾝﾀｯｸｽ識別記号	(文字 4byte)	ＣＩＩＢ(=X'43494942')
0002	ｼﾝﾀｯｸｽ ﾊﾞｰｼﾞｮﾝ番号	(数値 1byte)	1(=X'31')
S002	交換送信者		
0004	送信者識別コード	(文字35byte-Max)	発信センターコード（12文字）
0007	識別コード修飾子	(文字 4byte-Max)	使用しない。
0008	戻り経路用アドレス	(文字14byte-Max)	（発信者）所属ＶＡＮコード（12文字）
S003	交換受信者		
0010	受信者識別コード	(文字35byte-Max)	発信センターコード（12文字）
0007	識別コード修飾子	(文字 4byte-Max)	使用しない。
0014	経路アドレス	(文字14byte-Max)	（発信者）所属ＶＡＮコード（１２文字）
S004	作成日付		
0017	日付	(数値 6byte)	作成日付
0014	時間	(数値 4byte)	作成時刻（秒は使用しない）
0020	交換コントロール参照番号	(文字14byte-Max)	任意の値とする。受信確認要求をするときは A110 (=X'30')等を使用しない。
S005	受信者参照番号，パスワード		
0022	受信者参照 ﾊﾟｽﾜｰﾄﾞ	(文字14byte-Max)	使用しない。
0025	ﾊﾟｽﾜｰﾄﾞ 修飾子	(文字 2byte)	使用しない。
0026	ｱﾌﾟﾘｹｰｼｮﾝ参照番号	(文字14byte-Max)	ＢＰＩＤ（８桁）＋情報区分（４桁）
0029	処理優先順位コード	(文字 1byte)	使用しない。

0031	受信確認要求	(数値 1byte)	受信確認要求をする時は、1(=X'31')とし要求しない時は、省略か0(=X'30')とする。
0032	通信協定識別	(文字35byte-Max)	使用しない。
0035	テスト指示記号	(数値 1byte)	運用モード(通常モードは、0(=X'30')を使用)

3－3　メッセージの構造(TYPE－E)

TYPE12におけるメッセージ・ヘッダーがUNHセグメントに、TFDエリアがCIIセグメントに、メッセージ・トレーラがUNTセグメントにそれぞれ置き換えられる(図3－1参照)。

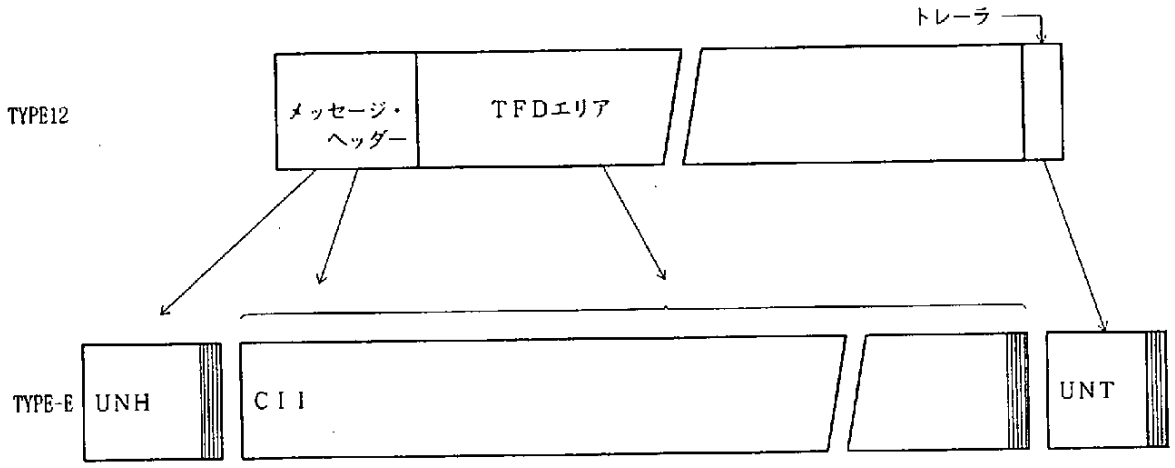


図3－1　TYPE12とTYPE－Eとの対応

3－3－1　UNHセグメント

UNHセグメントは、ISO9735と同一である。以下に、TYPE12のメッセージ・ヘッダーとの対応を示す。文字コードは、JIS-X0201 を使用する。

0062	メッセージ参照番号	(文字14byte-Max)	シーケンスNo.(5桁)
S009	ﾒｯｾｰｼﾞ修飾子		
0065	ﾒｯｾｰｼﾞ･ﾀｲﾌﾟ	(文字 6byte-Max)	情報区分(4文字)
0052	ﾒｯｾｰｼﾞ･ﾊﾞｰｼﾞｮﾝ 番号	(数値 3byte-Max)	版(2文字)

0054	ﾒｯｾｰｼﾞﾘﾘｰｽ 番号	(数値 3byte-Max)	使用しない。
0051	管理機関コード	(文字 2byte)	使用しない。
0057	協会指定コード	(文字 6byte-Max)	使用しない。
0068	共通アクセス参照番号	(文字35byte-Max)	使用しない。
S010	数次更新型データの伝送状況		
0070	伝送のシーケンス	(数値 2byte-Max)	使用しない。
0073	初回伝送か最終伝送か	(文字 1byte)	使用しない。

3－3－2　ＣＩＩセグメント

ＴＹＰＥ１２のメッセージ（非透過モードは不可、透過モードの非分割モードに限る）のメッセージ・ヘッダーが変更されたメッセージが、そのままＣＩＩセグメントとして用いられる。

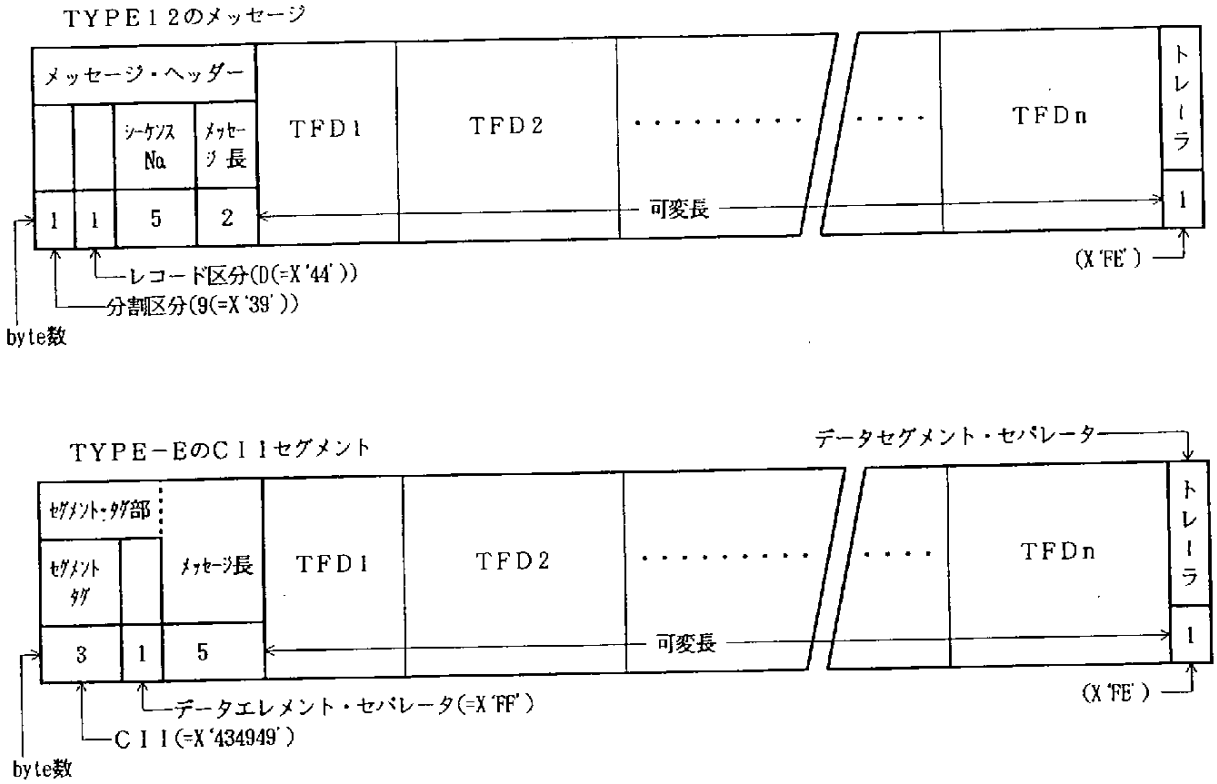


図3－2　ＴＹＰＥ－ＥのＣＩＩセグメント

① セグメントタグ（文字3byte）

ＣＩＩ(=X'434949')

② メッセージ長（文字5byte）

メッセージ長は、アンパック形式の数字5桁が用いられる。文字コードは、ＪＩＳ－Ｘ０２０１を用いる。メッセージ長が、３２７６７byteよりも長い時は、メッセージ長エリアを９９９９９とし、メッセージ長指示子により、図２－１９のＴＦＤ１の位置で、アンパック形式の数字7桁で表す。この時メッセージ長指示子（8byte）の分だけ、メッセージ長が増加する。

尚、ＴＦＤ内のバイナリーデータで、デリミターと衝突するデータについては、リリース文字(=X'FD')の付加を行う。拡張モードのデータタグの第2byte（X'FC' 以上の時），拡張形式のレングスタグ（データ長）の第2byte（X'FC' 以上の時）およびマルチ明細トレーラ（X'FC')が、衝突する。

例）　Ｘ'０２ＦＥ'　⟶　Ｘ'０２ＦＤＦＥ'

　　Ｘ'ＦＣＦＥ'　⟶　Ｘ'ＦＤＦＣＦＤＦＥ'

３－３－３　ＵＮＴセグメント

ＵＮＴセグメントは、ＩＳＯ９７３５と同一である。ＴＹＰＥ１２のメッセージ・トレーラ（X'FE'）に対応する。文字コードはJIS-X0201 を使用する。

0074	ﾒｯｾｰｼﾞ内のｾｸﾞﾒﾝﾄ 数	（数値 6byte-Max）	ＵＮＨとＵＮＴを含むセグメント数
0062	ﾒｯｾｰｼﾞ参照番号	（文字14byte-Max）	ＵＮＨの0062と同一の値（シーケンスNo.（５桁））

３－４　ＵＮＺセグメント

ＵＮＺセグメントは、ＩＳＯ９７３５と同一である。ＴＹＰＥ１２におけるメッセージグループ・トレーラに対応する。文字コードは、JIS-X0201 を使用する。尚、ＴＹＰＥ－Ｅでは、トータル項目No.（項目合計１，項目合計２）は、使用しない。

0036	交換ｺﾝﾄﾛｰﾙ･ｶｳﾝﾄ	（数値 6byte-Max）	メッセージの総数（最終シーケンス番号（５桁））
0020	交換ｺﾝﾄﾛｰﾙ参照番号	（文字14byte-Max）	ＵＮＢの0020と同一とする。

3－5　メッセージの格納構造（TYPE－E）

TYPE－Eでは、UNAセグメントとUNBセグメントを同一のレコードに格納し、その他のセグメントについては、一つのセグメントを一つの可変長レコードに格納する。

3－6　TYPE－Eとインターナルセグメント

CIIシンタックスルールには、ISO9735のセグメントに対応するものとして、マルチ明細がある。通常、TFDエリアはマルチ明細がない時一つのセグメントで構成され、マルチ明細がある時は、マルチ明細とそうでない部分（名無しセグメント）で構成される（図3－3参照）。

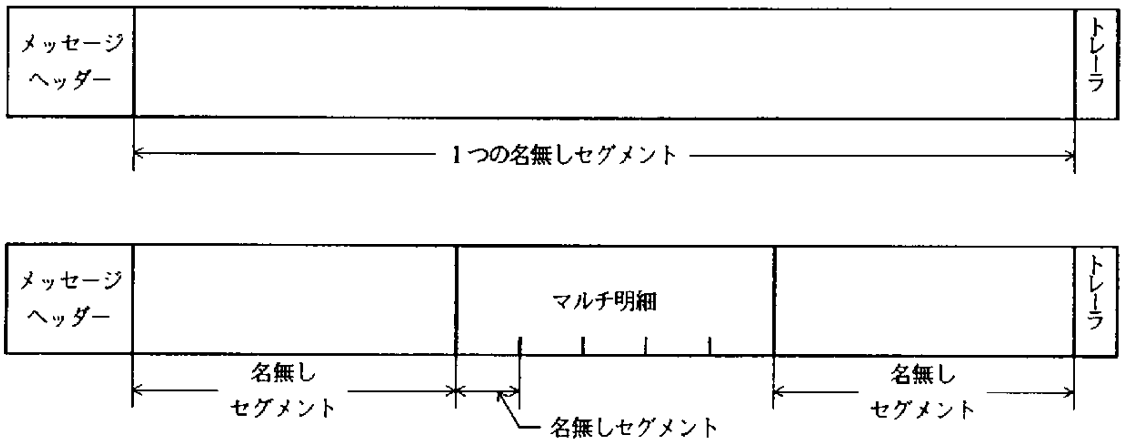


図3－3　通常のCIIシンタックスルール

によるTFDエリアの構造

これに対し、UN／EDIFACT電文への変換を対象にしたTFDエリアは、ISO9735と同等のセグメント構造が必要になる。このような時、インターナルセグメント指示子を用いて、TFDエリアをセグメント化する。

インターナルセグメント指示子を用いてセグメント化したTFDエリアを図3－4に示す。図3－4のインターナセグメントA，B，C及び、マルチ明細セグメント1をUN／EDIFACT電文におけるセグメントに対応させる。UN／EDIFACT電文では、セグメント内のデータ項目の順番は固定化されているが、CIIシンタックスルールにおけるセグメント内のデータ項目（TFD）の順番は可変である。インターナルセグメントはCII－UN／EDIFACTコンバーターで意味をもっており、受信用のCIIトランスレーターに対して何らの影響も与えない（無視する）。

尚、インターナセグメントの名は、１～９とＡ～Ｚの３５種類の使用が規定されているが、表２－１の８bit 文字すべての使用に拡張してもよい。この場合、１５６種類（ﾌﾞﾗﾝｸとゼロを除く）のセグメントをアプリケーションで設定できる。

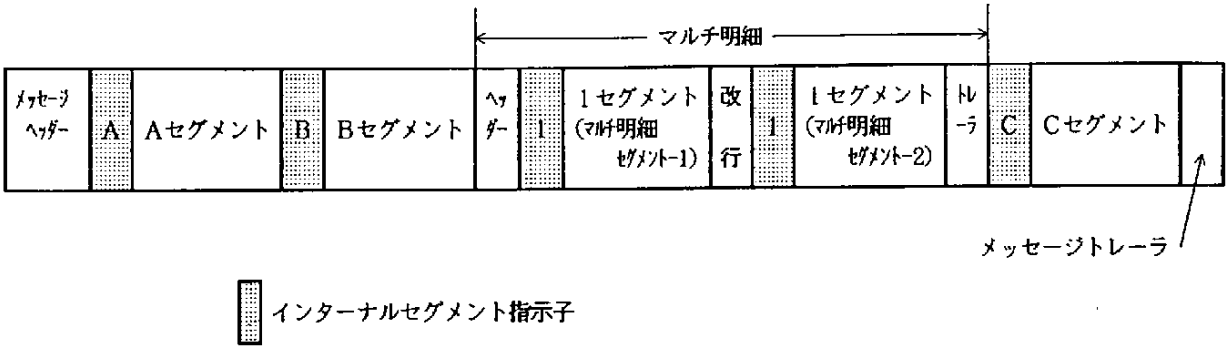


図３－４　インターナルセグメント指示子によるセグメント化

4．システム運用電文フォーマット

システム運用電文は、データ交換に係わるシステム運用を円滑に行うためにあるメッセージであり、受信確認メッセージとゼロ件メッセージの２種類がある。

受信確認メッセージとは、業務メッセージの受信済ステータスを、業務メッセージの受信者から送信者へ知らせるメッセージであり、このメッセージの発信は、業務上の約束の成立を意味しない。例えば、発注メッセージの受信済ステータスをこのメッセージで送信者へ通知しても、発注契約は成立しない。受信確認メッセージは、システム上の電文受信が成功したことを、送信者に伝達するものである。

ゼロ件メッセージは、受信者に対して、送信すべきメッセージ件数がゼロ（無）であることを伝達するために用いるメッセージである。

4－1　受信確認メッセージの構造（ＴＹＰＥ１２）

ＴＹＰＥ１２の受信確認メッセージの構造を、図４－１に示す。

以下に、ＴＹＰＥ１２の受信確認メッセージの詳細を示す。文字コードは、JIS-X0201 を使用する。

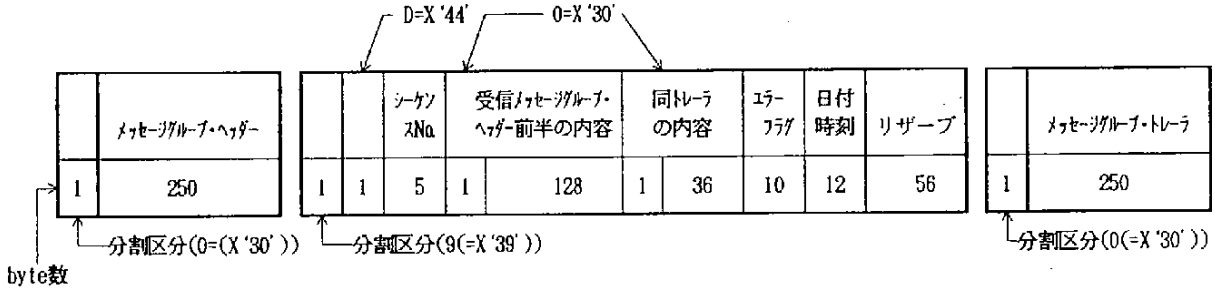


図４－１　ＴＹＰＥ１２の受信確認メッセージ

① ﾒｯｾｰｼﾞｸﾞﾙｰﾌﾟ･ﾍｯﾀﾞｰ　（文字251byte ）

図２－６と同一の固定フォーマット

a．情報区分コード	（文字４byte）	９００１（=X'39303031'）
b．トータル項目ナンバ	（文字３byte*2）	all ﾌﾞﾗﾝｸ（=X'20'）（ハッシュ・トータル無）
c．ﾌｫｰﾏｯﾄＩＤ	（文字２byte）	受信確認電文 20（=X'3230'）
d．その他の項目		図２－６と同一内容

② 受信確認メッセージ　（文字251byte ）

a．レコード区分	（文字１byte）	データレコード（受信確認メッセージ）であることを示す（D（=X'44'））。
b．シーケンスNo.	（数値５byte）	通常、００００１（=X'3030303031'）
c．受信ﾒｯｾｰｼﾞｸﾞﾙｰﾌﾟ･ﾍｯﾀﾞｰの内容	（文字129byte）	受信に成功したメッセージグループ・ヘッダーの分割識別子～作成日付時刻までの 129byteのコピー
d．同トレーラの内容	（文字37byte）	受信に成功したメッセージグループ・トレーラの分割識別子～トータル項目No.（項目合計２）までの 37byte のコピー
e．エラーフラグ	（文字 2byte*5）	受信側で、受信用トランスレーターでメッセージグループを処理した時発生したエラーのエラーコードをセットする。受信用トランスレーターで検出したエラーのエラーコードを検出順に最大５個までセットできる。エラーコードは、表４－１を用いる。 allﾌﾞﾗﾝｸ（=X'20'）か allゼロ（=X'30'）の時、エラー無とする。
f．（作成）日付時刻	（文字12byte）	当該受信確認メッセージの作成時刻（YYMMDDHHMMSSタイプ）
g．リザーブ	（文字56byte）	将来の拡張エリア（all ﾌﾞﾗﾝｸ（=X'20'））

③ ﾒｯｾｰｼﾞｸﾞﾙｰﾌﾟ･ﾄﾚｰﾗ　（文字251byte ）　図２－１８と同一の固定フォーマット

a．最終シーケンス番号	（数値５byte）	通常、００００１（=X'3030303031'）
b．トータル項目No.	（数値15byte*2）	allﾌﾞﾗﾝｸ（=X'20'）（ハッシュ・トータル無）
c．その他の項目		図２－１８と同一内容

尚、受信確認メッセージは、一組のメッセージグループ・ヘッダーとメッセージグループ・トレーラの中に、複数個入れることも可能である。

表４－１　受信確認電文で表示されるエラーコード（その１）

エラーコード	エ　ラ　ー　内　容
△△（ﾌﾞﾗﾝｸ）	エラーなし
００	エラーなし
０１	取決め以外の情報区分コード （トランスレーターの構造によっては、出力されない）
０２	メッセージグループ・ヘッダーが見つからない
０３	メッセージグループ・トレーラが見つからない
０４	シンタックスＩＤの不正
０５	分割区分シーケンスエラー
１０	未定義制御タグの検出
１１	不正データタグ検出
１２	マルチ明細ヘッダーが実行形式ＳＭテーブル上にない
１３	マルチ明細トレーラが実行形式ＳＭテーブル上にない
１４	固有側（標準側）繰り返しが標準側（固有側）繰り返しを超えた
１５	データのレングスが最大値を超えた
１６	チェック・サム項目の値が数値でない
１７	数値変換の項目の値が数値でない
１８	標準側データ長が固有側データ長より大きい
１９	レコード区分がＤでない（メッセージが見つからない）
２０	過大レコード長

表４－１　受信確認電文で表示されるエラーコード（その２）

エラーコード	エ　ラ　ー　内　容
２１	メッセージ・トレーラ（X'FE'）がない
２２	負のデータ有り（９属性の時など）
３０	シーケンスNo.が昇順ではない
３１	チェックサムの数値がメッセージグループ・トレーラ上の数値と一致しない
３２	実行形式ＳＭテーブルサーチ不能
３３	不正文字コードの検出
３４	非透過モード時のメッセージ長不正
３５	縮小モード中のマルチ明細のネスト、もしくは拡張モードマルチ明細ヘッダー有り
３６	Ｙ属性（日付）データの不正
４０	ＵＮＡセグメントの不正
４１	ＵＮＢセグメント無し
４２	ＵＮＨセグメント無し
４３	ＵＮＴセグメント無し
４４	ＵＮＺセグメント無し
９９	その他のエラー

4－2　受信確認メッセージの構造（ＴＹＰＥ－Ｅ）

ＴＹＰＥ－Ｅの受信確認メッセージは、図４－２に示すように、ＵＮＡ，ＵＮＢ，ＵＮＨ，ＵＮＴおよびＵＮＺセグメントで構成される。文字コードは、JIS-X0201 を使用する。尚、一組のＵＮＡ，ＵＮＢ，ＵＮＺセグメントの中に、複数のＵＮＨ，ＵＮＴセグメントの対を入れることができる。

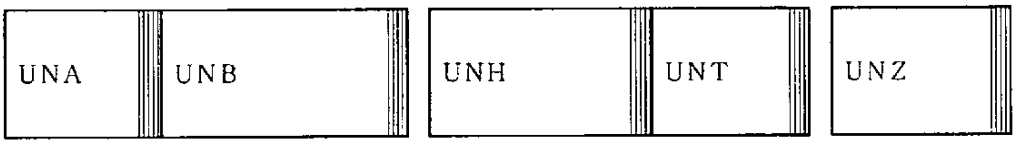


図４－２　ＴＹＰＥ－Ｅ受信確認メッセージの構造

(1)　ＵＮＡセグメント（３－２を参照）

(2)　ＵＮＢセグメント

①　0026　ｱﾌﾟﾘｹｰｼｮﾝ参照番号

ＢＰＩＤ（８桁）＋情報区分（９００１(=X'39303031')）

②　0031　受信確認要求

使用しない（省略か０(=X'30')）。受信確認要求を、してはいけない。

③　その他の項目は、３－２を参照

(3)　ＵＮＨセグメント

①　0065　ﾒｯｾｰｼﾞ･ﾀｲﾌﾟ

情報区分

②　0068　共通ｱｸｾｽ参照番号

受信に成功したＵＮＢセグメントの0020交換ｺﾝﾄﾛｰﾙ参照番号（文字14byte-Max）をセットする。

③　その他の項目は、３－２を参照

(4)　ＵＮＴセグメント（３－３－１を参照）

(5)　ＵＮＺセグメント（３－４を参照）

4－3　ゼロ件メッセージの構造

ゼロ件メッセージは、図4－3に示すように、TYPE12については、メッセージのない（0件）メッセージグループ・ヘッダーとメッセージグループ・トレーラの対であり、TYPE－Eについては、UNA，UNBおよびUNZセグメントだけで構成される電文（UNH，UNTセグメントの対が0件）である。

TYPE12のゼロ件メッセージでは、メッセージグループ・ヘッダー及びメッセージグループ・トレーラの中のトータル項目No.を、すべてブランク（=X'20'）としてハッシュ・トータル・チェックをバイパスし、さらに、メッセージグループ・トレーラの中の最終シーケンス番号をゼロ（メッセージ無）とする。

TYPE－Eのゼロ件メッセージでは、UNZセグメントの中の交換コントロール・カウントは、ゼロ（メッセージ無）とする。

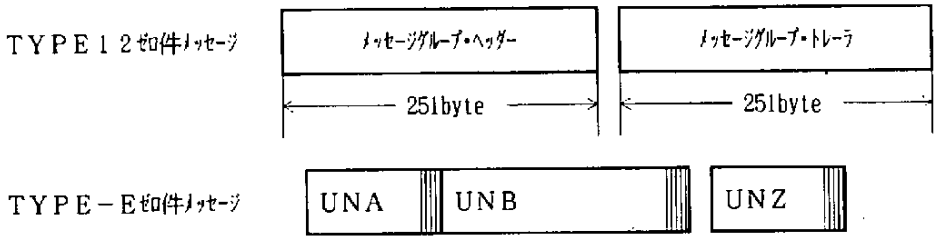


図4－3　ゼロ件メッセージの構造

5．設計画像ＥＤＩ用拡張仕様

本拡張仕様は、設計画像ＥＤＩ用に、ＣＩＩシンタックスルール１．１１に設計画像データ格納機能を追加したものである。拡張されたＣＩＩシンタックスルールは、バージョン１．５１として区別される。バージョン１．５１は、ＴＹＰＥ１２、通常／分割モード、透過モード、拡張モードの延長上で定義される。図５－１のように、設計画像データオプションが追加され、このオプションの無い規格がバージョン１．１１である。

また、バージョン１．５１は常に、拡張モードとして定義される。

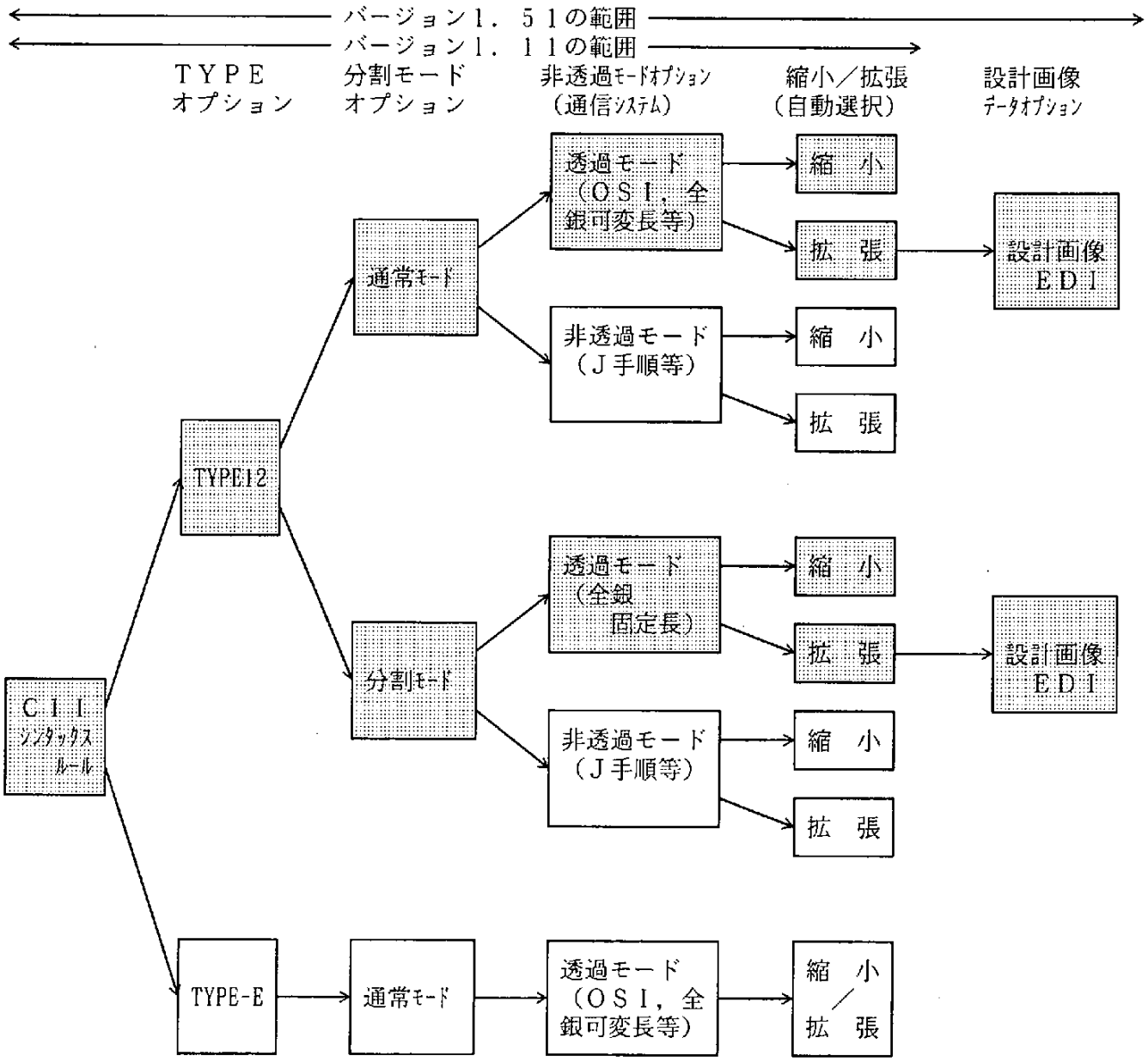


図５－１　ＣＩＩシンタックスルールの１．１１と１．５１

ＣＩＩシンタックスルール１．５１はＣＩＩシンタックスルール１．１１を完全に包含している。以下では、ＣＩＩシンタックスルール１．５１によって拡張された部分の仕様を記述する。

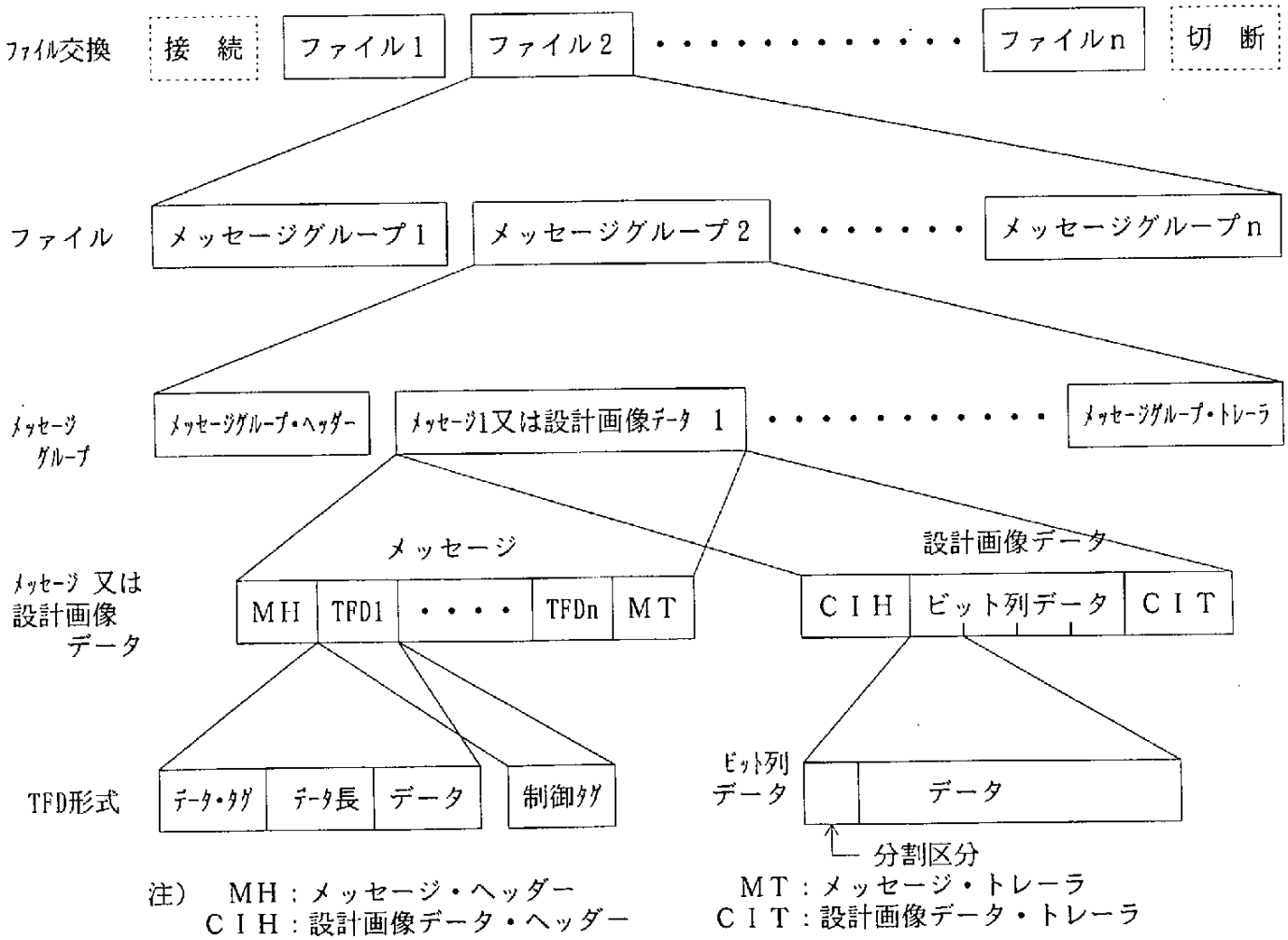
５－１　メッセージ／設計画像データと交換の階層

ＣＩＩシンタックスルール１．５１では、新たに『設計画像データ』が定義される。設計画像データはメッセージと同等の位置づけを持っており、メッセージグループは、メッセージと設計画像データで構成される。メッセージと設計画像データを任意にメッセージグループに含めることができ、メッセージグループは以下の３種が存在可能となる。また、交換の階層は、図５－２のようになる。

① メッセージだけのメッセージグループ（従来のメッセージグループに相当）

② メッセージと設計画像データの混在

③ 設計画像データだけのメッセージグループ



注）　ＭＨ：メッセージ・ヘッダー　　ＭＴ：メッセージ・トレーラ
　　ＣＩＨ：設計画像データ・ヘッダー　　ＣＩＴ：設計画像データ・トレーラ

図５－２　ＣＩＩシンタックスルール１．５１の交換の階層

5－2　設計画像データの基本構造

一つの設計画像データは一つの設計画像データ・ヘッダー、一つのビット列データ、一つの設計画像データ・トレーラで構成される。

5－2－1　設計画像データ・ヘッダー

設計画像データ・ヘッダーは、２５１byteの固定フォーマットのデータで、メッセージグループ内において一つの設計画像データの始まりを表す。

5－2－2　ビット列データ

ビット列データは、ＣＡＤ／ＣＡＭデータ、イメージデータそのもので、byte単位のビット列とする。但し、通常モードにおいて３２０００byteあるいは分割モードにおいて２５０byteを越える場合は、格納構造の規定にしたがって分割される。

なお、ビット列データのフォーマット、すなわちＣＡＤ／ＣＡＭデータ、イメージデータそのもののフォーマットについては、本シンタックスルールでは規定されない。

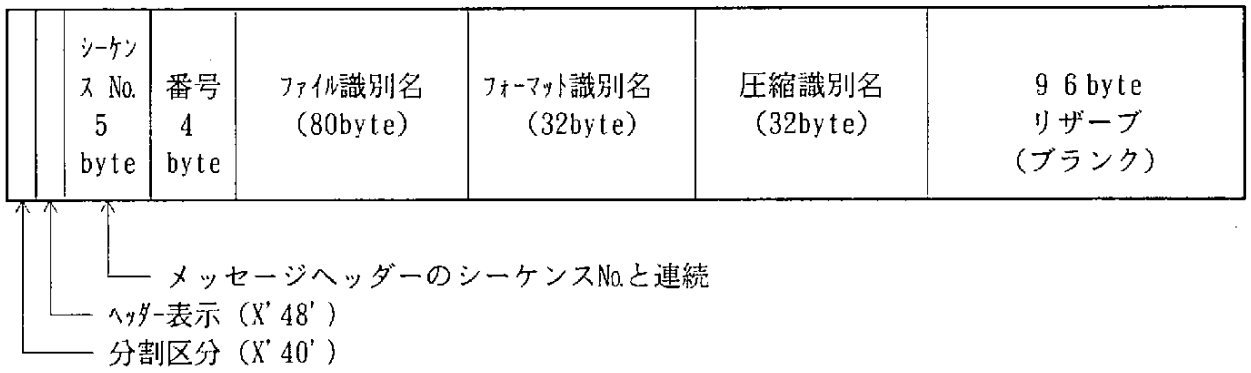
5－2－3　設計画像データ・トレーラ

設計画像データ・トレーラは、一つの設計画像データの終わりを表す。

5－3　設計画像データの詳細構造

5－3－1　設計画像データ・ヘッダーのフォーマット

設計画像データ・ヘッダーは、２５１byteの固定フォーマットで、図５－３の構造である。



注）文字コードはすべてＪＩＳ－Ｘ０２０１（表１－１参照）

図５－３　設計画像データ・ヘッダーのフォーマット

① 分割区分　　（1 byte：X'40'）

設計画像データのヘッダーかトレーラであることを表す。

② ﾍｯﾀﾞｰ表示　　（1 byte：X'48'）

設計画像データのヘッダーであることを表す。

③ シーケンス№　（5 byte：数字（文字コードはＪＩＳ－Ｘ０２０１（表１－１参照））

メッセージヘッダーのシーケンス№と同一の意味をもっており、メッセージヘッダーと設計画像データ・ヘッダーを区別せずに、１から１づつ昇順に付番する。

④ 番　号　（4 byte：数字（文字コードはＪＩＳ－Ｘ０２０１（表１－１参照））

同一のメッセージグループ内で、設計画像データとメッセージの論理的関係を示す『番号』で、同一の『番号』を持つ設計画像データとメッセージが論理的関係にあることを表す。

尚、メッセージ内では、タグ番号'ＥＦ００（＝６１１８４）'のＴＦＤが、この『番号』の項目になる。番号の値そのものは運用で決めるが、通常必要な範囲においてユニークでなければならない。

⑤ ファイル識別名（80byte：文字（文字コードはＪＩＳ－Ｘ０２０１（表１－１参照））

設計画像データをローカルに保存するときのファイル名で、左詰めにセットし余りはブランクで満たす。ファイル名そのものは運用で決めるが、通常必要な範囲においてユニークでなければならない。

⑥ ﾌｫｰﾏｯﾄ識別名　（32byte：文字（文字コードはＪＩＳ－Ｘ０２０１（表１－１参照））

設計画像データそのもの（ビット列データ）のﾌｫｰﾏｯﾄを表す識別名で、左詰めにセットし余りはブランクで満たす。識別名そのものは、運用で決める。

⑦ 圧縮識別名　（32byte：文字（文字コードはＪＩＳ－Ｘ０２０１（表１－１参照））

設計画像データそのもの（ビット列データ）のデータ圧縮方法を表す識別名で、左詰めにセットし余りはブランクで満たす。識別名そのものは、運用で決める。

５－３－２　ビット列データのフォーマット

ビット列データは設計画像データの本体である。ビット列ではあるが、伝送単位は８bit（１byte）単位である。２５１bit のような８の倍数でないビット列は扱わない。

ビット列データのフォーマットは、通常モード時と分割モード時で異なっており、通常モード時は３２００１byte、分割モード時は２５１byteの固定長フォーマットになっている。

図５－４にフォーマットを示す。尚、ビット列データそのもののフォーマットについては、本シンタックスルールで規定されない。運用で決定する。

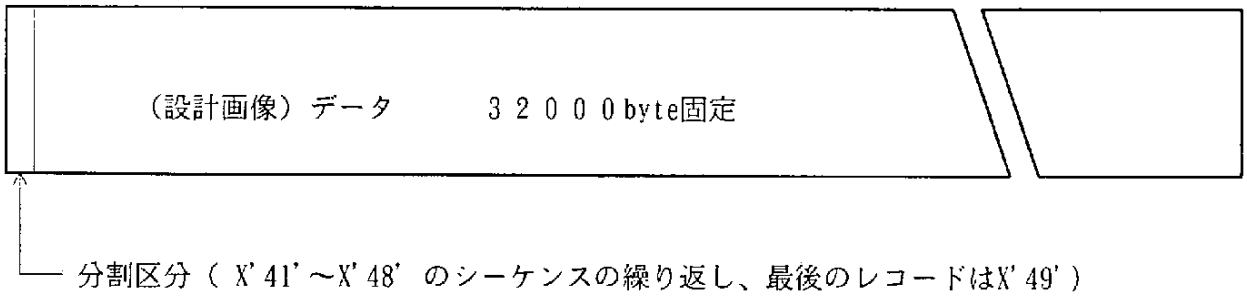
① 分割区分　　（1 byte）

ビット列データであることを示す。（X'41'～X'48' のシーケンスの繰り返し、最後のレコードは、X'49'）

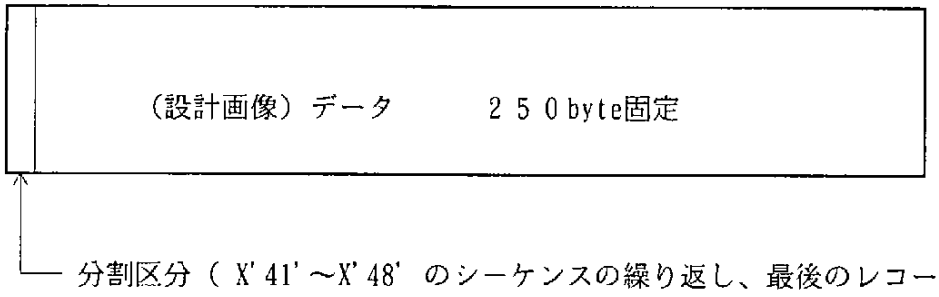
② データ

設計画像データそのものである。シンタックスルール上は、単純なビット列である。

通常モード時のビット列データのフォーマット：３２００１byte固定



分割モード時のビット列データのフォーマット　：２５１byte固定



最後のレコードのビット列データが２５０byteもしくは３２０００byteに満たない時

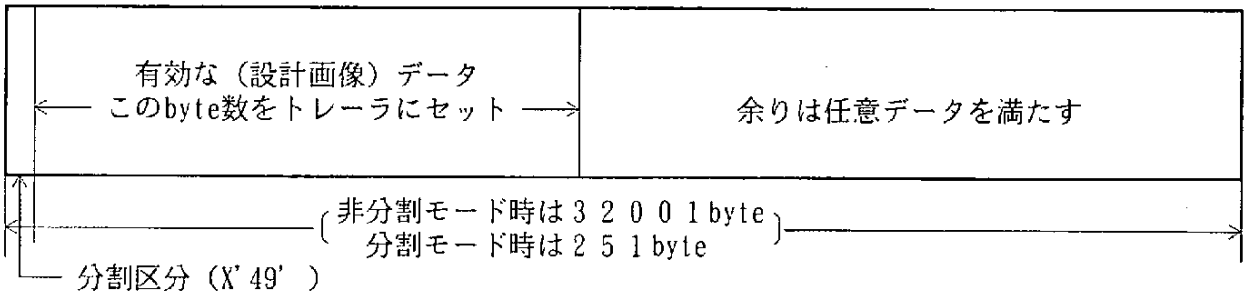


図５－４　ビット列データのフォーマット

５－３－３　設計画像データ・トレーラのフォーマット

設計画像データ・トレーラは、２５１byteの固定フォーマットで、図５－５の構造である。

① 分割区分　　（１byte：X'40'）

設計画像データのヘッダーかトレーラであることを表す。

② ﾍｯﾀﾞｰ表示　　（1 byte：X'54'）

設計画像データのトレーラであることを表す。

③ シーケンスNo.　（5 byte：数字（文字コードはＪＩＳ－Ｘ０２０１（表１－１参照））

設計画像データ・ヘッダーのシーケンスNo.と同一の番号とする。

④ 番　号　　（4 byte：数字（文字コードはＪＩＳ－Ｘ０２０１（表１－１参照））

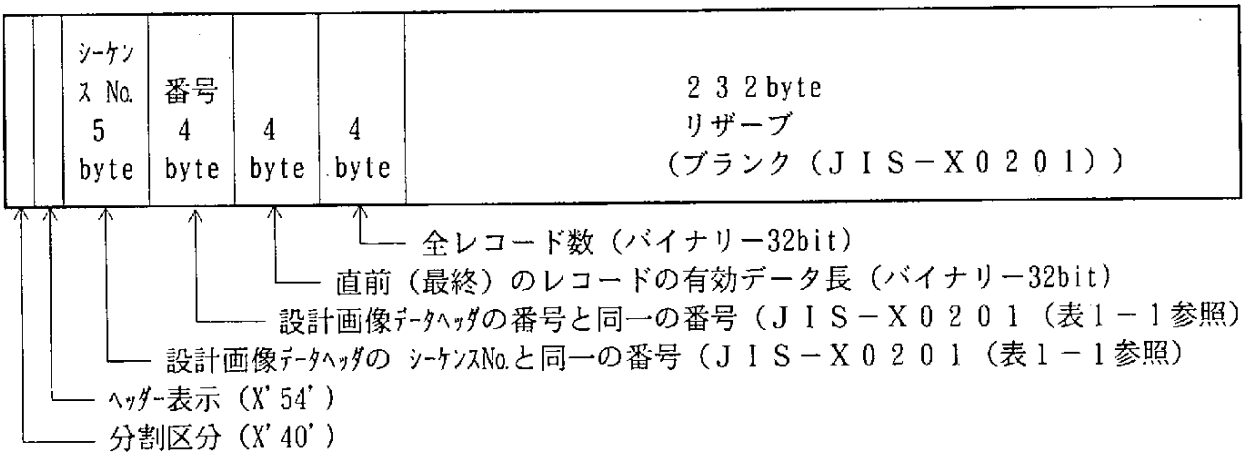
設計画像データ・ヘッダーの『番号』と同一の番号とする。

⑤ 直前（最終）のレコードの有効データ長（バイナリー32bit）

ビット列データは、格納構造にしたがって複数の固定長のレコードに分割して格納される。この場合、一般的に最後のレコードに端数のデータが発生する。有効データ長は、その端数の長さを表している。数値の１が１byteに相当し、以下、２が２byteに、ｎがｎbyteに相当する。

⑥ 全レコード数（バイナリー32bit）

設計画像データは、格納構造にしたがって複数の固定長のレコードに分割して格納する。全レコード数は、そのレコード数を表す（設計画像データ・ヘッダーと設計画像データ・トレーラを含む）。



注）全レコード数とは、一つの設計画像データのレコード数（ヘッダーとトレーラを含む）

図５－５　設計画像データ・トレーラのフォーマット

5－4　設計画像データの格納構造

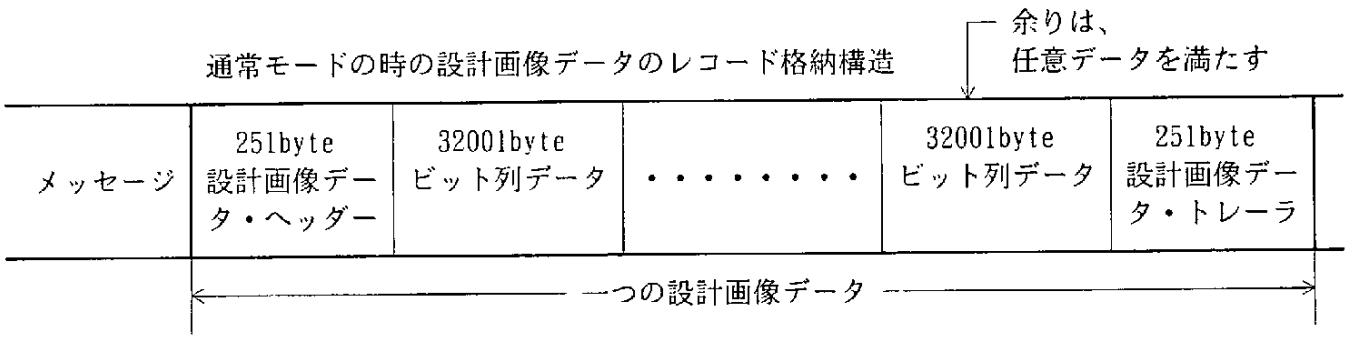
設計画像データは、通常モード時も分割モード時も固定長であり、下記のように格納する。（図5－6参照）

(1)　通常モード時

設計画像データ・ヘッダー（251byte）と設計画像データ・トレーラ（251byte）は、それぞれ一つの可変長レコードに格納する。そして、ビット列データは、32001byteの固定長データを一つの可変長レコードに格納する。

(2)　分割モード時

設計画像データ・ヘッダー（251byte）と設計画像データ・トレーラ（251byte）は、それぞれ一つの固定長レコードに格納する。そして、ビット列データは、251byteの固定長データを一つの固定長レコードに格納する。



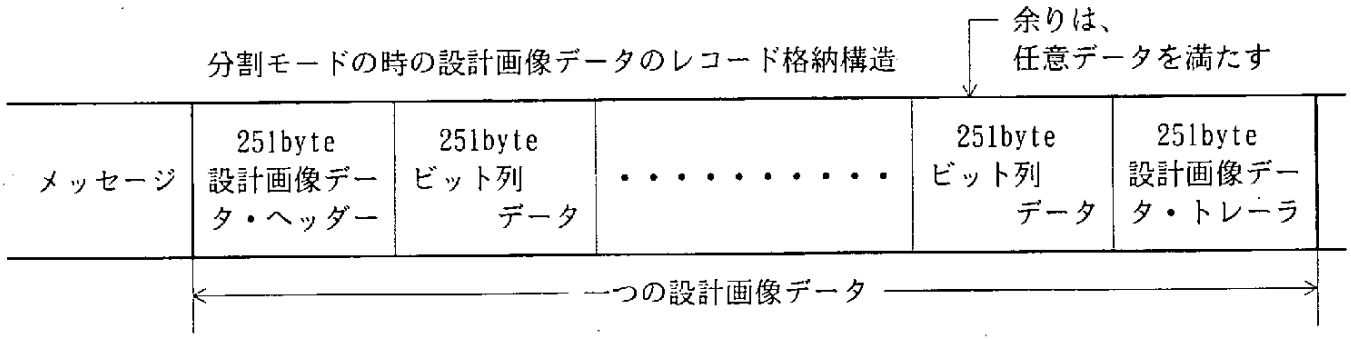


図5－6　設計画像データの格納構造

(3)　分割区分のシーケンス（各レコードの最初の1byte）

①　設計画像データ・ヘッダー …… X'40'

②　ビット列データ …… X'41'～X'48' の繰り返しシーケンス
最後のレコード（設計画像データ・トレーラ直前のレコード）は、X'49'

③　設計画像データ・トレーラ …… X'40'

5－5　設計画像ＥＤＩ専用データタグ番号

タグ番号『X’ＥＦ００’（＝６１１８４）』～『X’ＥＦ０Ｆ’（＝６１１９９）』までの１６種のＴＦＤを設計画像ＥＤＩ専用特殊ＴＦＤとして、リザーブする。これらの特殊ＴＦＤは設計画像データ指示子として、設計画像データ内ではなく、メッセージ内のＴＦＤ(データ項目)として用いる。この内、以下の８種について、その内容を規定する。残りは、将来の拡張に備えてのリザーブとする。

ＥＦ００	××	番　号	９属性 Max 4byte
ＥＦ０１	××	ファイル識別名	X属性 Max80byte
ＥＦ０２	××	ﾌｫｰﾏｯﾄ識別名	X属性 Max32byte
ＥＦ０３	××	圧縮識別名	X属性 Max32byte
ＥＦ０Ｃ	××	ファイル名メモ	X属性 Max 250byte
ＥＦ０Ｄ	××	ファイル名メモ	K属性 Max 250byte
ＥＦ０Ｅ	××	任意メッセージ	X属性 Max 250byte
ＥＦ０Ｆ	××	任意メッセージ	K属性 Max 250byte

① X’ＥＦ００’～ X’ＥＦ０Ｆ’はセットで用いる。但し、不要なものは省略できる。

② X’ＥＦ００’～ X’ＥＦ０Ｆ’はメッセージ中で用いる。

③　『番号（X’ＥＦ００’）』は、同一メッセージグループ内のｎ個の設計画像データを区別するともに、特定のメッセージに対応する設計画像データを特定する（図５－７）。

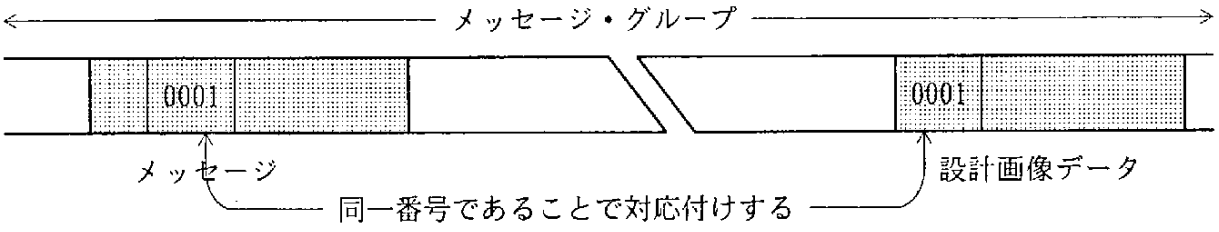


図５－７　番号によるメッセージと設計画像データのリンク
（同一メッセージグループ内の時、別々の時も考え方は同じ）

④　『ﾌｧｲﾙ識別名（X’ＥＦ０１’）』は③の『番号』で特定される設計画像データ・ヘッダー内の『ﾌｧｲﾙ識別名』と同一の値でなければならない。『ﾌｧｲﾙ識別名』は、通常は、受け側の社内システムで、メッセージと関連設計画像データファイルをリンクするために用いる。

⑤　『ﾌｫｰﾏｯﾄ識別名（X’ＥＦ０２’）』は、設計画像データそのもののフォーマットを表すデータコードである。③の『番号』で特定される設計画像データ・ヘッダー内の『ﾌｫｰﾏｯﾄ識別名』と同一の値でなければならない。このデータコードは登録方式とする必要があるが、本シンタックスルールでは規定しない。運用で決定する。

⑥　『圧縮識別名（X’ＥＦ０３’）』は、設計画像データそのものの圧縮方法を示すデータコードである。③の『番号』で特定される設計画像データ・ヘッダー内の『圧縮識別名』と同一の値でなければならない。このデータコードは『フォーマット識別名』と同じく登録方式とする必要があるが、本シンタックスルールでは規定しない。運用で決定する。

⑦　『ファイル名メモ（X’ＥＦ０Ｃ’はＸ属性、X’ＥＦ０Ｄ’はＫ属性）』は、ユーザーが使用する任意のメモである。このファイル名メモは、関連する設計・画像データヘッダーにはない。

⑧　『任意メッセージ（X’ＥＦ０Ｅ’はＸ属性、X’ＥＦ０Ｆ’はＫ属性）』は、ユーザーが使用する任意のメッセージである。このメッセージは、関連する設計・画像データヘッダーにはない。

⑨　複数組の設計画像データ指示子をメッセージ内で用いる時は、マルチ明細とする。

⑩　『番号』『ﾌｧｲﾙ識別名』『ﾌｫｰﾏｯﾄ識別名』『圧縮識別名』については、関連する（『番号でリンクされる）設計画像データ・ヘッダー内にも同一の値のデータがある。しかし、『ファイル名メモ』と『任意メッセージ』については、設計・画像データヘッダー内には、同一のデータはない。

5－6　補　足（トランスレーターにおける設計画像データ指示子の位置づけ）

5－6－1　送信側トランスレーター

① 変換テーブルには、設計画像特殊ＴＦＤが通常のデータ項目と同様に定義される。

② 変換テーブルで示されるローカルエリアには、設計画像データに関する情報が記述される。

③ 設計画像データは、一つのデータで一つのファイルを構成しているのが望ましい。この場合は、『ﾌｧｲﾙｱｸｾｽに関する記述』は、ファイル名のみでよい。そうでない場合は、ファイル名とアクセスキーの記述が必要になる。

④ 通常のトランスレーターでは『番号』を自動採番（昇順）する。『ﾌｧｲﾙ識別名』『ﾌｫｰﾏｯﾄ識別名』『圧縮識別名』『ファイル名メモ』『任意メッセージ』は、ユーザーが適当な値を設定しなければならない。

⑤ 変換テーブルには、メッセージパス機能が必要である。即ち、通常はメッセージと設計画像データ混在で生成するが（最小のメッセージは、設計画像特殊ＴＦＤのみとなる）、パス機能によりメッセージの作成を抑止する。

5－6－2　受信側トランスレーター

① 変換テーブルには、設計画像データ指示子が通常のデータ項目と同様に定義される。

② 変換テーブルで示されるローカルエリアには、設計画像データに関する情報が、受信メッセージ（設計・画像データ）から展開される。この情報は通常、メッセージの方から展開される。

③ 『ﾌｧｲﾙｱｸｾｽに関する記述』は、トランスレーターが自動発生し、ユーザー側にその値を通知するエリアとなる。

④ 『ﾌｧｲﾙｱｸｾｽに関する記述』は、受信した設計画像データを書き込むファイルを指示する。

⑤ トランスレーターが『ﾌｧｲﾙｱｸｾｽに関する記述』を自動発生する方法として、以下に示すような複数の方法の中から、ユーザーが選択できるようにする。

a．あらかじめ、ファイル名列をトランスレーターに与え、その中からトランスレーターが自動選択する。

b．プレフィックス（数文字）をトランスレーターに与え、トランスレーターがそれに４桁の番号を追加し、ファイル名とする。

c．ファイル名をトランスレーターに与え、アクセスキー（数字６桁程度）をトランスレーターが『０００００１』から昇順に付番する方式（メインフレーム向き）。

⑥ 設計画像データをファイルへ書き込むとき、メッセージ関係とは別のユーザーインタフェースエリアを用意しておき、ここへ、ハンドリングした設計画像データのヘッダー情報をおとす。

Ⅲ　附属資料

1．マルチ明細に関する補足

1－1　改行マークの省略

図９－１の上段と下段では、改行マークの省略以外は同一の構成であるが、意味は違っている。図９－１上段では、ＴＦＤの２番目と３番目が省略されたことを表すが、図９－１の下段では、ＴＦＤは１番目と２番目の２つしかないことを表している。すなわち、改行マークは通常省略できないことになる。しかしながら、図９－１の上段の構成で２番目と３番目のデータがなく、４番目のＴＦＤが２番目として認識されてもかまわない時は、図９－１の下段の構成にすることが可能である。

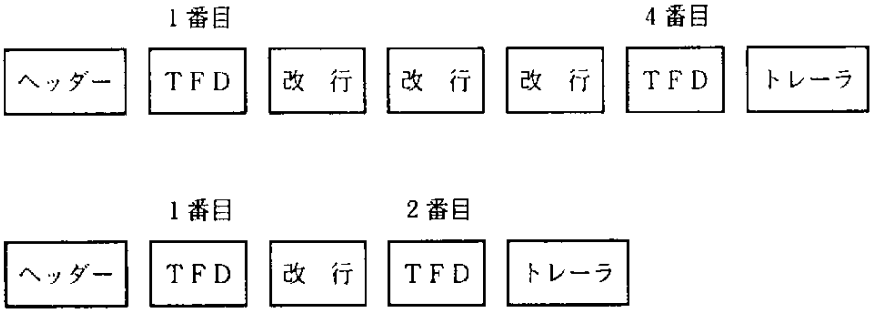


図９－１　マルチ明細における改行マークの意味

1－2　マルチ明細全体の省略

一つのマルチ明細全体が空の時は、そのマルチ明細全体を省略できる。ここでいうマルチ明細全体とは、マルチ明細ヘッダーからそれに対応するマルチ明細トレーラまでをいう（マルチ明細の次のトレーラではない）。また空という意味は、マルチ明細ヘッダーからそれに対応するマルチ明細トレーラまでの間に、ＴＦＤ制御子しかなかった場合である(有効データがない)。この一般則は、マルチ明細のネスティングがある時でも有効である。

省略可能であるというのは、省略しなければならないという意味ではない。受信用のトランスレーターでは、省略の有無にかかわらず正しくメッセージを解析できなければならない。

1－3　マルチ明細ヘッダーの明細番号の意味

拡張モードのマルチ明細ヘッダーには、明細番号がある。縮小モード(ＥＩＡＪ互換モード)では、明細番号がない。

図９－２に示すように、一つのメッセージ内に２つの独立したマルチ明細がある場合、縮小モードでは、マルチ明細内の適当なＴＦＤのタグ番号を用いて、マルチ明細の認識を行う。なぜなら、図９－２の左側のマルチ明細が省略された場合、右側のマルチ明細を左側のマルチ明

細として認識してしまう可能性があるからである。従って、右側のマルチ明細と左側のマルチ明細に同一のタグ番号を持つＴＦＤを入れてはいけない。

拡張モードでは、明細番号を用いてマルチ明細を区別するので、このような制限はない。明細番号は、同一のメッセージ内のマルチ明細を区別するために用いるので、同一のメッセージ内でユニークな番号を付ければよい。

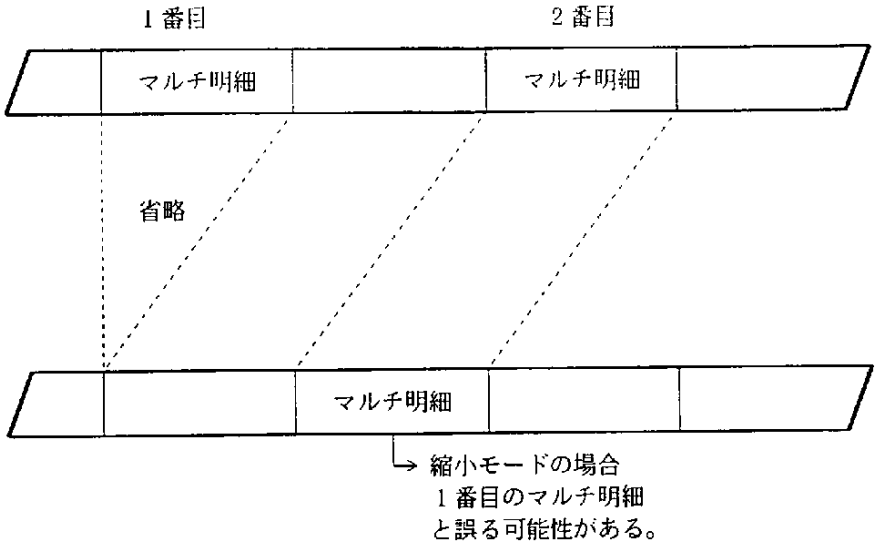


図９－２　マルチ明細全体の省略

２．外字とＪＩＳ補助漢字について

ＣＩＩシンタックスルール１．１０では、ＪＩＳ－Ｘ０２０１とＪＩＳ－Ｘ０２０８が標準として規定してある。実際の業務では、ＪＩＳ－Ｘ０２０８では規定されていない漢字も少なくない。そこで、外字の設定が必要になるが、外字の文字コードはコンピュータの機種によりまちまちでｍ：ｎを想定したＥＤＩのデータ交換には馴染まない。

そこで、外字の代わりにＪＩＳ－Ｘ０２１２（補助漢字）を導入する。ＪＩＳ－Ｘ０２１２と，ＪＩＳ－Ｘ０２０８は、同一の文字コードになっているので、そのまま混用することはできない。ＣＩＩシンタックスルール１．２０（近い将来改定）では、以下のようにして、ＪＩＳ－Ｘ０２１２を導入する。

ＪＩＳ－Ｘ０２１２（補助漢字）の第１byteのＭＳＢbit を１にする（第１byteにＸ'80'をＯＲする）。この方法は、非透過モードでも有効である。

３．ユニコードについて

ユニコードの導入は、ＣＩＩシンタックスルール１．３０以後になる。ユニコード導入時には、メッセージグループ・ヘッダーの１６bit 文字コード指定エリアにユニコードを表す識別

子『U(=X'55')』が、新たに追加になる。

4．ＥＩＡＪシンタックスルールとの互換性

ＣＩＩシンタックスルールはＥＩＡＪシンタックスルールとの互換性を可能なかぎり配慮してあるが、ＣＩＩトランスレーターとＥＩＡＪトランスレーターとを組み合わせる場合には、以下の点について考慮しなければならない。以下の制限を守れば、通常、ユーザーは特別な考慮なしにＣＩＩトランスレーターとＥＩＡＪトランスレーターとを混用できる。

(1) メッセージにおける制限（縮小モード）を用いる。

① N属性のデータ項目を使わない。

② ２４０以上のタグ番号（項目№）を持つデータ項目を使わない。

③ 文字コードは、標準を用いる（ＪＩＳ－Ｘ０２０１とＪＩＳ－Ｘ０２０８）。ＣＩＩトランスーターでは標準を指定することで、この条件が満足される。ＥＩＡＪトランスレーターで文字コード変換のサポートがない場合は、ユーザー側で変換する。

受信用のＣＩＩトランスレーターでは、ＥＩＡＪトランスレーターで作成されたメッセージは無条件で標準文字コードを使用していると判断する。

④ ２４１byte以上の長さを持つデータ項目を使用しない。

⑤ 明細番号付マルチ明細を使わない。

⑥ マルチ明細のネスティングを使わない。

⑦ 最大メッセージ長を 32767byte以下にする。

⑧ インターナルセグメントを使わない。

(2) メッセージグループ・ヘッダーに関する考慮事項

① トータル項目№を指定する時は、必ず、小数点以下が３桁の９属性のデータ項目を指定する。

(3) 通信系に関する考慮事項

① 透過モードを使用する（透過モードの通信システムを使用する）。

② 分割モードを使用する。非分割モードを使用すると、分割区分の分だけ（１byte）ずれを生ずる（ＣＩＩシンタックスルールでは、非分割モードの時も分割区分が付いているが、ＥＩＡＪシンタックスルールでは、非分割モードの時は、分割区分がない）。

(4) その他

ＥＩＡＪシンタックスルールの使用では、メッセージグループ・ヘッダーやトレーラも含めて、伝送電文全体の文字コード（バイナリー部分を除く）をＥＢＣＤＩＣコードとして運用している例がある。このケースでは、特別のＥＩＡＪ対応ＣＩＩトランスレーターでなければ、ＣＩＩシンタックスルールとの共通運用ができないので、注意を要する。

5．非透過モード・オプションについて

非透過モードは、Ｊ手順等の古い非透過モードの通信システムのために設けられたオプションで、可能なかぎり、全銀手順やＯＳＩ新手順でネットワークを構成し、非透過モードは使用しないことが望ましい。

6．ＣＩＩシンタックスルールのバージョン番号について

ＣＩＩシンタックスルールのバージョン番号は、３桁で表現されます。

① 一番右側の桁は、トランスレーターには影響を及ぼさない修正があった時、変更されます。主にシンタックスルールの矛盾の修正や本書（規格書）の記述の誤り修正の場合に変更されます。

② 中央の桁は、トランスレーターのマイナーチェンジが必要になるようなシンタックスルールの修正があった時、変更されます。

③ 一番左側の桁は、シンタックスルールの大変更があった時、変更されます。新しいバージョン対応のトランスレーターが必要になります。

下位２桁の変更では、新バージョンが旧バージョンを包含する方式をとります。しかしながら、上位１桁の変更では、必ずしも新バージョンが旧バージョンを包含しない場合もあります。

当センター（産業情報化推進センター）では、今後、図形・画像データを送れるＣＩＩシンタックスルールとインタラクティブＥＤＩ用のＣＩＩシンタックスルールを開発する予定ですが、前者は、下位２桁が変更され、後者は上位１桁のバージョン番号が変更されます。インタラクティブＥＤＩ用のＣＩＩシンタックスルールが、バッチ処理用ＣＩＩシンタックスルールを包含するか未定です。

インタラクティブＥＤＩ用シンタックスルールとバッチ処理用シンタックスルールが包含関係にならない場合、２つのバージョンのシンタックスルールが同時運用されることになります。

7．Ｙ属性追加について

7－1　目　　的

従来型のシステム設計では、年月日は、数字６桁で設計するのが普通であったが、最近、２０００年対策のために、数字８桁で設計するケースが増加している。

このことは、業界標準化における標準メッセージの設計にも影響し始めており、最近のメッセージ設計では、数字８桁とするのが多い。

６桁表示の年月日と８桁表示の年月日は、そのままでは互換性がないので、ユーザーは特別の対策が必要となるし、２０００年対策のためだけに、８桁表示が必要になるというのも通常の事務処理システムでは無駄が多い。歴史のように、数千年レンジの処理が必要な場合はとも

かくとして、事務処理システムのように高々数十年の処理で済む場合には、4桁表示は、いかにも冗長である。本規格の追加は、これらの問題の解決を目的としている。

7－2　基本的な考え方

現在１９９１年であるから、通常のシステムの寿命が２０年程度とすれば、これから２０年以内に構築するシステムでは、２０３０年まで有効な日付表現が設定されていれば、十分であると結論できる。

少し余裕を見て、２０５０年まで有効な日付表現を採用すれば、業務処理上の問題点はないとすることができる。一方、ＣＩＩシンタックスルールの寿命は、永遠であるが、２０５０年という６０年後以後も有効かどうかを議論することは、あまり意味がないだろう。コンピュータ実用化以来まだ４０年程の歴史しかないし、この間にも、何回かシステムの技術的体系は変化してきた。この面からも２０５０年まで有効な方式によりシステムを構築することは、妥当な選択と言える。

7－3　基本設計

日付表現の有効範囲を、１９５１年～２０５０年までとし、6桁表現と8桁表現の互換性を図る特別な変換をトランスレーターに設けることにより、6桁表現のシステムと8桁表現のシステム間のＥＤＩを実現するものである。

日付の6桁表現と8桁表現を以下のように、対応させる。

6桁表現		8桁表現
５１０１０１	←→	１９５１０１０１
９９１２３１	←→	１９９９１２３１
０００１０１	←→	２０００００１０１
５０１２３１	←→	２０５０１２３１

図９－３　6桁表現と8桁表現の対応

図9－3から、8桁から6桁への変換は、単純に、上位2桁をカットすればよいことが分かる。逆に6桁から8桁への変換では、

５１年～９９年の時（５１以上の時）……１９を追加する

００年～５０年の時（５０以下の時）……２０を追加する

のようにする必要のあることが分かる。

このような取扱をする属性をＹ属性として新たに定義し、シンタックスルールに追加する。

７－４　トランスレーターでの対応

７－４－１　送信用トランスレーター

Ｙ属性は基本的に数値（９属性の特別ケース）であるから、９属性と同一の変換を行う。

① ローカルエリアは一般的に、整数で、６桁か８桁でなければならない。

② ローカルエリアが２進表現の場合には、等価的に６桁か８桁でなければならない。

変換は、９属性と同一の変換である。したがって、送信側ではＹ属性を９属性で定義してもよい。

７－４－２　受信用トランスレーター

Ｙ属性は、基本的に数値（９属性の特別ケース）であるから同一の変換を行えばよいが、Y(6) と Y(8) の互換性を保つ処理が必要になる。

① ローカルエリアは一般的に、整数で、６桁か８桁でなければならない。

② ローカルエリアが２進表現の場合には、等価的に６桁か８桁でなければならない。

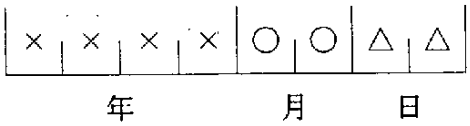
変換は、以下のように行う。

① 必要であれば、文字コードを変換する。

② 受信したＹ属性のデータは、８byteに変換する。この時８byteに満たないデータは、上位桁（左側）にゼロを追加して、８byteにする。

③ ②の処理の結果、年相当部分が、０００００～００５０の時は、上位１桁を２として、２０００～２０５０とする。年相当部分が、００５１～００９９の時は、上位２桁を１９として、１９５１～１９９９とする。

④ ③の処理の結果、ローカルエリアが６桁の時でも８桁の時でも、以下のような形になる。



⑤ ローカルエリアの形式に変換する。

- この時、ローカルエリアが６桁の時は、左側２桁をカットする。

７－４－３　その他のトランスレーターでの対策事項

(1) 拡張モード指示

拡張モード指示は、Ｙ変換の存在によって、特に影響を受けない。

相手側のトランスレーターがＥＩＡＪトランスレーターの場合やその他様々なケースでの互換性は、図９－４のようになる。

(2) エラーコードについて

『３６　：　不正日付の検出（Ｙ変換で検出）』が新たに追加された。

○印は可、×は互換性なし

	ＣＩＩ		ＥＩＡＪ		ＣＩＩ		ＣＩＩ
○	９（６）	→	９（６）	○	９（６）	→	９（６）
×	９（６）	→	９（８）	×	９（６）	→	９（８）
○	Ｙ（６）	→	９（６）	○	９（６）	→	Ｙ（６）
×	Ｙ（６）	→	９（８）	○	９（６）	→	Ｙ（８）
△注１	９（８）	→	９（６）	○	Ｙ（６）	→	９（６）
○	９（８）	→	９（８）	×	Ｙ（６）	→	９（８）
△注１	Ｙ（８）	→	９（６）	○	Ｙ（６）	→	Ｙ（６）
○	Ｙ（８）	→	９（８）	○	Ｙ（６）	→	Ｙ（８）
○	９（６）	←	９（６）	○注２	９（８）	→	９（６）
×	９（８）	←	９（６）	○	９（８）	→	９（８）
○	Ｙ（６）	←	９（６）	○	９（８）	→	Ｙ（６）
○	Ｙ（８）	←	９（６）	○	９（８）	→	Ｙ（８）
○注２	９（６）	←	９（８）	○注２	Ｙ（８）	→	９（６）
○	９（８）	←	９（８）	○	Ｙ（８）	→	９（８）
○	Ｙ（６）	←	９（８）	○	Ｙ（８）	→	Ｙ（６）
○	Ｙ（８）	←	９（８）	○	Ｙ（８）	→	Ｙ（８）

注１）　ＥＩＡＪトランスレーターに強制カットの機能があれば、○

注２）　強制カットを指定する。

注３）　９タイプは、いずれの場合も日付を表すとする。

図９－４　Ｙ属性をサポートしていないトランスレーターとの互換性