

Clinical Data Interchange Standards Consortium (CDISC : 臨床データ交換標準コンソーシアム)

オペレーショナルデータモデル (ODM) 仕様書

バージョン 1.3.1

ソースファイル : ODM1-3-1.htm

最終更新日 : 2010-02-11

Copyright © CDISC 2010 本仕様書の著作権は CDISC が所有する。本仕様書の使用は自由であり、(1) 修正を加えないこと、および (2) 複製には著作権に関する事項を全て記載すること。CDISC から書面による同意を得た場合のみ、本仕様書の修正が可能である。

本仕様書は CDISC のウェブサイト <http://www.cdisc.org/odm> の ODM ページ上で公開している。

レビュー期間 ; ライセンスの責務、表明および保証 ; 責任範囲および免責事項については [セクション 7.0](#) を参照のこと。

本書は ODM バージョン 1.3.1 の仕様書である。ODM バージョン 1.3.0 からの追加および変更の一覧は、[セクション 2.5 の過去バージョンからの変更](#)に記載している。変更はすべて下位互換性を有する (すなわち、ODM バージョン 1.2、1.2.1、または 1.3.0 のスキーマに準拠する ODM の XML ファイルは、ODM バージョン 1.3.1 のスキーマにも適合する)。

目次

1 はじめに (参考)

本オペレーショナルデータモデル (ODM) は、ベンダー中立でプラットフォームに依存しない臨床試験データの交換および保存のためのフォーマットである。本モデルは臨床データの他に、関連するメタデータ、管理データ、参照データ、および監査情報も含んでいる。初期設定、操作、解析、申請、または保存の一環としての長期保管のために使用する様々なソフトウェアシステム間で共有が必要な情報はすべて本モデルに含まれる。

本仕様書は ODM バージョン 1.3.1 である。新しい機能の要約については、[セクション 2.5 の過去バージョンからの変更](#)に記載している。新しい機能の詳細については、本仕様書の

本文内に統合して記載している。ODM バージョン 1.3.1 は、ほとんどの場合、本モデルの過去バージョンと下位互換性を保つように作成されている。下位互換性が影響を受けているところがあるのは、本標準の当初の意図に、より完全に一致させるために仕様を明確化したためである。

本モデルより前のバージョンでは、DTD（文書型定義）が含まれていた。バージョン 1.2 では、XML Schema と DTD が共に提供されていた。バージョン 1.3.0 以降のバージョンでは、XML Schema のみが利用可能である。

臨床データ管理システムでは、保存する情報および遵守すべき規則において、システム間の相違が著しい。現存するほとんどの臨床データ管理システムと互換性を保つために、本 ODM モデルは幅広い試験情報を記述できるように作成されている。本 ODM モデルで記述される一部の機能に対応していないシステムでも、[システムの適合性](#)に示している適合性規則に従う限り、ODM の互換性を保つことができる。

本 ODM は、臨床試験に用いられるコンピュータシステムに関して、米国食品医薬品局（FDA）より公表されているガイダンスおよび規定に準拠するよう作成されている。本書の目的は、ODM の公式仕様書であるとともに、本モデルを用いた臨床データの転送または保存に従事する利用者のユーザーガイドとなることである。

2 一般的な問題

2.1 本標準の内容

本文書に含まれる内容は CDISC の ODM 標準に関する仕様である。本標準に記載された内容のほとんどは、ODM ファイルを読み書きするシステムに対する要件を示したものである。ただし、注釈、使用上の注意、または例を記載しているところもある。そのような「非公式」な内容は、すべてパラグラフ（段落）またはセクション（節）を分け、**参考**、**注**、または**例**として強調表示している。

2.2 ファイルの適合性

本標準に詳述した基準をすべて満たす XML ファイルのみが ODM 1.3.1 標準に適合する。これらの基準には、構造上と意味上の 2 つの制約がある。

構造上の制約は以下のとおりである。

1. ODM ファイルは、well-formed（整形形式の）XML ファイルでなければならない。詳細については [XML 標準](#) を参照のこと。
2. ODM ファイルは、XML Namespace（名前空間）標準に適合しなければならない。詳細については [XML Namespace（名前空間）標準](#) を参照のこと。
3. ODM ファイルは、ODM 標準スキーマまたは有効なベンダー拡張スキーマにおいて定義された要素および属性のみを含むとともに、要素のネ스팅ならびに属性値および要素本体のフォーマットに関する規則を満たさなければならない。
4. ODM ファイルは、プロログと 1 つの（トップレベルの）[ODM 要素](#) を含むものでなければならない。
5. ODM ファイルは、"<http://www.cdisc.org/ns/odm/v1.3>" に掲載している ODM 1.3 の名前空間を使用しなければならない。
6. トップレベルの ODM 要素に対する ODMVersion 属性は、"1.3.1" に設定しなければならない。

ODM バージョン 1.3.1 のスキーマは、<http://www.cdisc.org/schemas/odm/v1.3> で確認することができる。本仕様書の最終バージョンは、<http://www.cdisc.org/odm> で掲載予定である。

プロログおよびトップレベルの ODM 要素の例：

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.3 ODM1-3-1.xsd"

  ODMVersion="1.3.1"
  FileOID="000-00-0000"
  FileType="Transactional"
  Description="Sample ItemData Extension"

  AsOfDateTime="2005-04-04T07:57:00"
  CreationDateTime="2005-04-06T10:30:00">
```

意味上の制約は、この後の本書全体にわたって記載している。

ODM バージョン 1.3 の XML Schema ファイルは、XML パーサー（構文解析ツール）により ODM ファイルが構造上において有効かどうかを検証することで使用可能である。意味上の正確性を確保するには、ODM 専用の検証アプリケーションの使用が推奨される。

2.3 システムの適合性

どのような情報標準でも言えるように、本 ODM 標準は、ODM ファイルが高いレベルの適合性を有していることを前提にシステムを開発できる場合に効力を発揮する。また一方で、本 ODM は過去には存在しなかった臨床データ交換のための標準と技術を同時に提供するという点でも価値がある。以下の適合性の記述に現れているのは、本標準に関して品質管理が必要なことと、本標準の採用と刷新を推進する際に柔軟性が必要なことを両立させようとしている点である。

ODM フォーマットで情報を処理するコンピュータシステムが本標準に適合していると言えるのは、以下の規則に従う場合に限る。

1. 生成された ODM ファイルは、本標準の構造上および意味上のいずれにおいても、その正確性の規則をすべて満たさなければならない。
2. 受信システムは、本標準の構造上および意味上のいずれにおいても、その正確性の規則をすべて満たす ODM ファイルであれば、読み込み可能でなければならない。
3. ODM ファイルは、ODM ルート要素内に明示された ODM バージョンの ODM スキーマに対して有効でなければならない。
4. 生成された ODM ファイルに含まれる情報は、本仕様書に定義した本標準の規則に照らして正確でなければならない。

5. 受信システムは、ODM ファイルから読み込んだ情報を、本仕様書に定義した本標準の規則に照らして正確に解釈しなければならない。
6. 生成される ODM ファイルには生成するシステムで、標準では処理または保存しない情報であれば、含める必要はない。
7. 受信システムは、標準では処理または保存しない情報であれば、ODM ファイルから読み込んだ情報を選択的に無視してもよい。
8. 受信システムは、処理または保存が可能なデータ値、キー、名前などの範囲を制限してもよい。
9. ODM の臨床データファイルを受信するが、標準ではセクション 2.14 に規定しているデータ型を 1 つ以上サポートしていないシステムは、サポートしていないデータ型は text (テキスト) 型として扱うべきである。
10. 全てのシステム制限 (規則 6~9) があれば、すべて記録しなければならない。
11. 適合性が特定のモードまたは設定に依存する場合も、記録しなければならない。

2.4 ベンダー拡張

本 ODM は、臨床試験のデータおよびメタデータの交換および保存のための標準フォーマットを提供することを目的としている。本モデルは多種多様な臨床データアプリケーションで使用可能なことが明らかにされているが、個別の臨床データシステムでは、本 ODM モデルではうまく表現できない情報を保持している例もある。このようなシステムに対してデータの交換および保存に本 ODM 標準を採用するように促し、臨床試験アプリケーションの相互運用性を向上させるには、本 ODM のベンダー拡張を利用することが推奨される。本 ODM モデルに対するベンダー拡張の要件は以下のとおりである。

1. ベンダーは、拡張した ODM フォーマットを完全に記述した XML Schema を提供しなければならない。
2. 拡張した ODM ファイルは、適切な拡張 XML Schema を参照すべきである。
3. ベンダー拡張では、新しい XML の要素および属性を追加してもよいが、標準 ODM の要素または属性を放棄してはならない。ODM の他の要素を用いて標準的に表現できる情報については、ベンダー拡張を使用することはできない。
4. 新しい要素および属性は、他のベンダー拡張と名前が重複することがないように、すべて異なった [XML 名前空間](#) を使用しなければならない。
5. 拡張した ODM ファイルからベンダー拡張をすべて取り除くと、意味のある正確な標準 ODM ファイルにならなければならない。
6. ベンダーは、要求に応じてベンダー拡張を取り除いた ODM ファイルを作成できなければならない。

拡張した ODM ファイルを使用するアプリケーションは、標準 ODM ファイルも受け入れなければならない。

注：一般に有用な情報を表現できるベンダー拡張は今後の標準化が考えられるため CDISC へ提供されるべきである。

注：個別のプログラムとして、あるいは標準入力処理の一部として、拡張 ODM ファイルから非標準の属性および要素をすべて取り除くフィルターを作成することが可能である。

受信システムは、拡張 ODM ファイルを受け取った場合、その処理を許可することが推奨される。

2.5 過去バージョンからの変更（参考）

ODM 1.3.1 は ODM 1.3.0 で定義された要素と属性をすべて含む。新たな機能を提供するために、新しい要素と属性を追加している。誤りを訂正し、明確化を図るとともに、様々なセクションで一貫性と明確性を高めるために軽微な修正を加えた。本 ODM バージョンでは、廃止予定となった要素または属性はない。

本ファイルをコピーし、ファイル先頭近くの ODM 1.3.1 スタイル定義の周辺にあるコメントタグを取り除くことで、バージョン 1.3.1 で行われた変更を強調表示した仕様書とすることができる。

2.5.1 エイリアスの使用拡張

以前のバージョンでエイリアス（別名）を記述できるのは、[ItemGroupDef](#) と [ItemDef](#) のみであった。ODM 1.3.1 では、本モデル全体にわたって [Alias](#) を追加し、今回、[Protocol](#)、[StudyEventDef](#)、[FormDef](#)、[CodeList](#)、[CodeListItem](#)、[EnumeratedItem](#)、[MethodDef](#)、[ConditionDef](#) でも記述できるようになっている。

2.5.2 Ref 要素の選択性

複数の [ItemRef](#) を含む [ItemGroupDef](#) のように、子の Ref を複数含む要素における Ref の有無は、今回任意となっている。以前のバージョンでは、Ref が少なくとも 1 つは必要であった。

2.5.3 データ型の追加

Datatype（データ型）として、[incompleteDatetime](#)（不完全日時）を補うために [incompleteDate](#)（不完全日付）および [incompleteTime](#)（不完全時刻）を追加している。

2.5.4 ReferenceData で AuditRecords、Signatures、Annotations をサポート

これらは、[untyped](#)（型なし）データに対しては以前から許されていたが、ODM 1.3.0 における [typed](#)（型付き）データの追加に伴い、これらをサポートするために異なった方式が追加された。この方式は [ClinicalData](#) に追加されたが、[ReferenceData](#) には追加されていなかった。ODM 1.3.1 では、この機能性を [ReferenceData](#) に拡張している。

2.5.5 明確化

[ItemDefLength](#) 属性および [SignificantDigits](#) 属性の記述が意味および用途を明確にするために更新している。

[MetaDataVersion](#) の [Include](#) の機能については、さらに詳細に、例を加えて説明している。

複数の [TranslatedText](#) 要素および [xml:lang](#) 属性の用途を明確化している。

2.6 entity（構成要素）および要素

本 ODM モデルでは、[study](#)（試験）の臨床データが数種類の [entity](#)（構成要素）から構成されることを前提としている。このような [entity](#)（構成要素）としては、[subject](#)（被験者）、[study event](#)（試験事象）、[form](#)（フォーム）、[item group](#)（項目群）、[item](#)（項目）、および [annotation](#)（注釈）がある。

[item](#)（項目）は、収縮期血圧の 1 回の測定値といった個々の臨床データ項目である。複数

の *item* (項目) を集めたものが *item group* (項目群) である。

item group (項目群) は、密接に関係した *item* (項目) のセットで、通常は一緒に解析される。(*item group* [項目群] は "record" [レコード] と呼ばれることもあり、"panel" [パネル] または "table" [表] と関係がある。) *item group* (項目群) の集合体が *form* (フォーム) である。

form (フォーム) は、紙 CRF (症例報告書) または電子的症例報告書画面の 1 ページに類似している。*form* (フォーム) は、一般的に論理的かつ時間的に関連する情報セットを集めたものである。一連の *form* (フォーム) は *study event* (試験事象) の一部として集められる。

study event (試験事象) は、複数の *form* (フォーム) の再利用可能なものをまとめたもので、通常は *study* (試験) のデータ収集事象に相当する。

試験実施計画書デザインでは、データを収集する予定の *subject* (被験者) の *visit* (ビジット) を 0 回またはそれ以上含む場合もある。各計画された *visit* (ビジット) が ODM の 1 つ以上の *study event* (試験事象) に相当する場合もあれば、ODM の *study event* (試験事象) が複数回の *subject* (被験者) の *visit* (ビジット) で使用される場合もある。

subject (被験者) は、*study* (試験) に参加している患者である。

annotation (注釈) は、*subject* (被験者)、*study event* (試験事象)、*form* (フォーム)、*item group* (項目群)、または *item* (項目) に対して付けたコメントである。*entity* (構成要素) のペアに対して、*annotation* (注釈) を付けることもできる。

study (試験) のメタデータは、その *study* (試験) において許される *study event* (試験事象)、*form* (フォーム)、*item group* (項目群)、および *item* (項目) の様式を記述したものである。メタデータの *entity* (構成要素) には *StudyEventDef*、*FormDef*、*ItemGroupDef*、および *ItemDef* が含まれる。

StudyEventDef では、(主に含めることができる *form* [フォーム] の様式を列記することにより) 詳細な *study event* (試験事象) の様式を記述する。

FormDef では、詳細な *form* (フォーム) の様式を記述する。

ItemGroupDef では、詳細な *item group* (項目群) の様式を記述する。

ItemDef では、詳細な *item* (項目) の様式を記述する。

study (試験) の臨床データには、各 *StudyEventDef* に相当する実際の様々な *study event* (試験事象)、各 *FormDef* に相当する現実の様々な *form* (フォーム) などを概して含んでいる。

ODM ファイルは (すべての XML ファイルと同様に) *element* (要素) のツリーで構成されている。ODM ファイル内の臨床データ要素は、臨床の *entity* (構成要素) の状態またはその *entity* (構成要素) の状態に対する変更のいずれかを表している。これらの要素には、*SubjectData*、*StudyEventData*、*FormData*、*ItemGroupData*、*ItemData*、および *Annotation* がある。

このような要素は、どのデータの *entity* (構成要素) に関する情報を記述しているかを識別するキー属性をそれぞれ含んでいる。多くのデータ要素が 1 つのデータの *entity* (構成要素) に対応することもよくある。このようなことが生じる可能性があるのは、1 つの *entity* (構成要素) に対して一連の更新が適用される場合、または監査証跡が表示される場合である。

同様に、メタデータの *entity* (構成要素) に対応している XML 要素がある。

注：すべての臨床試験がデータを visit (ビジット) または form (フォーム) にまとまるわけではない。そのような study (試験) では、subject (被験者) 当たり 1 つの共通な study event (試験事象)、および/または study event (試験事象) 当たり 1 つの共通な form (フォーム) とするのが妥当である。

2.7 臨床データキー

本 ODM には、内部キーおよび外部キーの 2 種類のキーがある。内部キーは、本モデル内の entity (構成要素) を指定し、ODM ファイル内 (および ODM ファイルにまたがる) entity (構成要素) 間の相互参照を可能にするために使用する。内部キーは人が使うためのものではなく、変化しないことが前提となっている。内部キーとしては、subject key (被験者キー)、element OID (要素 OID)、および repeat key (リピートキー) がある。

臨床データの entity (構成要素) を完全に識別するには、以下の内部キーが必要となる。

entity (構成要素) の種類	識別キー
study (試験)	StudyOID 属性
subject (被験者)	上記 + SubjectKey 属性
study event (試験事象)	上記 + StudyEventOID 属性および StudyEventRepeatKey 属性
form (フォーム)	上記 + FormOID 属性および FormRepeatKey 属性
item group (項目群)	上記 + ItemGroupOID 属性および ItemGroupRepeatKey 属性
item (項目)	上記 + ItemOID 属性
annotation (注釈)	注釈付き entity (構成要素) に対するキー + SeqNum 属性

StudyOID 属性により、1 件の study (試験) を一意に識別する。SubjectKey 属性により、1 件の study (試験) 内の 1 人の subject (被験者) を一意に識別する。複数の study (試験) にわたって 1 人の subject (被験者) を識別するために、SubjectKey 属性を使用することはできない。

StudyEventOID 属性により、1 件の study (試験) 内の 1 つの StudyEventDef を一意に識別する。ただし、特定の subject (被験者) に関して特定のタイプの study event (試験事象) がいくつか存在する場合がある。そのため、特定の study event (試験事象) を完全に識別するには、その StudyOID 属性、SubjectKey 属性、StudyEventOID 属性、および StudyEventRepeatKey 属性が必要になる。

各 form (フォーム) は、1 つの study event (試験事象) に属し、FormOID 属性 (form [フォーム] のタイプを設定する) および FormRepeatKey 属性により (その study event [試験事象] 内で) 識別することができる。

臨床データの各 item group (項目群) は、1 つの form (フォーム) に属し、ItemGroupOID 属性 (item group [項目群] のタイプを設定する) および ItemGroupRepeatKey 属性により (その form [フォーム] 内で) 識別することができる。

その他の item group (項目群) は参照データで、単に ItemGroupOID 属性および ItemGroupRepeatKey 属性によって識別可能である。

各 item (項目) は、1 つの item group (項目群) に属する。ただし、1 つの item group (項目群) 内で 1 つの item (項目) が参照されるのは 1 回のみと思われる。そのため、1 つの item (項目) をそれが属する item group (項目群) 内で一意に識別するには、ItemOID 属性のみでよい。

外部キーは、臨床関係者が使用する任意のキーである。このような外部キーには、被験者ランダム化コード、試験実施施設コードなどがある。本 ODM では、外部キーをあたかも臨床データであるかのように (すなわち、今回定義している ItemData[TYPE] の 1 つを用いて) 記述してはいる。そのため、外部キーは必要に応じて変更可能で、通常の監査プロセスの対象となる。

2.8 単一ファイルおよびその集合

ODM 文書 (ファイル) は、ソースデータベースとの関連付けが可能なように対象となるデータベースを修正する方法を示した説明文とみなすことができる。ソースデータベースに関するすべての情報、つまり全 subject (被験者)、全メタデータ、全臨床データ、および臨床データの各 item (項目) に対する全変更をすべて 1 つの ODM 文書に含めることも実際には可能である。しかし、そのような大きな文書を作成しても、常に使い勝手がよいとは限らない。そのため、このような情報を一連の ODM ファイルで徐々に提示していくことも認めている。

各 ODM 文書には FileOID 属性 ([ODM 要素](#)を参照) がある。この FileOID 属性によって、その文書を一意に識別すべきである。

ODM 文書では PriorFileOID 属性も設定可能である。この属性には、一連のファイル中、直前のファイルの FileOID 属性を記述する。このように、PriorFileOID 属性によって関連付けられたファイルの集合を利用して、ソースデータベースを徐々に追記していくことが可能である。(当然、一連の文書中の最初の文書には PriorFileOID 属性がない。)

一連のファイル中、1 つのファイル内のデータが、その前のファイルに含まれる定義に準拠することが許される。ファイル内に ReferenceData または ClinicalData のいずれかが含まれていれば、そのデータをそのまま解釈するのに必要なメタデータ定義をそのファイル内に含むか、あるいはこのような定義を含む前のファイルが一連のファイル中になければならない。同様に、ファイル内に ClinicalData が含まれていれば、監査および署名のレコードで参照される管理データをそのファイル内に含むか、あるいはそのデータを含む前のファイルが一連のファイル中になければならない。

ツリー構造になった ODM ファイル一式 (その場合、複数のファイルが同一の前ファイルを参照する) も許される。

複数の試験に関する情報を 1 つの ODM 文書 (またはその一式) に含めることもできる。審議を簡素化するために、あたかも試験が 1 件であるかのように記述しようとすることもある。1 つのファイル (またはファイルの一式) に複数の試験を含める場合は、各試験に対して以降に記載した規則が個々に適用されるべきである。

注 : FileOID 属性により、その文書内容を識別する。この属性は、その文書が複写されても

変ってはいけない。**注**：FileOID 属性は、可能な限り普遍的に一意とすべきである。これを確実に行う方法の 1 つは、その ODM ファイルまたはデータベースの作成者が所有するインターネットドメインネームを（最後に"/"を付加して）すべての FileOID 属性の先頭に付与することである。例えば、Best Pharmaceuticals 社による試験 5894 に関する一連のファイル中、最初のファイルを示すよい方法として、FileOID="BestPharmaceuticals.com/Study5894/1"とすることが考えられる。

注：同様に、StudyOID 属性も可能な限り普遍的に一意とすべきである。例えば、StudyOID="BestPharmaceuticals.com/Study5894"などが考えられる。

注：アプリケーションによっては、受け取った ODM 文書を保存し、後でその FileOID 属性に基づいて検索する必要がある。特に、一連のファイル中、新しいファイルを解明するには、前のファイルとその FileOID 属性によって見つける必要がある。この為に簡単な方法の 1 つは、その文章自体のファイル名の一部として、その FileOID 属性の（場合により変換された）値を使用することである。例えば、上述の ODM 文書では、...¥BestPharmaceuticals.com¥Study5894¥1.xml として保存することが考えられる。

その他のファイル属性

前述したように、ほとんどの ODM 文書は、ソースデータベースが保有する（現在および過去の）全情報の一部しか含んでいない。

所定の文書で発信される情報は、様々な範囲により異なる可能性がある。文書の内容の例をいくつか挙げる。

- メタデータのみ、
- 臨床データのみ、
- ドメインデータのみ（SDTM：試験データ表形式モデル）、
- 1 人の subject（被験者）に関する 1 つの値の一連の変更、
- 試験全体の現在の状態、
- 試験のサブセット（特定の subject [被験者]、特定の form [フォーム] など）の現在の状態、
- 最終報告の試験への変更、または
- 試験で発生したすべての変更履歴。

内容にこのような違いがあるため、文書の利用者がそれから読み取れるように、文書自体の説明を記述することも重要である。このような要求に応えるため、ODM 要素に文書自体を説明する属性をいくつか備えている。

CreationDateTime 属性は、その ODM 文書が作成された日時を示す。対照的に、AsOfDateTime 属性は、その文章の内容が確認された日時を示す。データベースへの変更を理解するために一連のファイルが使用される場合、特に重要である。

FileType 属性および Granularity 属性を使用することで、時間およびデータの全体にわたって特定の文書が対象としている範囲を文書の送り側が定義できる。Archive 属性を使用することで、その文書の内容が FDA 21 CFR 11 規則において定義される電子記録に適する基準の仕様を満たしていると送り側が主張できる。最後に、Description 属性は、その文書の説明で他の属性を補う為に、必要に応じて詳細に記述するテキスト文字列を提供する。このセクションでは、これらの属性に設定できる値を詳細に示し、単一の文書および一連の文書においてそれらの属性の値について説明している。

FileType 属性

ODM 文書の FileType 属性は、Snapshot (スナップショット) 型または Transactional (トランザクション) 型のいずれかでなくてはならない。Snapshot (スナップショット) 型の文書は、含まれているデータに関してソースデータベースの現在の状態をどのように再現するかを示しているが、過去に遡ってどのようにその状態になったのかは示してはいない。Transactional (トランザクション) 型の文書は、各 entity (構成要素) について、ソースデータベースの最新の状態、およびソースデータベース内の (任意の) いくつか前の状態の両方を示している。Snapshot (スナップショット) 型の文書では、TransactionType 属性は不要である。Transactional (トランザクション) 型の文書进行处理する際に、1 つのデータ点に関してトランザクションのセットを順序付けする規則は、[要素の順序付け](#)のセクションに記載している。

Granularity 属性

ODM 文書の Granularity 属性は、その文書に記述された情報の範囲を特定の共通文書型として記述する簡易表記を送信側に提供することを目的としている。以下に、これらのカテゴリーが意図している内容を示す。

カテゴリー	文書の内容
All (すべて)	任意およびすべてのタイプのデータおよびメタデータ
Metadata (メタデータ)	メタデータのみ
AdminData (管理データ)	管理データのみ
ReferenceData (参照データ)	参照データのみ
AllClinicalData (全臨床データ)	臨床データのみ
SingleSite (単一実施施設)	単一実施施設の臨床データのみ
SingleSubject (単一被験者)	1 人の被験者の臨床データのみ

これらの簡易表記のカテゴリーで不十分な場合は、その文書の Description 属性を利用して詳細を記述する。

Archival 属性

Archival 属性は任意である。

Archival=Yes は、このファイル (またはファイル一式) が 21 CFR 11 で定義される電子記録

の要件を満たすように作成されていることとなる。より厳密に言えば、そのファイル（またはファイルのセット）では、データ値の挿入、更新、および削除の利用可能であれば全ての監査および署名の情報を伴った、完全かつ非冗長なセット、いでなければならない。

Archival=Yes とするには以下が必要である。

- 現在のファイルは **Transactional**（トランザクション）型でなくてはならない。
- 現在のファイルがあるファイル一式の中の 1 つであれば、その中の他のファイルもすべて **Archival** 型で、かつ **Transactional**（トランザクション）型でなくてはならない。
- トランザクションのセットは、そのファイル内またはファイル一式内で完全かつ非冗長なセットでなくてはならない。特に、各ファイルの内容は以下でなければならない。
 - その **PriorFile**（前ファイル）（もしあれば）の **AsOfDateTime** 属性の日時より前の日付けの監査記録日があるトランザクションを含まない、
 - 全てのトランザクションはそのファイルの **AsOfDateTime** 属性の日時までとする（**Granularity=All** の場合、この「すべて」はその試験におけるすべてのトランザクションを意味する。他の **Granularity** 属性に対して「すべて」の意味については [Granularity 属性](#)を参照のこと）、
 - ソースデータベースで利用可能な **AuditRecord**（監査記録）および **Signature**（署名）の情報をすべて含む、かつ
 - **Upsert**（更新挿入）のトランザクションは含まない。

2.9 トランザクション

各データ要素には、設定が任意の **TransactionType** 属性がある。この属性には、**Insert**（挿入）、**Update**（更新）、**Remove**（削除）、**Upsert**（更新挿入）、または **Context**（コンテキスト）の 1 つが設定可能である。

Insert（挿入）の **TransactionType** 属性は、そのデータの **entity**（構成要素）が新しい（その試験では現在存在していない）ため、得られたすべてのプロパティとともに試験データに追加されなければならないことを意味している。そのデータの **entity**（構成要素）が既に存在している場合、またはそのデータの **entity**（構成要素）の親が存在していない場合、この設定はエラーとなる。

Update（更新）の **TransactionType** 属性は、そのデータの **entity**（構成要素）が既に存在していて、得られた新しいプロパティを含めるために修正しなければならないことを意味している。関係の無いプロパティは修正されない。データの **entity**（構成要素）が存在しない場合、この設定はエラーとなる。

Remove（削除）の **TransactionType** 属性は、そのデータの **entity**（構成要素）が既に存在していて、そのプロパティと子の要素とともにすべて削除しなければならないことを意味している。データの **entity**（構成要素）が存在しない場合、この設定はエラーとなる。

注：このセクションにおける"delete"（削除）という用語の使用は、そのデータの **entity**（構成要素）のレコードすべてが全くそん存在しないことを意味しているわけではなく、その **entity**（構成要素）に対してその後にはトランザクションが（もしあれば）**Update**（更新）または **Remove**（削除）ではなく、**Insert**（挿入）でなければならないことのみを意味している。

Upsert（更新挿入）の `TransactionType` 属性は、そのデータの `entity`（構成要素）が存在していれば `Update`（更新）と同じで、それ以外では `Insert`（挿入）と同じである。

注：Upsert（更新挿入）を使用することで、送信側はそのデータの `entity`（構成要素）が存在するかどうか確認しなくて済む。（この機能は特定のデータ収集システムで必要とされる可能性がある。）この機能は潜在的に危険であり、本標準の将来バージョンでは削除される可能性がある。

`Context`（コンテキスト）の `TransactionType` 属性は、そのデータがコンテキストを伝える目的のみの為に明示的に再送信されていることを意味している。この例として、送信側と受信側のシステム間で `Patient ID`（患者識別子）のマッチングおよびチェックが可能のように、その都度（`Context` を `TransactionType` として）、被験者背景データを含む `ItemGroup`（項目群）を送信することが考えられる。

自身の `TransactionType` 属性を設定していない子の要素は、その親の `TransactionType` 属性を引き継ぐ。トップレベルのデータ要素では、`TransactionType` 属性を明示しなければならない。

トランザクションの処理は、その ODM 文書内でデータ要素が出現する順序で進める。ODM 文書内に 1 つの `entity`（構成要素）に対するトランザクションが 2 つ含まれる場合は、出現する順序で適用する。

`TransactionType` 属性を `Remove`（削除）とした `entity`（構成要素）が子孫を含むことがある。そのような子孫もそれぞれ `TransactionType` 属性を `Remove`（削除）としなければならないが、そうしない場合は `TransactionType` 属性を設定してはならない（その場合、子孫は親の `TransactionType` 属性を引き継ぐ）。ODM ファイルを処理するプログラムは、要素がこの規則に違反していないか、要素を処理する前の段階で判定できるように、検討しておくべきである。このような子孫に対応するデータが、親の削除に起因する連携より削除されている可能性があることに注意する。そのような場合、処理プログラムが削除された子孫のデータの `entity`（構成要素）に対してそれぞれ監査証跡を生成するかどうかは任意である。

`Snapshot`（スナップショット）型文書に `TransactionType` 属性を設定する場合は、`TransactionType=Insert` のみが許される。

2.10 要素の順序付け

ODM ファイルのデータ部分において 1 つの `entity`（構成要素）に適用する要素は、時間的順序で出現しなければならない。つまり、各 `entity`（構成要素）に対するトランザクションは、その `entity`（構成要素）の要素がファイル内で発生する順に適用され、`Signatures` および `AuditRecords` は、その `DateTimeStamp` の値が増加していなければならない。

一般に、異なるエンティティに対する要素は、お互いに関連した順になっている必要はない。ただし、`entity`（構成要素）が署名される場合、その `entity`（構成要素）に関して署名前に出現したすべてのトランザクションは、同じ ODM ファイル内、または関連した一連のファイル中の前の ODM ファイル内で、先に存在していなければならない。

ファイル内のすべての `DateTimeStamp` の日時は、そのファイルの `CreationDateTime` 属性の日時より前でなければならない。

関連した一連の ODM ファイルの中で、ファイルの `AsOfDateTime` 属性は、時間順に出現しなければならない。ファイルの `DateTimeStamp` の日時は、それぞれ前のファイルの `AsOfDateTime` 属性の日時より後でなければならない。

データの entity (構成要素) は (form [フォーム] のように)、一連のデータ要素として記述することができる。このようなデータ要素の内容には、その entity (構成要素) を分離した部分 (プロパティおよび子) のデータ、その同じ部分に対する連続した変更、またはその 2 つを混在させて記述することが可能である。

study (試験) における臨床データの初期値またはデフォルト値は、すべて NULL (ヌル) 値とみなされる。そのため、Snapshot (スナップショット) 型の送信では、NULL (ヌル) 値の item (項目) を含める必要がない。一連の item (項目) の insertion (挿入) および update (更新) でも、最初の値として NULL (ヌル) 値を含める必要がない。

これらのファイルでは、ODM ファイルを生成するシステムは、要素の数を最小限に抑えるようにすべきである。そのため、記述した事象の意味を変えずに (かつ、順序規則に違反しないで)、同じ entity (構成要素) に対する 2 つの要素を 1 つにまとめることが可能な場合は、1 つにまとめた様式が推奨される。同様に、不必要な要素は省略すべきである。

2.11 要素の識別子と参照

本 ODM では、1 つの要素が、同一ファイル内、および一連の ODM 文書内のファイルにわたって他の要素の参照を必要とするインスタンスが多くある。これを実現するために、目的とする要素に一意的「名前」(その OID) を与える。目的とする要素への参照を必要とする要素はすべて、その OID を利用するだけでよい。

例:「収縮期血圧 (SBP)」と呼ぶ item (項目) の定義 (ItemDef) があるとする。この ItemDef の OID 属性を"SBP"に設定する。その結果、収縮期血圧データの item (項目) の特定のインスタンス (ItemData) があれば、それ自身の ItemOID 属性を"SBP"に設定することによって、元の定義へ結び付けることができる。

使用する OID は一意でなければならないが、特定の範囲内だけでよい。一意性の規則は以下のとおりである。

- MetaDataVersion 内に含まれる各要素型 (例えば FormDef) に対する OID は、その MetaDataVersion 要素の範囲内で一意でなければならない。
 - その MetaDataVersion 全体の中で一意の OID とするのが最もよいが、例えば、FormDef を ItemGroupDef と同じ OID とするなど、必要であれば異なった要素型に対して同じ OID を使用することも許される。ただし、推奨はしない。
 - MetaDataVersion 内の子要素は、Included (インクルードされた) MetaDataVersion 要素内の子要素と同じ OID としてもよい。MetaDataVersion 内の子要素が、Included (インクルードされた) MetaDataVersion 内の同じタイプの子要素と OID が同じ場合は、先の定義を置き換えているとみなされる。 [3.1.1.3.1 Include 要素](#)を参照のこと。
- MetaDataVersion の OID は、それを含む Study 内で一意でなければならない。
- ArchiveLayout の OID は、1 つの FormDef 内で一意でなければならない。ArchiveLayout の OID は、FormDef が異なれば再度使用してもよい。
- Study の OID は、それを含む ODM 内で一意でなければならない。
- MeasurementUnit の OID は、それを含む Study 内で一意でなければならない。
- User、Location、および Signature の OID は、それを含む AdminData 内で一意でなければならない。

ある要素が他の要素を参照するために OID を使用する場合は、参照される要素は参照する

要素と同じ文書内、**または**関連する一連の文書中の前の文書に必ず存在しなければならない。従って、PriorFileOID 属性のリンクを介して遡ることによって、参照する要素を必ず検出することができる。

注:OID 参照では複数のファイルを横断する必要があるので、標準 XML の ID および IDREF の値 (1 つのファイル内でのみ参照される) の使用を避けている。本標準の要素定義では、ID および IDREF の代わりに oid および oidref を使用することによって、このことを明確にしている。

注: 上記の規則は以下のことを示している。

- 現在の文書内、またはその一連の文書中の前の文書内のいずれかで、OID の値を定義していない限り、OID 値への参照を送信することはできない。
- OID により識別される定義を変更した場合、その定義の変更は、変更した定義が参照される前に送信するか、あるいはその変更された定義が参照される文書に存在しなければならない。
- 一連の文書中、最初の文書は、それが参照する定義をすべて含んでいなければならない。
- Transactional (トランザクション) 型の文書は、新しい定義、またはその前の文書と比べて変更された定義に関わる要素を含んでいなければならない。例えば、文書 B3207 が item (項目) の定義 X への参照を含み、B3207 の前の文書以降にコードリストの制約が X に追加されていたとする。その場合、B3207 は、item (項目) X の更新された定義を含む (OID が新しい) MetaDataVersion 要素も含んでいなければならない。

2.12 構文表記法

各 XML 要素の構文は以下のように規定する。

要素名

本体 :

内容の規定

属性 : *属性名、データフォーマット、選択制、意味上の詳細*

...

内容の規定は、"EMPTY (空)"、データフォーマット名、または要素グループとすることができる。EMPTY (空) は、この要素には本体 (ネスト化された内容) がないことを意味する。データフォーマット名は、その名前のフォーマットに一致する任意のテキスト文字列を本体に記述できることを意味する。要素グループは、その名前の要素をこの要素内で直接ネスト化できることを意味する。

要素グループは、1 つ以上の要素名 (または要素グループ) をコンマまたは縦棒で区切り、括弧で囲んで示す。コンマは、その要素 (または要素グループ) が、そのグループに記載された順序で XML 文書内に連続して出現しなければならないことを示す。縦棒は、その要素 (または要素グループ) の中の 1 つだけが出現しなければならないことを示す。要素または要素グループの後に、? (出現が任意)、* (出現しないか、1 回以上は出現する)、または+ (1 回以上は出現する) を付加している場合もある。

属性は、1 行に 1 つの属性を表形式で示し、属性名、データフォーマット、選択性、および場合により意味上の詳細を記載している。選択性の欄に"*optional*"(*任意*)が記載されていない限り、その属性は必ず必要である。本 ODM の基本スキーマは、XML Schema を用いた仕様を実装している。

2.13 データフォーマット

XML のすべての属性値および一部の要素本体はテキスト文字列である。これらの文字列はデータフォーマットで定義される。各データフォーマットでは、文字列の許される書式、対応する XML Schema のデータ型、およびその値の意図された用法を規定している。ODM データフォーマットの一覧を以下に示す。

フォーマット名	XML Schema のデータ型	許される文字列パターン
integer (整数)	xs:integer	-?digit+
positiveInteger (正の整数)	xs:positiveInteger	+?digit+ (かつ 0 より大の整数を表記)
nonNegativeInteger (非負整数)	xs:nonNegativeInteger	+?digit+ (かつ 0 以上の整数を表記)
float (浮動小数点数)	xs:decimal	-?digit+(.digit+)?
date (日付)	xs:date	YYYY-MM-DD
time (時刻)	xs:time	hh:mm:ss(.n+)?(((+ -)hh:mm) Z)?
datetime (日時)	xs:dateTime	YYYY-MM-DD T hh:mm:ss(.n+)?(((+ -)hh:mm) Z)?
text (テキスト)	xs:string	任意の文字列
oid	xs:string	任意の文字列 (minLength="1")
oidref	xs:string	任意の文字列 (minLength="1")
ID	xs:ID	任意の文字列 (minLength="1")
IDREF	xs:IDREF	任意の文字列 (minLength="1")
subjectKey (被験者キー)	xs:string	任意の文字列 (minLength="1")
repeatKey (リピートキー)	xs:string	任意の文字列 (minLength="1")
name (名前)	xs:string	任意の文字列 (minLength="1")

sasName (SAS 名)	xs:string	(letter _)(letter digit _)* (maxLength="8")
sasFormat (SAS フォーマット)	xs:string	(letter _ \$)(letter digit _ .)* (maxLength="8")
fileName (ファイル名)	xs:anyURI	任意の文字列
languageTag (言語タグ)	xs:language	LL (-CC)* (下記参照)
string (文字列)	xs:string	意味上では text(テキスト)と同じだが、XML Schema のデータ型として直接サポートされている
boolean (真偽値)	xs:boolean	(true false 1 0)
double (倍精度実数)	xs:string	((([+-]?)?[0-9]+([.][0-9]+)?)([Dd Ee]([+-]?)?[0-9]+)? (-?INF) (NaN))
hexBinary (16 進数)	xs:hexBinary	16 進数でエンコードしたバイナリ数列のデータ
base64Binary (Base64 進数)	xs:base64Binary	Base64 アルファベットによりエンコードされたバイナリデータ
hexFloat (16 進浮動小数点数)	xs:hexBinary	最大で 16 文字まで
base64Float (Base64 進浮動小数 点数)	xs:base64Binary	最大で 12 文字まで
partialDate (部分日付)	xs:date	[YYYY[-MM[-DD]]]
partialTime (部分時刻)	xs:time	[hh[:mm[:ss(.n+)?((+ -)hh:mm) Z]?]]
partialDatetime (部分日時)	xs:dateTime	[YYYY[-MM[-DD[T hh[:mm[:ss(.n+)?((+ -)hh:mm)?]]]]]
intervalDatetime (日時間隔)	xs:string	partialDatetime/partialDatetime) (durationDatetime/partialDatetime) (partialDatetime/durationDatetime)
durationDatetime (経過日時)	xs:duration	((([+-]?)?P(((n(n+)?)Y)?((nn+)?)M)?((nn+)?)D)?(T(((n(n+)?)H)?((n(n+)?)M)?((n(n+)?)([+-]?)S)?)? (((n(n+)?)W))))

<code>incompleteDatetime</code> (不完全日時)	<code>xs:string</code>	<code>[YYYY -]-[MM -]-[DD -]]T[hh -]:[mm -]:[ss.s -][?+(-)nn:nn Z]</code>
<code>incompleteDate</code> (不完全日付)	<code>xs:string</code>	<code>[YYYY -]-[MM -]-[DD -]</code>
<code>incompleteTime</code> (不完全時刻)	<code>xs:string</code>	<code>T[hh -]:[mm -]:[ss.s -][?+(-)nn:nn Z]</code>
<code>URI</code>	<code>xs:anyURI</code>	

数値は 10 進法で表す。float (浮動小数点数) では少数部分が許される。

注: 特定の整数属性 (KeySequence 属性、OrderNumber 属性、および SeqNum 属性) は、関連する entity (構成要素) 中の順序を定義するために使用する。これらの属性では、1 から始まる連続した整数値を使用するよう推奨する。

text (テキスト) の値は、XML 文書化の範囲内で有効な任意の文字または文字列である。テキスト文字列は ODM ファイルに組み込む場合、標準の XML 引用規則および文字エスケープ規則を用いて表記しなければならない。

date (日付) はグレゴリオ暦で表記する。年 (YYYY) は 0001~9999、月 (MM) は 01~12、日 (DD) は 01~31 の範囲とする。ODM の *date* (日付) は XML Schema のデータ型の **date** と同じである。

time (時刻) は 24 時間制で表記する。時 (hh) は 00~23、分 (mm) および秒 (ss) は 00~59 の範囲とする。記述が任意の秒の少数部分 (.nnn) では、少数点以下の秒を表記する。*timezone* (タイムゾーン) は、±hh:mm のフォーマット、または "Z" フォーマット (UTC : 協定世界時) のいずれかを用いて表記可能である。

datetime (日時) は、*date* (日付)、*time* (時刻)、および (任意の) *timezone* (タイムゾーン) を組み合わせたものである。

date (日付)、*time* (時刻)、および *datetime* (日時) は、データの収集場所で測定された現地時間と解釈しなければならない。*datetime* (日時) における ±hh:mm は、世界標準時に加算 (または減算) する時および分の補正值で、これによりデータ収集時の現地時間が得られる。*timezone* (タイムゾーン) の指定が無い場合は、現地時間と世界時との間の関係が不明なことを意味している。

例: シカゴ (グリニッジから西方向へ 6 時間のタイムゾーンで標準時使用) における 2001 年 1 月 3 日午後 3 時 14 分は、"2001-01-03T15:14:00-06:00" と表記する。シカゴ (夏時間使用) における 2001 年 7 月 20 日朝 0 時過ぎの 3.5 秒は、"2001-07-20T00:00:03.500-05:00" と表記する。

注: 上記の *date* (日付)、*time* (時刻)、および *datetime* (日時) のフォーマットは、[ISO 8601](#) ("Complete Representations" 「完全表記」を参照) と互換性がある。本 ODM の *date* (日付)、*time* (時刻)、および *datetime* (日時) のデータフォーマットは、日時成分をすべて含んでいなければならない。

partialDate (部分日付)、*partialTime* (部分時刻)、および *partialDatetime* (部分日時) のフォーマットは、ISO8601 ("Representations other than complete" 「完全表記以外の表記」を参照) と互換性がある。これらのフォーマットは、データ取得時にそれほど重要でない日時

成分が 1 つ以上不明な場合でも、そのデータの送信をサポートするために提供される。例えば、`partialDate`（部分日付）では、その月の日付が不明でもよい。

`intervalDatetime`（日時間隔）のフォーマットでは、時間間隔の始まりと終わりを示す `partialDatetime`（部分日時）のペア値、始まりを示す `partialDatetime`（部分日時）と `durationDatetime`（経過日時）の値、または `durationDatetime`（経過日時）と終わりを示す `partialDatetime`（部分日時）の値をそれぞれ斜線（/）で区切って表記する（それぞれ、"Representation of time-interval identified by its start and its end"「開始時と終了時により識別される時間間隔の表記」、"Representation of time-interval identified by its start and its duration"「開始時と経過時間により識別される時間間隔の表記」、および"Representation of time-interval identified by its duration and its end"「経過時間と終了時により識別される時間間隔の表記」を参照）。

`durationDatetime`（経過日時）のデータフォーマットでは、ISO 8601 の経過時間（"Representation of time-interval by duration only"「経過時間のみによる時間間隔の表記」を参照）で表記する。例えば、4 時間 35 分は `PT4H35M` と表記する。本 ODM の実装では、個々の日時成分のキャリーオーバーおよび負の経過時間をサポートしている。

`incompleteDatetime`（不完全日時）のフォーマットを使用することで、日時成分（必ずしも重要性が低いとはいえない）が 1 つ以上欠けている場合でも日時の送信が可能となる。完全な `datetime`（日時）のように区切り文字はすべて使用し、日時成分が欠けているところは 1 個のダッシュ記号（-）で表記する（"Truncated Representations"「不完全な表記」を参照）。本 ODM の実装では、最も重要な `partialDatetime`（部分日時）の書式もサポートしている。例えば、2004 年の不明月の 15 日で、不明時の 5 分は、`2004---15T:05` と表記する。

`incompleteDate`（不完全日付）のフォーマットを使用することで、日付成分（必ずしも重要性が低いとはいえない）が 1 つ以上欠けている場合でも日付の送信が可能となる。完全な `date`（日付）のように区切り文字はすべて使用し、日付成分が欠けているところは 1 個のダッシュ記号（-）で表記する（"Truncated Representations"「不完全な表記」を参照）。例えば、`2001---30` は、2001 年の不明月の 30 日を意味し、`---30` は不明年の不明月の 30 日を意味する。

`incompleteTime`（不完全時刻）のフォーマットを使用することで、時刻成分（必ずしも重要性が低いとはいえない）が 1 つ以上欠けている場合でも時刻の送信が可能となる。完全な `time`（時刻）のように区切り文字はすべて使用し、時刻成分が欠けているところは単一のダッシュ記号（-）で表記する（"Truncated Representations"「不完全な表記」を参照）。例えば、`-:55:30` は不明時の 55 分 30 秒を意味し、`-:-:30` は不明時不明分の 30 秒を意味する。

`hexFloat`（16 進浮動小数点数）および `base64Float`（Base64 進浮動小数点数）のデータフォーマットは、機械またはプラットフォームに依存しない方式で、浮動小数点数のデータを正確に交換することを目的としたバイナリ形式である。この形式は、SAS V5 移送フォーマットで用いられた IBM メインフレームのフォーマットを基にしている。このフォーマットは、SAS 技術サポート文書の["TS-140 The Record Layout of a Data set in SAS Transport \(XPORT\) Format"](#)に記載されており、値を `hexBinary` フォーマットまたは `base64Binary`（MIME）表記のいずれかに変える直接変換に適している。この二次変換により XML エンコードの互換性が確保される。

注：`hexFloat` または `Base64Float` のフォーマットで転送されるデータ値は、XML マークアップ言語でも、標準の XML オーサリング・ツールを使用しても人間が読みやすいものとはいえない。このため、このようなデータ型の採用を決定すると、このフォーマットで受け取ったデータを処理するためにアプリケーションが必要になることが想定される。

hexFloat (16 進浮動小数点数) および hexBinary (16 進数) のデータ値は、2 つの 16 進文字 ([0-9A-F]) からなる文字のタプル (組) としてエンコードされる。

base64Binary (Base64 進数) のデータ値は、[Base64 アルファベット](#) の 65 個の文字を用いてエンコードされる。これらの文字には、[\[XML 1.0 \(第 2 版\)\]](#) で定義されている a-z、A-Z、0-9、+、/、=、およびホワイトスペースが含まれる。

hexBinary (16 進数) および base64Binary (Base64 進数) のデータ型も浮動小数点数の交換に使用されるが、プラットフォーム依存またはアプリケーション依存を前提としていることがある。この場合、受信側のアプリケーションが受信した値を float (浮動小数点数) ではなく、string (文字列) として解釈しているかどうか判定できなくなる可能性があることに、実装者は注意すべきである。これらのデータ型を用いてエンコードされた値は、類似したハードウェアのプラットフォーム間でしか確実に交換されない可能性がある。

注 : ODM 1.1 では、「タイムゾーン情報なし」を指定するために、-99:99 のタイムゾーン仕様を認めていた。しかし、このフォーマットはもはやサポートしていない。

oid (OID) は、要素を一意に識別する文字列である (詳細については[要素の識別子と参照](#)を参照のこと)。oidref は、(その OID の決定的な出現とは対照的に) 要素を参照するために OID を使用することである。oid および oidref は、空文字列としない。

subjectKey (被験者キー) または repeatKey (リピートキー) は、臨床データの entity (構成要素) の指定に有用である ([臨床データキー](#)を参照のこと)。subjectKey (被験者キー) は、空文字列としない。

name (名前) は、一部の entity (構成要素) の名前を人間に読みやすくすることを目的としている。name (名前) は、空文字列としない。

sasName (または sasFormat) は、SAS バージョン 5 移送フォーマットで有効な任意の名前 (またはフォーマット) である。文字数は 8 文字以内とする。

fileName (ファイル名) は、1 つのファイルを指定する。ファイル名はシステムに依存し、処理している ODM ファイルを含むディレクトリーを基準として表記する。fileName (ファイル名) は、空文字列としない。

languageTag (言語タグ) は、自然言語またはその国特有の方言を表す。これに許される値は、[IETF RFC 3066](#) の [Tags for the Identification of Languages \(言語識別のためのタグ\)](#) で規定されている。languageTag (言語タグ) は NULL (ヌル) 値にできないことに注意すること。

例 : "fr-CA" はフランス語のカナダ方言を意味する。languageTag (言語タグ) の使用法に関する説明は、[TranslatedText](#) を参照のこと。

一般に、NULL (ヌル) 値が許されるデータフォーマットで NULL (ヌル) の属性値を表記するためには、空文字列 (例えば、attribute-name="") を使用すべきである。要素本体内で転送されるデータが NULL (ヌル) であることを表記するには、空要素 (例えば、<element-name/>) としてその要素を送信する。不明値が存在する場合と、ある値を NULL (ヌル) 値と置き換えたい場合とを区別するために、臨床データの識別子として特殊な IsNull 属性がある。[ItemData 要素](#) の IsNull 属性を参照のこと。

列挙型の定義については ODM 基本スキーマを、また ODM 臨床データフォーマットのそれぞれの正値および不正値の詳細例については [ODM Example XML](#) ファイルを参照のこと。

2.14 ItemData[TYPE] 要素

ODM 1.3.0 以降、本 ODM で臨床データの item(項目)値を転送するために使用する ItemData 要素として 2 種類が認められている。untyped (型なし) と typed (型付き) の要素型である。ItemData 要素は過去のすべての ODM バージョンにもあるが、untyped (型なし) データの転送に使用する要素型である。この ItemData[TYPE] 要素は typed (型付き) データの転送に使用する要素型である。

バージョン 1.3.1 の ODM スキーマには、以下の ItemData[TYPE] 要素がある。

ItemDataAny

ItemDataString

ItemDataInteger

ItemDataFloat

ItemDataDate

ItemDataTime

ItemDataDatetime

ItemDataBoolean

ItemDataDouble

ItemDataHexBinary

ItemDataBase64Binary

ItemDataHexFloat

ItemDataBase64Float

ItemDataPartialDate

ItemDataPartialTime

ItemDataPartialDatetime

ItemDataIntervalDatetime

ItemDataIncompleteDatetime

ItemDataIncompleteDate

ItemDataIncompleteTime

ItemDataURI

typed (型付き) データ値の要素はすべて、その要素の PCDATA コンテンツとしてデータ値を持つ。例えば、ItemDataInteger 要素では以下のようなになる。

```
<ItemDataInteger ItemOID="ID.114">100</ItemDataInteger>
```

この ItemData[TYPE] 要素を使用することで、ODM の XML ファイルを受け取ったアプリケーションがその ODM ファイルを構文解析する際に ClinicalData を検証できるようになる。

ItemDataAny 要素は、対応する ItemDef の DataType 属性値と異なる臨床データ値を転送するための回避方式として提供されている。受信側のアプリケーションは、ItemDataAny 要素を用いて転送されたデータ値を、対応するデータベースフィールドにロードする必要がある。

ない。

1つのODMファイル内でtyped（型付き）とuntyped（型なし）のデータが混在してはならない。

注：untyped（型なし）データの転送では、'、"、<、>、&などの特殊文字は避けなければならない。typed（型付き）データの転送では、これらの特殊文字をCDATAブロック内で転送できるが、それ以外では避ける。

```
<ItemDataString ItemOID="ID.115">
<![CDATA[
    The five special characters that must normally be escaped are ' , " , < , > , &
    ]]>
</ItemDataString>
```

ODM ファイルを生成または受信するシステムは、typed（型付き）データの転送をサポートすることが推奨される。

ODM 臨床データファイルを受信するが、このセクションに記載したデータ型を1つでも標準でサポートしていないシステムは、サポートしていないデータ型の臨床データをテキストとして受け付けるべきである。このようなシステムが、サポートしていないデータ型を含む新しいODM ファイルを生成して臨床データを送信する場合は、生成したファイルに元のデータ型の情報を含めなければならない。

例えば、標準ではデータ型としてhexBinary（16進数）またはhexFloat（16進浮動小数点数）をサポートしていないシステムが、データ型を'hexFloat'と定義したItem（項目）のデータを含むODMのXMLファイルを受け取った場合、そのItem（項目）の臨床データをText（テキスト）として保存してもよい。そのシステムがそのItem（項目）のデータを含むODMファイルを生成する場合でも、ItemDef要素はやはりhexFloat（6進浮動小数点数）としてデータ型を定義しなければならない。

3 一般的要素

3.1 ODM

本体：

([Study*](#), [AdminData*](#), [ReferenceData*](#), [ClinicalData*](#), [Association*](#), [ds:Signature*](#))

属性：

Description	text	(任意)	受信側にとって文書の正確な解釈に役立つ情報があれば、送信側はこのDescription属性に記録しておくべきである。
FileType	(Snapshot Transactional)		Snapshot（スナップショット）型は、文書に含まれる内容がデータおよびメタデータの現在の状態のみで、トランザクション履歴を含んでいないことを意味する。Snapshot（スナップショット）

ト) 型の文書では、データ点当たり 1 つのみ、命令が含まれる可能性がある。臨床データの場合、**Snapshot** (スナップショット) 型のファイルでは **TransactionType** 属性を記述しないか、あるいは **Insert** (挿入) に設定しなければならない。**Transactional** (トランザクション) 型は、データ点当たり 2 つ以上の命令が文書に含まれる可能性があることを意味する。データの現在の状態およびそれがどのようにして得られたかを送信するには、**Transactional** (トランザクション) 型の文書を使用すること。

Granularity	(All Metadata AdminData ReferenceData AllClinicalData SingleSite SingleSubject)	(任意)	<p>Granularity 属性は、文書における情報の範囲を特定の共通文書型として記述する簡易表記法を送信側に提供することを目的としている。All (すべて) は試験全体、Metadata (メタデータ) は MetaDataVersion 要素、AdminData (管理データ) および ReferenceData (参照データ) はそれぞれに対応する要素を示しており、AllClinicalData (全臨床データ)、SingleSite (単一実施施設)、および SingleSubject (単一被験者) は、試験の臨床データの中で、より細かく対象を絞り込んでいったサブセットである。これらの簡易表記分類で不十分な場合は、Description 属性を使用して詳細に記述すること。</p>
Archival	(Yes No)	(任意)	<p>ファイルが 21 CFR 11 で定義される電子記録の要件を満たすように作成されていることを示すには、この属性を Yes に設定する。ODM の他の要素の属性との相互関係に加え、この属性の意味に関するより完全な説明については、単一ファイルとその集合を参照のこと。</p> <p>この属性は FileType 属性が Transactional (トランザクション) 型の場合のみ使用すること。</p>
FileOID	oid		このファイルに対する一意の識別子。
CreationDateTime	datetime		この文書を含むファイルの作成日時。
PriorFileOID	oidref	(任意)	一連のファイル中、前のファイル (もしあれば) への参照。
AsOfDateTime	datetime	(任意)	この文書を作成するためにソースデータベースを検索した日時。

ODMVersion	(1.2 1.2.1 1.3 1.3.1)	(任意)	使用した ODM 標準のバージョン。
Originator	text	(任意)	この ODM ファイルを作成した組織名。
SourceSystem	text	(任意)	このファイル内の情報の元になるコンピュータシステム名またはデータベース管理システム名。
SourceSystemVersion	text	(任意)	上記の SourceSystem 属性に記述したシステムのバージョン。
ID	ID	(任意)	この ODM 要素を逆に参照するために、ds:Signature 要素により使用する場合もある。

AsOfDateTime 属性は、現在のファイルに含まれる新規データの発生日時または変更データの最終更新日時である。AsOfDateTime 属性を明示する必要はないが、省略した場合は CreationDateTime 属性の日時と等しいとみなされる。CreationDateTime 属性より後の日時を AsOfDateTime 属性に設定するとエラーとなる。

ODM 1.1 では ODMVersion 属性が定義されていないため、ODMVersion 属性が記述されていない場合は、ODM バージョンが"1.1"と解釈すべきである。

ODM 1.3.1 に基づいた文書は、このバージョンの新しい機能を利用するために ODMVersion="1.3.1"と設定しなければならない。下位互換性を保つために、ODM 1.2.0 を基にした文書は ODMVersion="1.2"、ODM 1.3.0 を基にした文書は ODMVersion="1.3"と設定しなければならない。

これらの ODM 要素の属性に関しては、[単一ファイルとその集合](#)のセクションで詳しく説明している。

3.1.1 Study

本体：

([GlobalVariables](#), [BasicDefinitions](#)?, [MetaDataVersion](#)*)

属性：

OID [oid](#)

親：

[ODM](#)

この Study 要素では、個々の study (試験) に関する静的な構造上の情報をまとめて記述する。

3.1.1.1 GlobalVariables

本体：

([StudyName](#), [StudyDescription](#), [ProtocolName](#))

属性：

NONE (無し)

親 :

[Study](#)

この GlobalVariables では、その study (試験) に関する一般的な要約情報を記述する。

3.1.1.1.1 StudyName

本体 :

[name](#)

属性 :

NONE (無し)

親 :

[GlobalVariables](#)

この StudyName では、その study (試験) の外部名を短く記述する。

3.1.1.1.2 StudyDescription

本体 :

[text](#)

属性 :

NONE (無し)

親 :

[GlobalVariables](#)

この StudyDescription では、その study (試験) に関する説明を自由に記述する。

3.1.1.1.3 ProtocolName

本体 :

[name](#)

属性 :

NONE (無し)

親 :

[GlobalVariables](#)

この ProtocolName では、sponsor (試験依頼者) が内部で使用する試験実施計画書名を記述する。

3.1.1.2 BasicDefinitions

本体 :

[\(MeasurementUnit*\)](#)

属性：

NONE (無し)

親：

[Study](#)

3.1.1.2.1 MeasurementUnit

本体：

[\(Symbol, Alias*\)](#)

属性：

OID [oid](#)

Name [text](#)

親：

[BasicDefinitions](#)

データ item (項目) つまりデータ値に用いる物理的な測定単位を記述する。MeasurementUnit の意味は、その Name 属性で決まる。例として、キログラム (kg)、センチメートル (cm)、細胞/ミリリットル (cells/mL) などが挙げられる。

注：測定単位名を標準化することは、本 ODM の範疇外である。

3.1.1.2.1.1 Symbol

本体：

[\(TranslatedText+\)](#)

属性：

NONE (無し)

親：

[MeasurementUnit](#)

測定単位として人間が読みやすい名前を記述する。

3.1.1.2.1.1.1 TranslatedText 要素

本体：

[text](#)

属性：

xml:lang [languageTag](#) (任意)

親：

[Decode](#), [ErrorMessage](#), [Question](#), [Symbol](#), [Description](#)

特定の言語に対応した人間が読みやすいテキストを記述する。TranslatedText 要素は連続して記述することが多く、異なった言語に翻訳した代替テキスト一式を提供する。

目的の言語に対応したテキストを言語タグにより検出するには、xml:lang 属性が言語タグに（大文字小文字を区別しないで）正確に一致する TranslatedText 要素を検索する。それで検出できなかった場合は、言語タグから最後のサブタグを取り除いて、再度検索する。それでも検出できなかった場合は、xml:lang 属性のない TranslatedText 要素を検索して、それを使用する。それでも検出できなければ、適切なテキストは得られない。以下に例を示す。

TranslatedText の記述例	求める言語	検索手順
<TranslatedText xml:lang="fr-CA">... <TranslatedText xml:lang="en-GB">... <TranslatedText>...	fr-FR	xml:lang="fr-FR"を検索。 これが検出されなければ、xml:lang="fr"を検索。 これも検出できなければ、xml:lang 属性がない TranslatedText を検索。それが使用すべきテキストと なる。

曖昧さを避けるために、一連の TranslatedText 要素における特定の言語タグの出現回数は 1 回のみとしなければならない。同様に、同じ親内で xml:lang 属性のない TranslatedText 要素が複数出現することも許されない。

注：xml:lang 属性は XML 標準の一部である。

3.1.1.3 メタデータの要素

試験のメタデータは、一連の MetaDataVersion 要素内で定義する。この方式（複数の MetaDataVersion 要素）を通じて、本モデルは"mid-study changes"（試験中の変更）の追加配布をサポートしているため、メタデータで複数のバージョンを同時に使用している状況（様々な実施施設で IRB 承認に遅れがあることが原因と考えられる）に対処できる。

メタデータのバージョンングの機能は次のとおりである。個々の Study 要素内に複数の MetaDataVersion を記述することも可能で、それは試験デザインの途中変更が 1 回以上あったことを示している。最初のバージョンでは全てのメタデータを記述する。その後の各バージョンに記述する内容は、主に、修正または新規追加したメタデータ要素のみで、明示的な参照によって前のメタデータを引き継ぐ。バージョンが異なっても同じメタデータ要素では同じ OID を使用する。この方式を用いることで、旧バージョンのメタデータをそのまま使用できると同時に、変更を表す簡潔な方法が得られる。

3.1.1.3 MetaDataVersion

本体：

[Include?](#), [Protocol?](#), [StudyEventDef*](#), [FormDef*](#), [ItemGroupDef*](#), [ItemDef*](#), [CodeList*](#), [ImputationMethod*](#) [廃止予定], [Presentation*](#), [ConditionDef*](#), [MethodDef*](#)

属性：

OID	oid	
Name	name	
Description	text	(任意)

親：

[Study](#)

MetaDataVersion では、study (試験) のデータを構成する study event (試験事象)、form (フォーム)、item group (項目群)、および item (項目) のタイプを定義する。

Include 要素は前のメタデータバージョンを参照する。これにより、参照された前のメタデータバージョンにあるすべての定義が、このメタデータバージョンに自動的に組み込まれることになる。組み込まれた定義は、新しいメタデータバージョンにおいて新しいバージョンの定義を (同じ OID で) 明示的に記述することによって、いずれも置換 (上書き) することが可能である。Include 要素の機能に関する詳しい情報については、[Include](#) を参照のこと。同様な方法で、新しい定義を (新しい OID で) 追加することも可能である。

注：現在のメタデータモデルには、事象の時間的順序、事象の間隔、事象の条件的発生などのスケジューリング情報は含まれていない。また、form (フォーム) に関しても、類似した情報は含まれていない。

3.1.1.3.1 Include

本体：

EMPTY (空)

属性：

StudyOID	oidref	前のメタデータバージョンを記述している Study 要素を参照する。
MetaDataVersionOID	oidref	前の MetaDataVersion 要素 (上記の Study 要素内) を参照する。

親：

[MetaDataVersion](#)

前のメタデータバージョンへの参照を記述する。このバージョンは、同じ ODM ファイル内、または一連のファイル中の前のファイル内で、先に存在していなければならない。[ODM](#) 要素の PriorFileOID 属性を参照のこと。

前の MetaDataVersion をインクルードした MetaDataVersion で ItemDef などのメタデータ要素を再定義すると、前の MetaDataVersion に同タイプで OID が同じメタデータ要素があれば完全に置き換えられる。これにはすべての属性および子の要素が含まれる。このことは、新しい定義に存在しないすべての属性および子の要素が前のバージョンから事実上削除されることも意味している。

以下の例では、修正した ItemGroupDef に ItemRef 要素を追加することによって、質問 "I.003"

が ItemGroup（項目群）の"IG.001"に追加されている。新しいバージョンの ItemGroupDef に含まれている Aliase は現在 1 つのみで、"MDV.001"の MetaDataVersion で定義された他の Aliase は削除されている。これと同様な方式が、ItemGroupDef から ItemRef を削除したり、その順序を並べ替えたりするために使用される。

```
<Study OID="S.001">
<MetaDataVersion OID="MDV.001" Name="First Metadata version">
<ItemGroupDef OID="IG.001" Name="First ItemGroup">
<ItemRef ItemOID="I.001" Mandatory="Yes" OrderNumber="1"/>
<ItemRef ItemOID="I.002" Mandatory="Yes" OrderNumber="2"/>
<Alias Context="Context1" Name="IG1"/>
<Alias Context="Context2" Name="FIRST"/>
</ItemGroupDef>
</MetaDataVersion>

<MetaDataVersion OID="MDV.002" Name="Second Metadata version">
<Include StudyOID="S.001" MetaDataVersionOID="MDV.001"/>
<ItemGroupDef OID="IG.001" Name="First ItemGroup (modified)">
<ItemRef ItemOID="I.001" Mandatory="Yes" OrderNumber="1"/>
<ItemRef ItemOID="I.003" Mandatory="Yes" OrderNumber="2"/>
<ItemRef ItemOID="I.002" Mandatory="Yes" OrderNumber="3"/>
<Alias Context="Context1" Name="IG1"/>
</ItemGroupDef>
</MetaDataVersion>
</Study>
```

注：前にインクルードされた MetaDataVersion に記述されている定義（例えば、ItemDef）は、後にインクルードされた MetaDataVersion で削除することはできず、そのような Item（項目）への参照のみが削除可能である。参照される Item（項目）は、親の定義（例えば、ItemGroupDef）を書き換え、不要な参照を取り除くことによって削除する。

注：StudyOID 属性により、他の study（試験）のメタデータバージョンを Include 要素で参照できる。従って、様々な study（試験）で共通のメタデータ定義セットを共有することが可能である。実際、メタデータ以外は何も含んでいない study（試験）が、メタデータのライブラリーとして役立つことがある。

注：このようなインクルード情報を MetaDataVersion のサブ要素（属性ではなく）に置くことによって、この機能の将来的な展開を考慮に入れている。例えば、共通のメタデータ辞書を 2 つ含むことを認める、あるいは試験に特異的な前のメタデータバージョンに新しい共通のメタデータを追加するといったことは有用と考えられる。

3.1.1.3.2 Protocol

本体：

[\(Description?, StudyEventRef*, Alias*\)](#)

属性：

NONE（無し）

親：

[MetaDataVersion](#)

この Protocol では、特定のバージョンの Study 内に出現する可能性がある study event（試験事象）のタイプを列記する。これらの study event（試験事象）の 1 つの中にすべての臨床データが出現しなければならない。

注：メタデータに試験実施計画書定義を含んでいない study（試験）は、どのような臨床データも持つことができない。このような study（試験）は、「共通メタデータ辞書」として役立つことがあり、この辞書により study（試験）間でメタデータの共有が可能になる。

3.1.1.3.2.1 Description

本体：

[\(TranslatedText+\)](#)

属性：

NONE（無し）

親：

[Protocol](#), [StudyEventDef](#), [FormDef](#), [ItemGroupDef](#), [ItemDef](#), [ConditionDef](#), [MethodDef](#)

親の試験メタデータ要素に関する説明を自由に記述する。

3.1.1.3.2.2 StudyEventRef

本体：

EMPTY（空）

属性：

StudyEventOID	oidref		StudyEventDef への参照。
OrderNumber	integer	(任意)	
Mandatory	(Yes No)		
CollectionExceptionConditionOID			
D	oidref	(任意)	ConditionDef への参照。

親：

[Protocol](#)

特定バージョンの Study 内に出現する StudyEventDef への参照を記述する。StudyEventRef を列記することで、その study (試験) 内に出現できる study event (試験事象) のタイプを識別する。1つの Protocol 内の StudyEventRef では、StudyEventOID 属性も OrderNumber 属性も重複してはならない。

ユーザーに対して StudyEventDef のリストを提示する際の使用するため、その StudyEventDef の順序を OrderNumber 属性に設定する。事象のスケジュール作成、時間的順序、またはデータの正確さに関して、OrderNumber 属性は無関係である。

Mandatory 属性のフラグは、MetaDataVersion に対応して含まれる臨床データが、このタイプの Study Event (試験事象) がないと不完全であることを示す。このような不完全な ODM 臨床データファイルは、試験のレビューおよび解析目的では不完全とみなされることがある。

CollectionExceptionConditionOID 属性が設定されるならば、この study event (試験事象) のデータを収集すべきではない状況を記述する ConditionDef を参照する。

3.1.1.3.3 StudyEventDef

本体：

([Description?](#), [FormRef*](#), [Alias*](#))

属性：

OID	oid	
Name	name	
Repeating	(Yes No)	
Type	(Scheduled Unscheduled Common)	
Category	text	(任意)

親：

[MetaDataVersion](#)

StudyEventDef は、form (フォーム) のセットをまとめたものである。Scheduled (計画された) Study Event (試験事象) は数セットの form (フォーム) に相当し、その study (試験) で計画された一連の visit (ビジット) の一部として各 subject (被験者) を対象に収集される。Unscheduled (計画されていない) Study Event (試験事象) は、重篤な有害事象による早期終了に対して記入された 1 セットの form (フォーム) など、特定の subject (被験者) で発生するかどうか明らかではないデータを収集するために用意されている。Common (共通の) Study Event (試験事象) は、Adverse Event (有害事象) または Concomitant Medications (併用薬) 歴など、様々な異なるデータ収集事象で使用する form (フォーム) である。

Repeating 属性のフラグは、このタイプの study event (試験事象) が、所定の subject (被験者) で繰り返し発生する可能性があることを示す。

Category 属性は、このタイプの study event (試験事象) に適した試験フェーズを示すためによく使用される。その例としては、Screening (スクリーニング)、PreTreatment (治療前)、

Treatment（治療）、および FollowUp（経過観察）が考えられる。

3.1.1.3.3.1 FormRef

本体：

EMPTY（空）

属性：

FormOID	oidref		FormDef への参照。
OrderNumber	integer	(任意)	
Mandatory	(Yes No)		
CollectionExceptionConditionOID	oidref	(任意)	ConditionDef への参照。

親：

[StudyEventDef](#)

特定の [StudyEventDef](#) 内に出現する [FormDef](#) への参照を記述する。[FormRef](#) を列記することで、このタイプの study event（試験事象）内に出現できる form（フォーム）のタイプを識別する。1 つの [StudyEventDef](#) 内の [FormRef](#) では、[FormOID](#) 属性も [OrderNumber](#) 属性も重複してはならない。

ユーザーに対して form（フォーム）のリストを提示する際に常に使用するため、(study event [試験事象] 内に含まれている) form（フォーム）の順序を [OrderNumber](#) 属性に設定する。事象のスケジュール作成、時間的順序、またはデータの正確さに関して、[OrderNumber](#) 属性は無関係である。

[Mandatory](#) 属性のフラグは、それを含む study event（試験事象）のインスタンスに対する臨床データが、このタイプの form（フォーム）のインスタンスがないと不完全であることを示す。この意味で、不完全な ODM 臨床データファイルは、試験のレビューおよび解析目的では不完全とみなされることがある。

[CollectionExceptionConditionOID](#) 属性を設定すると、この form（フォーム）に関するデータを収集すべきではない状況を記述する [ConditionDef](#) を参照する。

3.1.1.3.4 FormDef

本体：

([Description?](#), [ItemGroupRef*](#), [ArchiveLayout*](#), [Alias*](#))

属性：

OID	oid
Name	name
Repeating	(Yes No)

親：

MetaDataVersion

FormDef では、1 件の study（試験）で出現する可能性がある form（フォーム）のタイプを記述する。

Repeating 属性のフラグは、このタイプの form（フォーム）が、それを含む study event（試験事象）内で繰り返し出現する可能性があることを示す。

3.1.1.3.4.1 ItemGroupRef

本体：

EMPTY（空）

属性：

ItemGroupOID	oidref		ItemGroupDef への参照。
OrderNumber	integer	(任意)	
Mandatory	(Yes No)		
CollectionExceptionConditionOID	oidref	(任意)	ConditionDef への参照。

親：

FormDef

特定の FormDef 内に出現する ItemGroupDef への参照を記述する。ItemGroupRef を列記することで、このタイプの form（フォーム）内に出現できる item group（項目群）のタイプを識別する。1 つの FormDef 内の ItemGroupRef では、ItemGroupOID 属性も OrderNumber 属性も重複してはならない。

ユーザーに対して ItemGroupDef のリストを提示する際に常に使用するため、(FormDef 内に含まれる) ItemGroupDef の順序を OrderNumber 属性に設定する。事象のスケジュール作成、時間的順序、またはデータの正確さに関して、OrderNumber 属性は無関係である。

Mandatory 属性のフラグは、それを含む form（フォーム）のインスタンスに対する臨床データが、このタイプの item group（項目群）のインスタンスがないと不完全であることを示す。この意味で、不完全な ODM 臨床データファイルは、試験のレビューおよび解析目的では不完全とみなされることがある。

CollectionExceptionConditionOID 属性を設定すると、この ItemGroup（項目群）に関するデータを収集すべきではない状況を記述する ConditionDef を参照する。

3.1.1.3.5 ItemGroupDef

本体：

([Description?](#), [ItemRef*](#), [Alias*](#))

属性：

OID	oid
Name	name

Repeating	(Yes No)	
IsReferenceData	(Yes No)	(任意)
SASDatasetName	sasName	(任意)
Domain	text	(任意)
Origin	text	(任意)
Role	name	(任意) 廃止予定
Purpose	text	(任意)
Comment	text	(任意)

親：

[MetaDataVersion](#)

ItemGroupDef では、1 件の study (試験) で出現する可能性がある item group (項目群) のタイプを記述する。

Repeating 属性のフラグは、このタイプの item group (項目群) が、それを含む form (フォーム) (または参照データ) 内で繰り返し出現する可能性があることを示す。

IsReferenceData 属性が Yes の場合、このタイプの item group (項目群) が ReferenceData 要素内のみ出現する可能性がある。IsReferenceData 属性が No の場合、このタイプの item group (項目群) が ClinicalData 要素内のみ出現する可能性がある。この属性のデフォルト値は No である。

Domain 属性、Origin 属性、Purpose 属性、および Comment 属性では、[SDTM Metadata Submission Guidelines \(SDTM メタデータ申請ガイドライン\)](#) に含まれる CDISC Submission Metadata Model (CDISC 申請メタデータモデル) に記載された申請情報を記述する。

注：Role 属性は、Purpose 属性と同義とみなすことができる。新しいアプリケーションは Role 属性ではなく、Purpose 属性を使用すべきである。

3.1.1.3.5.1 ItemRef

本体：

EMPTY (空)

属性：

ItemOID	oidref	ItemDef への参照
OrderNumber	integer	(任意)
Mandatory	(Yes No)	
KeySequence	integer	(任意)
MethodOID	oidref	(任意)
ImputationMethodO ID	Deprecated (廃止予定)	
Role	text	(任意)

RoleCodeListOID	oidref	(任意)	
CollectionException	oidref	(任意)	ConditionDef への参照
ConditionOID			

親 :

[ItemGroupDef](#)

特定の [ItemGroupDef](#) 内に出現する [ItemDef](#) への参照を記述する。[ItemRef](#) を列記することで、このタイプの [item group](#) (項目群) 内に出現できる [item](#) (項目) のタイプを識別する。1 つの [ItemGroupDef](#) 内の [ItemRef](#) では、[ItemOID](#) 属性も [OrderNumber](#) 属性も重複してはならない。

ユーザーに対して [item](#) (項目) のリストを提示する際に常に使用するため、([item Group](#)[項目群] に含まれる) [item](#) (項目) の順序を [OrderNumbers](#) 属性に設定する。事象のスケジュール作成、時間的順序、またはデータの正確さに関して、[OrderNumber](#) 属性は無関係である。

[Mandatory](#) 属性のフラグは、それを含む [item group](#) (項目群) のインスタンスに対する臨床データが、このタイプの [item](#) (項目) のインスタンスがないと不完全であることを示す。この意味で、不完全な [ODM](#) 臨床データファイルは、試験のレビューおよび解析目的では不完全とみなされることがある。

[KeySequence](#) 属性 (設定した場合) は、この [item](#) (項目) が、それを含む [item group](#) (項目群) のキーであることを示す。さらに、そのキーの順序も表す。

[MethodOID](#) 属性では、この [item](#) (項目) の値を得るために使用する [MethodDef](#) を参照する。

[Role](#) 属性には、このデータの [item](#) (項目) の使用を記述する 1 つの規則名を設定する。

[Role](#) 属性を標準の用語体系によって定義する場合、[RoleCodeListOID](#) 属性を使用し、フルセットの規則を定義している [CodeList](#) を参照して、その中から [Role](#) 属性の値を採用してもよい。この属性は、[Role](#) 属性を定義しない限り設定してはならない。また、[Role](#) 属性を定義する場合でも、[RoleCodeListOID](#) 属性の設定は任意である。

[CollectionExceptionConditionOID](#) 属性を設定すると、その [item](#) (項目) の収集が省略される可能性がある状況 (すなわち、[FormalExpression](#) が [True](#) [真] の場合) を定義している [ConditionDef](#) を参照する。例えば、'[InclusionCriteria](#)' という [item group](#) (項目群) に対する '[IsPregnant](#)' という [item](#) (項目) に関するデータを収集するようデザインされたデータ収集ツールでは、男性被験者に関するデータ収集が省略されることがある。参照される [ConditionDef](#) は、以下のように定義されると考えられる。[FormalExpression](#) が [True](#) (真) の場合は、[OID](#) が [IG.ISPREG](#) の [item](#) (項目) を収集する必要はない。

```
<ItemGroupDef OID="INCLUSION" Name="Inclusion Criteria" >
  ...
  <ItemRef ItemOID="IDef.GENDER" Mandatory="Yes"/>
  <ItemRef ItemOID="IDef.ISPREG" Mandatory="Yes"
CollectionExceptionConditionOID="CECID.ISMALE"/>
</ItemGroupDef>
```

```

<ConditionDef OID="CECID.ISMALE">
<TranslatedText xml:lang="en">Do not collect data for Male subjects </TranslatedText>
  <FormalExpression context="PL/SQL">
    INCLUSION.IDef.GENDER = 'Male'
  </FormalExpression>
</ConditionDef>

```

3.1.1.3.6 ItemDef

本体：

([Description?](#), [Question?](#), [ExternalQuestion?](#), [MeasurementUnitRef*](#),
[RangeCheck*](#), [CodeListRef?](#), [Role*](#) [廃止予定], [Alias*](#))

属性：

OID	oid	
Name	name	
DataType	(text integer float date time datetime string boolean double hexBinary base64Binary hexFloat base64Float partialDate partialTime partialDatetime durationDatetime intervalDatetime incompleteDatetime incompleteDate incompleteTime URI)	
Length	positiveInteger	(任意)
SignificantDigits	nonNegativeInteger	(任意)
SASFieldName	sasName	(任意)
SDSVarName	sasName	(任意)
Origin	text	(任意)
Comment	text	(任意)

親：

[MetaDataVersion](#)

ItemDef では、1 件の study (試験) で出現する可能性がある item (項目) のタイプを記述する。item (項目) のプロパティには、名前、データ型、測定単位、範囲またはコードリストの制約の他に、いくつかのプロパティを含む。

DataType 属性では、対応する値の要素が比較および保存でどのように解釈されるべきかを規定する。Length 属性は、DataType 属性が text (テキスト) または string (文字列) の場合に必要であり、DataType 属性が integer (整数) または float (浮動小数点数) では任意で、その他のデータ型では設定すべきではない。SignificantDigit 属性は、DataType 属性が float

(浮動小数点数)では任意で、その他のデータ型では設定すべきではない。DataType 属性が float (浮動小数点数)の場合、Length 属性および SignificantDigit 属性の両方とも設定するか、あるいは両方とも未設定でなくてはならない。

注:バージョン 1.3.0 では、Length 属性および SignificantDigit 属性の使用法が曖昧であった。上記の内容は、これらの 2 つの属性の使用法を明確化している。

DataType=integer、Length=N がその受信システムで要件となっている場合、 10^N 未満の整数値はすべて保存する。これ以上の整数値は受け付けなくてもよい。

DataType=float、Length=N、および SignificantDigits=S がその受信システムで要件となっている場合、 10^N に 10^{-S} を乗じた 10^{N-S} 未満の実数値はすべて保存する。これ以上の実数値は受け付けなくてもよい。中間値については、最も近い 10^{-S} の倍数になるように端数を切り捨ててもよい。

DataType=text、Length=N がその受信システムで処理可能な要件である場合、N 以下の長さのテキスト文字列の値はすべて保存する。テキスト文字列の値にはすべての文字が許される。データ型が Text (テキスト) のデータは、ItemDataString 要素内で転送すべきである。

注: Length 属性および SignificantDigits 属性は、item (項目) のデータ値に関する宣言であり、値の要素においてこれらの値を表記するために用いる文字の数ではない。例えば、"<" という文字は "<" と表記される場合がある。

注:特定の ODM ファイルに対するエンコード文字セットに含まれていないデータ文字は、XML のエンティティまたは文字参照を用いて表記しなければならない。例えば、 Æ は "Æ" と表記できる。

SDSVarName 属性、Origin 属性、および Comment 属性では、[CDISC SDTM](#) の最新バージョンに記載されている申請情報を設定する。

注:本 ODM モデルにおいて、内部キーはすべて変更不可とされている。これは、監査証跡に支障が発生するためである。つまり、本モデルの SubjectKey 属性が、ある患者の実際の外部被験者識別子 (またはランダム化 ID) となっていて、1 つの ODM ファイルでその値が誤って送られた場合、継続するファイルでその誤りを正す方法はないと考えられる。こうした場合、外部被験者キー (および他の外部から見えるキー変数) は、そのメタデータ内の item (項目) として定義すべきであると考えている。こうすることで、その誤りは通常の修正または監査のメカニズムを通して修正される可能性がある。これにより、試験キーの修正に対応する問題は解消されるが、ユーザーがどの ItemDef に特別な意味があるのか、またはその意味は何かを確認する方法は依然としてないことになる。この問題が最も明らかになるのは、外部ソースからデータをロードする際に患者をマッチングさせる場合である。患者の ID が不明であれば、どのようにマッチングを行うのであろうか。

その答は ItemDef の SDSVarName 属性を利用することである。SDSVarName 属性は任意の属性で、その item (項目) にビジネスを意味するタグを付けるために使用することもできる。本 ODM モデルで可能性のある意味をすべて列挙しようとするよりも、CDISC の SDTM に定義されている変数名セットは臨床データを管理する上で使用される中核変数を網羅していることから、この変数セットを利用することが最善であると ODM ワーキンググループは考えた。こうすれば、ODM に準拠した XML インスタンスを処理するソフトウェアは、SDSVarName 属性で規定した値を用いて、頻繁に使用される変数である標準変数を識別可能になる。この属性の利用は SDTM

モデルで定義された変数に限定される。変数にタグを付けておき、その変数に対して SDTM 定義にマッチするものを識別する。一般に使用されている値のうち、一部のリストを以下に示す。

- STUDYID (申請の中で一意の試験識別子)
- USUBJID (申請の中で一意の試験識別子)、
- SUBJID (試験の中で一意の被験者識別子)、
- SITEID (試験実施施設に対する一意の識別子)
- SEX (性別のコード値)、
- VISITNUM (臨床における対面を示す番号)
- VISIT (試験実施計画書で定義された臨床における対面の記述)、
- VISITDY (計画された試験日である VISIT)

SDTM 変数に関する詳細情報については SDTM 仕様書および実装ガイドを参照のこと。

Question 要素では、この **item** (項目) に関するデータをユーザーに求めた場合に表示するテキストの内容を記述する。**ExternalQuestion** 要素も同様な機能があるが、外部で定義された質問を参照する。両方とも記述する場合は、一貫性を持たせるべきである。

MeasurementUnitRef では、このタイプの **item** (項目) に対して許容される測定単位を列挙する。測定単位を記述できるのは数値の **item** (項目) のみである。**MeasurementUnitRef** が 1 つのみの場合、このタイプの **item** (項目) はすべてデフォルトでこの測定単位を持つことになる。すなわち、**ItemDef** で **MeasurementUnitRef** を定義し、それに対応する **ItemData** で **MeasurementUnitRef** を記述していない場合、その **ItemData** に記述した値は、**ItemDef** 内の **MeasurementUnitRef** で設定した単位となる。

数値の **item** (項目) 定義で **MeasurementUnitRef** を記述しない場合、その **item** (項目) の値はスカラー (すなわち、単位を持たない純数) である。

RangeCheck では、このタイプの **item** (項目) に対して許容する値の範囲を制限する。

CodeListRef (記述した場合) は、このタイプの **item** (項目) に対して許容する値を、参照するコードリストに含まれる値に限定する。

注：1 つの **item group** (項目群) 内に同じ **item** (項目) が繰り返し出現することはない。

3.1.1.3.6.1 Question

本体：

[\(TranslatedText+\)](#)

属性：

NONE (無)

親：

[ItemDef](#)

ある **item** (項目) に関するデータを印刷または画面上に表示するようユーザーに求めた場

合に提示するラベルを記述する。

3.1.1.3.6.2 ExternalQuestion

本体：

EMPTY (空)

属性：

Dictionary	text	(任意)	外部の質問辞書名。
Version	text	(任意)	外部の質問辞書のバージョン指定子。
Code	text	(任意)	外部辞書内の特定の質問を選択するコード。

親：

[ItemDef](#)

外部で定義された質問への参照を記述する。

3.1.1.3.6.3 MeasurementUnitRef

本体：

EMPTY (空)

属性：

MeasurementUnit nitOID	oidref		Reference to the MeasurementUnit 定義への参照。
-----------------------------------	------------------------	--	--

親：

[ItemData](#), [ItemDef](#), [RangeCheck](#)

測定単位定義への参照を記述する。

3.1.1.3.6.4 RangeCheck

本体：

(([CheckValue+](#) | [FormalExpression+](#)), [MeasurementUnitRef?](#), [ErrorMessage?](#))

属性：

Comparator	(LT LE GT GE EQ NE IN NOTIN)	(任意)	項目と値を比較するために使用する比較演算子
SoftHard	(Soft Hard)	(任意)	

親：

[ItemDef](#)

RangeCheck は、この要素で囲んだ item (項目) の値に対する範囲の制限を定義する。ItemData の値が有効な場合は True (真)、無効な場合は False (偽) と判定する比較式を記述する。この比較式は、Comparator 属性と CheckValue を用いるか、FormalExpression を用いて規定する。

Comparator 属性と CheckValue を用いた範囲チェック

Comparator (比較演算子) と Check Value (チェック値) を用いる場合、1 つの RangeCheck で片側の制限を記述する。より複雑な制限を規定するために複数の RangeCheck を記述することも可能である。例えば、上限且つ下限の指定では 2 つの RangeCheck が必要である。

各制限式は以下と同じである。

項目値 比較演算子 チェック値

実際のデータ値が制限範囲を外れた場合は、そのデータを受け付けない (Hard [ハード] 制約)、あるいは警告を発する (Soft [ソフト] 制約) のいずれかとなる。

以下の比較演算子では、CheckValue が 1 つ必要である。

LT	未満 (<)
LE	以下 (\leq)
GT	超過 (>)
GE	以上 (\geq)
EQ	等しい (=)
NE	異なる (\neq)

以下の比較では、CheckValue が 1 セット必要である。

IN	リスト値の 1 つに該当する
NOTIN	リスト値のいずれにも該当しない

測定単位を規定する場合、それに対応する item (項目) の値は、相互換算可能な測定単位を (明示またはデフォルトのいずれかで) 持たなければならない。範囲チェックの一環として、適切な単位の換算を行わなければならない。測定単位を規定しない場合、それに対応する item (項目) の値は、測定単位が (明示またはデフォルトのいずれでも) あってはならない。

以下に例を示す。

値は正でなければならない	<code><RangeCheck Comparator="GT"> <CheckValue>0</CheckValue> </RangeCheck></code>
値は 18~65 でなければならない	<code><RangeCheck Comparator="GE"> <CheckValue>18</CheckValue> </RangeCheck> <RangeCheck Comparator="LE"></code>

```
<CheckValue>65</CheckValue>
</RangeCheck>
値は (1、5、7) のいずれかでなければならぬ
<RangeCheck Comparator="IN">
  <CheckValue>1</CheckValue>
  <CheckValue>3</CheckValue>
  <CheckValue>5</CheckValue>
</RangeCheck>
```

FormalExpression を用いた範囲チェック

FormalExpression を用いる場合、1 つの RangeCheck ですべてのチェック（例えば、片側チェックまたは複数方向でのチェック）が記述可能である。CheckValue、Comparator 属性、または MeasurementUnitRef は、FormalExpression 自体ですべて表記できるため、このタイプのチェックでは、これらを記述してはならない。FormalExpression は ItemData 要素の値を判定し、その式の結果を真偽値で返す。それぞれに異なった Context 属性を設定すれば、複数の FormalExpression が記述可能であり、そうすることで、同じ比較式を多くのシステムに適した形式で記述することができる。複数の異なった比較式はそれぞれ意味が異なるため、別の RangeCheck として記述しなければならない。

3.1.1.3.6.4.1 CheckValue

本体：

[value](#)

属性：

NONE（無し）

親：

[RangeCheck](#)

範囲チェックに使用する比較値を記述する。この比較値は、この要素を含む ItemDef で宣言したデータ型の値に一致しなければならない。

3.1.1.3.6.4.2 ErrorMessage

本体：

[\(TranslatedText+\)](#)

属性：

NONE（無し）

親：

[RangeCheck](#)

範囲チェックで逸脱した場合に使用するエラーメッセージを記述する。

3.1.1.3.6.5 CodeListRef

本体 :

EMPTY (空)

属性 :

CodeListOID [oidref](#) [CodeList](#) 定義への参照。

親 :

[ItemDef](#)

CodeList 定義への参照を記述する。参照する CodeList および親である ItemDef の DataType 属性は同じでなければならない。

3.1.1.3.6.6 Alias

本体 :

EMPTY (空)

属性 :

Context [text](#)

Name [text](#)

親 :

[Protocol](#), [StudyEventDef](#), [FormDef](#), [ItemGroupDef](#), [ItemDef](#), [CodeList](#),
[CodeListItem](#), [EnumeratedItem](#), [MethodDef](#), [ConditionDef](#)

Alias では、ある要素に対する別名 (エイリアス名) を記述する。Context 属性には、この別名が関係するアプリケーションドメインを指定する。

3.1.1.3.7 CodeList

本体 :

([Description?](#), ([CodeListItem](#)+ | [EnumeratedItem](#)+ | [ExternalCodeList](#)), [Alias](#)*)

属性 :

OID [oid](#)

Name [name](#)

DataType (integer | float | text |
string)

SASFormatName [sasFormat](#) (任意)

親 :

MetaDataVersion

ある item (項目) に対して許容される個々の値のセットを定義する。この定義により、明示的な値リスト (CodeListItem+ | EnumeratedItem+)、または外部で定義されたコードリストへの参照 (ExternalCodeList) が可能になる。

DataType 属性では、内部または外部に関わらず、CodeList で記述可能な値を制限する。

SASFormatName 属性は、正当な SAS フォーマットでなければならない。

3.1.1.3.7.1 CodeListItem

本体 :

(Decode, Alias*)

属性 :

CodedValue	<u>text</u>		コードリスト項目の値 (臨床データで出現すると考えられる値)。
Rank	<u>float</u>	(任意)	CodeList 内の他の CodeListItem と比較して、その CodeListItem の重要性を数値で表す。 Rank 属性は任意であるが、1 つの CodeList 内でいくつかの CodeListItem に対して設定する場合は、すべての CodeListItem に対して設定しなければならない。

親 :

CodeList

表示フォーマットを含むコードリストの個々のメンバー値を定義する。1 セットの印刷/表示用の form (フォーム) に加えて、その実際の値を記述する。CodedValue 属性は、CodeList に含まれる DataType 属性の中で許容される値でなければならない。

1 つの CodeList 内の CodeListItem では、(CodeList の DataType 属性により解釈される) CodedValue 属性の値が重複してはならない。

列挙値に対応する相対値がその辞書順で決定できない、またはそうすべきでない場合に、Rank 属性を使用してもよい。例えば、"Low"、"Medium"、および"High"を含むテキスト値を列挙したリストがあり、これらのテキスト値にそれぞれ 1、2、および 3 という相対的な数値を割り付けたい場合、定義する各 CodeListItem に Rank 属性を設定すべきである。Rank 属性を指定しないと、通常の辞書順は"High"、"Low"、"Medium"の順となる。

3.1.1.3.7.1.1 Decode

本体 :

(TranslatedText+))

属性 :

NONE (無し)

親 :

[CodeListItem](#)

親の CodedValue 属性に関して表示する値を記述する。

3.1.1.3.7.2 ExternalCodeList

本体 :

EMPTY (空)

属性 :

Dictionary	text	(任意)	外部のコードリスト名。
Version	text	(任意)	外部コードリストのバージョン指定子。
ref	text	(任意)	その辞書のローカルインスタンスへの参照。
href	text	(任意)	その辞書の外部インスタンスの URL。

親 :

[CodeList](#)

外部で定義されたコードリストへの参照を記述する。

3.1.1.3.7.3 EnumeratedItem

本体 :

([Alias](#)*)

属性 :

CodedValue	text		列挙される item (項目) の値 (臨床データで出現すると考えられる値)。
Rank	float	(任意)	CodeList 内の他の CodeListItem と比較して、その CodeListItem の重要性を数値で表す。 Rank 属性は任意であるが、1 つの CodeList 内でいくつかの EnumeratedItem に対して設定する場合は、すべての EnumeratedItem に対して設定しなければならない。

親 :

[CodeList](#)

有効値リストの個々のメンバー値を定義する。コード化された値のみを記述する。印刷/表示用の form (フォーム) が必要な場合は、CodeListItem を使用すべきである。1 つのコードリスト内に CodeListItem と EnumeratedItem が混在しないようにする。CodedValue 属性

は、CodeList に含まれる DataType 属性で許容される値でなくてはならない。

1 つの CodeList 要素内の EnumeratedItem では、(CodeList の DataType 属性によって解釈される) CodedValue 属性の値が重複してはならない。

列挙値に対応する相対値がその辞書順で決定できない、またはそうすべきでない場合に、Rank 属性を使用してもよい。例えば、"Low"、"Medium"、および"High"を含むテキスト値を列挙したリストがあり、これらのテキスト値にそれぞれ 1、2、および 3 という相対的な数値を割り付けたい場合、定義する各 EnumeratedItem に Rank 属性を設定すべきである。Rank 属性を適用しないと、通常の辞書順は"High"、"Low"、"Medium"の順となる。

3.1.1.3.8 ArchiveLayout

本体：

EMPTY (空)

属性：

OID	oid	
PdfFileName	fileName	表示画面レイアウトを含む Adobe 社の PDF のファイル名。
PresentationOID	oidref	(任意) Presentation 定義への参照。

親：

[FormDef](#)

1 つの form (フォーム) でデータを収集するために使用する表示画面レイアウトの表示イメージを記述する。試験の保存に有用である。

3.1.1.3.9 MethodDef

本体：

([Description](#), [FormalExpression*](#), [Alias*](#))

属性：

OID	oid
Name	name
Type	(Computation Imputation Transpose Other)

親：

[MetaDataVersion](#)

MethodDef は、他のデータ値より、あるデータ値を取得する方法を定義する。Description 要素は必ず記述し、散文体で説明を加えるべきである。この Description 要素は MethodDef の

規定内容である。

FormalExpression を記述する場合は、Description 要素を実装し、ある値を返す機械による判読が可能な式を含まなければならない。それぞれに異なった Context 属性を設定すれば、複数の FormalExpression が記述可能であり、そうすることで、同じ式を多くのシステムに適した書式で記述することができる。

3.1.1.3.10 Presentation

本体：

[text](#)

属性：

OID	oid	
xml:lang	languageTag	(任意)

親：

[MetaDataVersion](#)

Presentation では、study (試験) に関する情報をシステムユーザーに提示する方法を定義する。xml:lang 属性は、適切な提示言語の選択に有用である。[TranslatedText](#) を参照のこと。

注：CDISC ODM V1.3 において Presentation 要素は詳細に定義されていない。Presentation 要素は今後のさらなる発展のためのプレースホルダ (代替要素) である。

3.1.1.3.11 ConditionDef

本体：

([Description](#), [FormalExpression](#)*, [Alias](#)*)

属性：

OID	oid
Name	name

親：

[MetaDataVersion](#)

ConditionDef では、真偽条件を定義する。Description 要素は必ず記述し、散文体で説明を加えるべきである。この Description 要素は ConditionDef の規定内容である。ConditionDef は、study (試験) のメタデータ構成要素内の CollectionExceptionConditionOID 属性によって参照されるが、その条件 (すなわち、FormalExpression が True [真] の場合) によって定義される状況下では省略されることがある。

FormalExpression を記述する場合は、機械による判読が可能な式で、True (真) または False (偽) と判定される式を含めなければならない。それぞれ異なった Context 属性を設定すれば、複数の FormalExpression が記述可能であり、そうすることで、同じ式を多くのシス

テムに適した書式で記述することができる。

アプリケーションが **FormalExpression** のいずれかを解釈できないか、条件付きのデータ収集を標準でサポートしていない場合は、**Condition** が規定されていないかのように、参照している study (試験) のメタデータ構成要素に関するデータを収集すべきである。

3.1.1.3.11.1 FormalExpression

本体：

(PCDATA)

属性：

Context [text](#) 適切なコンピューター言語を示す自由形式の修飾子で、
FormalExpression の内容を判断する際に使用する。

親：

[ConditionDef](#), [MethodDef](#), [RangeCheck](#)

ConditionDef または **RangeCheck** 内で使用する **FormalExpression** は、**True** (真) または **False** (偽) でなければならない。Type 属性が **Imputation** (補完)、**Computation** (計算)、または **Transpose** (置換) の **MethodDef** 内で参照する **FormalExpression** は、その **Method** (方法) を用いて項目を補完または計算する可能性があるため、その **DataType** (データ型) は正確でなければならない。

3.1.2 管理上の要素

3.1.2 AdminData

本体：

([User*](#), [Location*](#), [SignatureDef*](#))

属性：

StudyOID [oidref](#) (任意) [StudyDef](#) への参照。

親：

[ODM](#)

ユーザー、試験実施施設、電子署名に関する管理情報を記述する。

StudyOID 属性を指定しない場合は、現状の **ODM** ファイル内に定義されたすべての study (試験)、および/または **PriorFileOID** 属性の仕様を通してリンクされる一連の **ODM** ファイル内に定義されたすべての study (試験) に対して、**AdminData** の定義を適用するものとみなす。

3.1.2.1 User

本体：

([LoginName?](#), [DisplayName?](#), [FullName?](#), [FirstName?](#), [LastName?](#), [Organization?](#),

[Address*](#), [Email*](#), [Picture?](#), [Pager?](#), [Fax*](#), [Phone*](#), [LocationRef*](#), [Certificate*](#))

属性：

OID [oid](#)
UserType (Sponsor | (任意)
 Investigator |
 Lab | Other)

親：

[AdminData](#)

臨床データ収集システムの特定期間に関する情報を記述する。このようなユーザーとしては、Investigator（試験責任者）、モニター（CRA）、またはデータマネージャーが考えられる。この意味で study（試験）の subject（被験者）はユーザー **ではない**。

3.1.2.1.1 LoginName

本体：

[text](#)

属性：

NONE（無し）

親：

[User](#)

ユーザーのログイン識別名を記述する。

3.1.2.1.2 DisplayName

本体：

[text](#)

属性：

NONE（無し）

親：

[User](#)

表示可能な短いユーザー名を記述する。

3.1.2.1.3 FullName

本体：

[text](#)

属性：

NONE（無し）

親：

[User](#)

ユーザーの正式なフルネームを記述する。

3.1.2.1.4 FirstName

本体：

[text](#)

属性：

NONE（無し）

親：

[User](#)

ユーザーの名（ファーストネームまたはミドルネームを含む名）を記述する。

3.1.2.1.5 LastName

本体：

[text](#)

属性：

NONE（無し）

親：

[User](#)

ユーザーの氏（ファミリーネーム）を記述する。

3.1.2.1.6 Organization

本体：

[text](#)

属性：

NONE（無し）

親：

[User](#)

ユーザーが所属する組織を記述する。

3.1.2.1.7 Address

本体：

([StreetName](#)*, [City](#)?, [StateProv](#)?, [Country](#)?, [PostalCode](#)?, [OtherText](#)?)

属性 :

NONE (無し)

親 :

[User](#)

ユーザーの住所を記述する。

3.1.2.1.8 StreetName

本体 :

[text](#)

属性 :

NONE (無し)

親 :

[Address](#)

ユーザーの住所のうち、所在地の部分を記述する。

3.1.2.1.9 City

本体 :

[text](#)

属性 :

NONE (無し)

親 :

[Address](#)

ユーザーの住所のうち、市名の部分を記述する。

3.1.2.1.10 StateProv

本体 :

[text](#)

属性 :

NONE (無し)

親 :

[Address](#)

ユーザーの住所のうち、州名または行政区域名の部分を記述する。

3.1.2.1.11 Country

本体：

[text](#)

属性：

NONE（無し）

親：

[Address](#)

ユーザーの住所のうち、国名の部分を記述する。これは ISO 3166 による 2 文字の国コードで表記しなければならない。

例：フランスは FR、日本は JP である。

3.1.2.1.12 PostalCode

本体：

[text](#)

属性：

NONE（無し）

親：

[Address](#)

ユーザーの住所のうち、郵便番号の部分を記述する。

3.1.2.1.13 OtherText

本体：

[text](#)

属性：

NONE（無し）

親：

[Address](#)

ユーザーの住所の一部として他に必要なテキストがあれば記述する。

3.1.2.1.14 Email

本体：

[text](#)

属性：

NONE（無し）

親：

[User](#)

ユーザーの電子メールアドレスを記述する。

3.1.2.1.15 Picture

本体：

EMPTY（空）

属性：

PictureFileName [fileName](#)

ImageType [name](#) (任意)

親：

[User](#)

ユーザーの肖像（写真またはイメージ画）を指定する。

3.1.2.1.16 Pager

本体：

[text](#)

属性：

NONE（無し）

親：

[User](#)

ユーザーのポケットベル用電話番号を記述する。

3.1.2.1.17 Fax

本体：

[text](#)

属性：

NONE（無し）

親：

[User](#)

ユーザーのファクシミリ装置用電話番号を記述する。

3.1.2.1.18 Phone

本体 :

[text](#)

属性 :

NONE (無し)

親 :

[User](#)

ユーザーの音声電話番号を記述する。

3.1.2.1.19 LocationRef

本体 :

EMPTY (空)

属性 :

LocationOID [oidref](#) [Location](#) 定義への参照。

親 :

[AuditRecord](#), [Signature](#), [User](#)

ユーザーの所在地への参照を記述する。

3.1.2.1.20 Certificate

本体 :

[text](#)

属性 :

NONE (無し)

親 :

[User](#)

ユーザーの電子署名証明書を記述する。

注 : CDISC の ODM V1.3 において Certificate 要素は詳細に定義されていない。この要素は今後のさらなる発展のために用意したプレースホルダ (代替要素) である。

3.1.2.2 Location

本体 :

[\(MetaDataVersionRef+\)](#)

属性 :

OID	oid	
Name	name	
LocationType	(Sponsor Site CRO Lab Other)	(任意)

親：

[AdminData](#)

所在地（一般には臨床試験の Site [実施施設] または Sponsor [試験依頼者] の事務所）を記述する。

3.1.2.2.1 MetaDataVersionRef

本体：

EMPTY (空)

属性：

StudyOID	oidref	このメタデータバージョンを使用する Study を参照。
MetaDataVersionOID	oidref	MetaDataVersion (上記 Study 内) を参照。
EffectiveDate	date	

親：

[Location](#)

親の Location で使用する MetaDataVersion への参照を記述する。EffectiveDate 属性は、ある所在地で使用されるメタデータが時間とともに変化する可能性があるという事実を表している。

3.1.2.3 SignatureDef

本体：

([Meaning](#), [LegalReason](#))

属性：

OID	oid	
Methodology	(Digital Electronic)	(任意)

親：

[AdminData](#)

一種の電子署名で、[21 CFR Part 11](#) で要求される *meaning* (意味) を含めて定義する。署名

の属性が Digital (デジタル型) であれば、暗号技術に基づいた署名である。それ以外の署名の属性は Electronic (電子型) である。

注：データ編集の追跡は署名ではなく [AuditRecords](#) を用いて行う。

3.1.2.3.1 Meaning

本体：

[text](#)

属性：

NONE (無し)

親：

[SignatureDef](#)

この種の署名の短い名前 (例えば、データ作成者、承認者など) を記述する。

3.1.2.3.2 LegalReason

本体：

[text](#)

属性：

NONE (無し)

親：

[SignatureDef](#)

署名に伴う責任の説明 (例えば、「署名者はこのデータの正確性に関する責任を負う」など) を記述する。

3.1.3 参照データの要素

3.1.3 ReferenceData

本体：

([ItemGroupData*](#), [AuditRecords*](#), [Signatures*](#), [Annotations*](#))

属性：

StudyOID [oidref](#)

この参照データに関するメタデータを定義する [Study](#) を参照する。

MetaDataVersionOID [oidref](#)

この参照データに対する [MetaDataVersion](#) (上記の Study 内) を参照する。

親：

ODM

参照データは臨床データの解釈方法に関する情報を提供する。例えば、参照データが臨床検査の基準値を含む場合があるなど。ReferenceData 内に組み込まれた Signature 要素は意味がなく、無視すべきである。

StudyOID 属性および MetaDataVersionOID 属性により、ある特定のメタデータバージョンを選択する。この ReferenceData 要素内のすべてのメタデータの参照 (OID) は、選択されたメタデータバージョン内の定義を参照する。

TransactionType 属性は、ReferenceData 内でも ClinicalData 内と同じように機能する。

注：参照データは特定の study (試験) から独立している可能性があるため、その参照メタデータを臨床メタデータとは分離しておくことが望ましい場合がある。これは、Protocol、StudyEventDef、または FormDef を含まない Study 要素を作成することで実現可能である。その場合、ItemGroupDef はすべて IsReferenceData=Yes となる。このような study (試験) には臨床データがない。

3.1.4 臨床データの要素

3.1.4 ClinicalData

本体：

([SubjectData*](#), [AuditRecords*](#), [Signatures*](#), [Annotations*](#))

属性：

StudyOID	oidref	この要素内でネスト化されたデータを使用する Study を参照する。
MetaDataVersionOID	oidref	この要素内でネスト化されたデータを管理する MetaDataVersion (上記の Study 要素内) を参照する。

親：

ODM

複数の subject (被験者) に関する臨床データを記述する。

StudyOID 属性および MetaDataVersionOID 属性により、ある特定のメタデータバージョンを選択する。この ClinicalData 要素内のすべてのメタデータの参照 (OID) は、選択されたメタデータバージョン内の定義を参照する。

3.1.4.1 SubjectData

本体：

([AuditRecord?](#), [Signature?](#), [InvestigatorRef?](#), [SiteRef?](#), [Annotation*](#), [StudyEventData*](#))

属性：

SubjectKey	<u>subjectKey</u>		
TransactionType	(Insert (任意) Snapshot (スナップショット) 型文書では Update TransactionType 属性を記述する必要はない。 Remove い。 Upsert Context)		

親：

ClinicalData

1 人の subject (被験者) に関する臨床データを記述する。

SubjectKey 属性は特定の subject (被験者) を識別するために使用する。このキーは、親の ClinicalData 要素により規定した study (試験) 内の 1 人の subject (被験者) を一意に識別する。これと同じ SubjectKey を、1 件の study (試験) における複数の ClinicalData 要素内で使用した場合は、同じ subject (被験者) であると解釈される。

例：

```
<ClinicalData StudyOID="P2006-101" MetadataVersionOID="101.01">
<SubjectData SubjectKey="1000" TransactionType="Insert">
<StudyEventData StudyEventOID="Screen">
<FormData FormOID="DEMOG">
<ItemGroupData ItemGroupOID="DM">
<ItemDataString ItemOID="USUBJID">101-001-001</ItemDataString>
<ItemDataString ItemOID="SEX">F</ItemDataString>
</ItemGroupData>
</FormData>
<FormData FormOID="LABDATA">
<ItemGroupData ItemGroupOID="LB">
<ItemDataDatetime ItemOID="LBDTC">2006-07-14T14:48</ItemDataDatetime>
<ItemDataString ItemOID="LBTESTCD">ALT</ItemDataString>
<ItemDataString ItemOID="LBORRES">245</ItemDataString>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
<ClinicalData StudyOID="P2006-101" MetadataVersionOID="101.02">
<SubjectData SubjectKey="1000" TransactionType="Insert">
<StudyEventData>
<FormData FormOID="AENONSER">
<ItemGroupData ItemGroupOID="AE">
```



```

<ItemDataString ItemOID="AETERM">Fever</ItemDataString>
<ItemDataDate ItemOID="AESTDTC">2006-08-21</ItemDataDate>
</ItemGroupData>
</FormData>
<FormData FormOID="LABDATA">
<ItemGroupData ItemGroupOID="LB">
<ItemDataDatetime ItemOID="LBDTC">2006-07-14T14:48</ItemDataDatetime>
<ItemDataString ItemOID="LBTESTCD">ALT</ItemDataString>
<ItemDataString ItemOID="LBORRES">300</ItemDataString>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>

```

3.1.4.1.1 StudyEventData

本体 :

([AuditRecord?](#), [Signature?](#), [Annotation*](#), [FormData*](#))

属性 :

StudyEventOID	oidref		StudyEventDef への参照。
StudyEventRepeatKey	repeatKey	(任意)	1 人の subject (被験者) に関して同タイプの study event (試験事象) が繰り返し出現する場合、それを区別するために使用するキー。
TransactionType	(Insert Update Remove Upsert Context)	(任意)	Snapshot (スナップショット) 型文書では TransactionType 属性を記述する必要はない。

親 :

[SubjectData](#)

1 つの study event (試験事象) つまり visit (ビジット) に対する臨床データを記述する。本モデルでは、study event (試験事象) の繰り返し (例えば、一連の患者 visit (ビジット) に対して同一セットの情報を収集する場合) をサポートしている。

StudyEventOID 属性および StudyEventRepeatKey 属性は、特定の study event (試験事象) を識別するために同時に使用する。このペアの属性値により、その subject (被験者) に含まれる StudyEvent (試験事象) を一意に識別する。StudyEventRepeatKey 属性を設定するのは、StudyEventDef が繰り返し出現する場合に限る。

3.1.4.1.1.1 FormData

本体 :

([AuditRecord?](#), [Signature?](#), [ArchiveLayoutRef?](#), [Annotation*](#), [ItemGroupData*](#))

属性 :

FormOID	oidref		FormDef への参照。
FormRepeatKey	repeatKey	(任意)	1 件の study event (試験事象) 内で同タイプの form (フォーム) が繰り返し出現する場合、それを区別するために使用するキー。
TransactionType	(Insert Update Remove Upsert Context)	(任意)	Snapshot (スナップショット) 型文書では TransactionType 属性を記述する必要はない。

親 :

[StudyEventData](#)

1 つの form (フォーム) つまり page (ページ) に対する臨床データを記述する。本モデルでは、1 つの study event (試験事象) における form (フォーム) の繰り返し (例えば、1 回の患者 visit (ビジット) で複数の有害事象が記録された場合) をサポートしている。

FormOID 属性および FormRepeatKey 属性は、特定の form (フォーム) を識別するために一緒に使用される。このペアの属性値により、その study event (試験事象) に含まれる form (フォーム) を一意に識別する。FormRepeatKey 属性を設定するのは、FormDef が繰り返す場合に限る。

3.1.4.1.1.1.1 ItemGroupData

本体 :

([AuditRecord?](#), [Signature?](#), [Annotation*](#), ([ItemData*](#) | [ItemData\[TYPE\]*](#)))

属性 :

ItemGroupOID	Oidref		ItemGroupDef への参照。
ItemGroupRepeatKey	repeatKey	(任意)	1 つの form (フォーム) 内で同タイプの item group (項目群) が繰り返し出現する場合、それを区別するために使用するキー。
TransactionType	(Insert Update Remove Upsert Context)	(任意)	Snapshot (スナップショット) 型文書では TransactionType 属性を記述する必要はない。

親 :

FormData, ReferenceData

1 つの item group (項目群) つまり record (レコード) に対する臨床データを記述する。本モデルでは、1 つの form (フォーム) 内での item group (項目群) の繰り返し (例えば、1 既往歴で過去の数回の入院が報告される場合)、または参照データ内での繰り返しをサポートしている。

ItemGroupOID 属性および ItemGroupRepeatKey 属性は、特定の item group (項目群) を識別するために同時に使用する。このペアの属性値により、その form (フォーム) に含まれる item group (項目群) を一意に識別する。ItemGroupRepeatKey 属性を設定するのは、ItemGroupDef が繰り返し出現する場合に限る。

ItemGroupData は参照データとしても出現する可能性がある。その場合、ItemGroupOID 属性と ItemGroupRepeatKey 属性のペアは、その参照データ内で一意でなければならない。

3.1.4.1.1.1.1.1 ItemData

本体 :

([AuditRecord?](#), [Signature?](#), [MeasurementUnitRef?](#), [Annotation*](#))

属性 :

ItemOID	oidref		ItemDef への参照。
TransactionType	(Insert Update Remove Upsert Context)	(任意)	Snapshot (スナップショット) 型文書では TransactionType 属性を記述する必要はない。
Value	text	(任意)	1 つの item (項目) について収集されたデータである。このデータは ItemDef の DataType 属性に従って記述する。
IsNull	(Yes)	(任意)	IsNull 属性は、item (項目) の値が NULL (ヌル) 値に設定されるべきであることを示すフラグである。Value 属性を設定する場合は、IsNull 属性を設定してはならない。IsNull 属性を設定する場合は、Value 属性を設定してはならない。冗長でない XML インスタンスを作成する為には、未収集データを示すために IsNull 属性を "Yes" に設定した ItemData 要素を使用すべきではない。その場合は、収集したデータのみを送信する方がよい。

親 :

ItemGroupData

この ItemData 要素は、1 つの item (項目) に対する臨床データの un-typed (型なし) の転送に使用する場合がある。本モデルでは、1 つの item group (項目群) 内での同じ item (項目) の繰り返しをサポートしていない。

ItemOID 属性を使用して、特定の item (項目) 定義を識別する。この属性値により、その item group (項目群) に含まれる item (項目) を一意に識別する。

参照される [ItemDef](#) は、この item (項目) の DataType (データ型) を定義している。Value 属性の文字列は、[データフォーマット](#)のセクションに記載したデータフォーマットに関する規則に従わなければならない。text (テキスト)、string (文字列)、integer (整数)、float (浮動小数点数) のデータ型では、ItemDef で定義した Length 属性による制限長以内でその item (項目) の値を保存することが可能でなければならない。

1 つの ODM ファイル内で typed (型付き) データと untyped (型なし) データが混在してはならない。

3.1.4.1.1.1.2 ItemData[TYPE]

本体 :

(PCDATA)

属性 :

ItemOID	oidref		ItemDef への参照。
TransactionType	(Insert Update Remove Upsert Context)	(任意)	Snapshot (スナップショット) 型文書では TransactionType 属性を記述する必要はない。
AuditRecordID	IDREF	(任意)	AuditRecord への参照。
SignatureID	IDREF	(任意)	Signature への参照。
AnnotationID	IDREF	(任意)	Annotation への参照。
MeasurementUnitOID	oidref	(任意)	MeasurementUnit 定義への参照。
IsNull	(Yes)	(任意)	ItemDataAny に対してのみ許される。 IsNull 属性は、item (項目) の値が NULL (ヌル) 値に設定されることを示すフラグである。要素の内容 (PCDATA) を記述する場合は、IsNull 属性を設定してはならない。 IsNull 属性を設定する場合は、要素の内容 (PCDATA) を記述してはならない。冗長でない XML インスタンスを作成する為には、

未収集データを示すために IsNull 属性を "Yes"に設定した ItemData 要素を使用すべきではない。その場合は、収集したデータのみを送信する方がよい。

親：

ItemGroupData

ItemData[TYPE]要素では、typed (型付き) データの内容を要素の PCDATA として記述する。PCDATA のデータ型は、対応する ItemDef の DataType 属性に一致しなければならない。例えば、ItemDataInteger 要素は、対応する ItemDef の Datatype 属性値が integer (整数) である item (項目) に対してのみ使用できる。この場合はさらに、PCDATA の内容が構文解析可能な整数値でなければならない。内容の構文解析における例外については、[ItemDataAny](#) も参照のこと。

1 つの ODM ファイル内で、typed (型付き) データと untyped (型なし) データが混在してはならない。

以下の ItemData[TYPE]要素が使用可能である。

ItemData**Any**

ItemData**String**

ItemData**Integer**

ItemData**Float**

ItemData**Date**

ItemData**Time**

ItemData**Datetime**

ItemData**Boolean**

ItemData**HexBinary**

ItemData**Base64Binary**

ItemData**HexFloat**

ItemData**Base64Float**

ItemData**PartialDate**

ItemData**PartialTime**

ItemData**PartialDatetime**

ItemData**DurationDatetime**

ItemData**IntervalDatetime**

ItemData**IncompleteDatetime**

ItemData**IncompleteDate**

ItemDataIncompleteTime

ItemDataURI

DataType (データ型) が *TYPE* に一致しない値を転送するには、ItemDataAny 要素を使用しなければならない。ItemDataAny のデータを受信するアプリケーションは、これらのデータ値を、対応するデータフィールドにロードする必要がない。

DataType (データ型) が *TYPE* に一致しない値を転送するために、ItemDataAny 要素が使用できる。

例 :

[型付き (バージョン1.3) データの転送]

有効データ	<ItemDataInteger ItemOID="ID.INT">1</ItemDataInteger>	
無効データ	<ItemDataAny ItemOID="ID.INT">text</ItemDataAny>	正しい
無効データ	<ItemDataInteger ItemOID="ID.INT">text</ItemDataInteger>	誤り

[型なし (バージョン1.2) データの転送]

有効データ	<ItemData ItemOID="ID.INT" Value="1"/>
無効データ	<ItemData ItemOID="ID.INT" Value="text"/>

3.1.4.1.1.1.2 ArchiveLayoutRef

本体 :

EMPTY (空)

属性 :

ArchiveLayoutO [oidref](#) [ArchiveLayout](#) への参照。

ID

親 :

[FormData](#)

form (フォーム) のデータを収集するために使用する保存用レイアウトへの参照を記述する。

更新により、NULL (ヌル) 値の ArchiveLayoutOID 属性も許される。

3.1.4.1.2 AuditRecord

本体 :

([UserRef](#), [LocationRef](#), [DateTimeStamp](#), [ReasonForChange?](#), [SourceID?](#))

属性：

EditPoint	(Monitoring DataManagement DBAudit)	(任意)	
UsedImputationMethod	(Yes No)	(任意)	
ID	<u>ID</u>	(任意)	AuditRecords 要素内に AuditRecord を含む場合、ID 属性を記述しなければならない。

親：

[FormData](#), [ItemData](#), [ItemGroupData](#), [StudyEventData](#), [SubjectData](#), [AuditRecords](#)

AuditRecord では、臨床データの作成、削除、修正に関する情報を伝える。この情報には、だれが行って、どこで、いつ、なぜ行ったが含まれる。

EditPoint 属性により、その行為が行われたデータ処理フェーズを識別し、UsedImputationMethod 属性により、Method (方法) を使用したかどうかを明示する。(注：ODM 1.3 では、新しく MethodDef 要素が導入され、ImputationMethod 要素は廃止予定となった。)

注：Monitoring (モニタリング) フェーズは、臨床医、試験実施施設人員、または試験モニターが関与したデータ収集および修正を含む。DataManagement (データ管理) フェーズは、内部データ処理の必要性 (例えば、導出データの更新やコード化)、およびデータベースのロック前に行ったすべての変更を含む。DBAudit (データベース監査) フェーズは、データベースのロック後に発生する。

AuditRecord の情報は、臨床データに対する変更を表すが、それ自体は臨床データではない。一部の臨床データの値は、その後のトランザクションによって常に変更される可能性があるが、履歴は変更できない (追加のみ可能)。

TransactionType 属性を (明示または継承のいずれかで) 持つ要素であれば、常に AuditRecord がなければならない。ただし、Snapshot (スナップショット) 型文書の転送では AuditRecord は無意味である。

TransactionType 属性と同様に、AuditRecord を持つことが許されていて、且つ、まだなにも記載されていない任意の下位要素では AuditRecord が継承される。

注：[Signature](#) と対比すること。

3.1.4.1.2.1 UserRef

本体：

EMPTY (空)

属性：

UserOID [oidref](#) [User](#) 定義への参照。

親：

[AuditRecord, Signature](#)

3.1.4.1.2.2 DateTimeStamp

本体 :

[datetime](#)

属性 :

NONE (無し)

親 :

[AuditRecord, Signature](#)

データの入力、修正、または署名を行った日時を記述する。この要素は、その後のコンピュータシステム間の転送では無く、その行為を最初に行った場合に適用する。

DateTimeStamp のデータ型は datetime (日時) である。ISO 8601 の Complete Representation (完全表記) に適合する値を記述しなければならない。partialDatetime (部分日時) または incompleteDatetime (不完全日時) は無効である。

3.1.4.1.2.3 ReasonForChange

本体 :

[text](#)

属性 :

NONE (無し)

親 :

[AuditRecord](#)

ユーザーから得たデータ変更理由を記述する。

3.1.4.1.2.4 SourceID

本体 :

[text](#)

属性 :

NONE (無し)

親 :

[AuditRecord](#)

送信元のシステム内にあるデータソースを識別する情報を記述する。この情報は、そのシステム内でのみ意味を持つ。

例 : ItemGroup (項目群) に付けた SourceID は、そのデータに対する送信元システムの内

部レコード OID として使用できる。

3.1.4.1.3 Signature

本体：

([UserRef](#), [LocationRef](#), [SignatureRef](#), [DateTimeStamp](#), [CryptoBindingManifest](#)?~~廃止予定~~)

属性：

ID [ID](#) (任意) Signatures 要素に Signature 要素を含める場合は、ID 属性が必要である。

親：

[FormData](#), [ItemData](#), [ItemGroupData](#), [StudyEventData](#), [SubjectData](#), [Signatures](#)

電子署名は臨床データの収集に適用する。これは、一部のユーザーがそのデータに対して法的責任を負うことを示す。[21 CFR Part 11](#)を参照のこと。この **Signature** により、署名者、署名場所、署名の意味（参照される **SignatureDef** による）、署名日時、および（デジタル署名の場合に）含まれるデータの暗号化されたハッシュ値を識別する。

電子署名は、それが付与される entity（構成要素）（典型的には form [フォーム]）に適用する。この署名は、署名時点でその entity（構成要素）内にあるすべての臨床データを対象とし、任意の下位 entity（構成要素）内にある臨床データもすべて対象に含まれる。そのため、ある form（フォーム）に対する署名は、その form（フォーム）、item group（項目群）、および item（項目）のレベルのすべての臨床データが対象となる。

このように定義するために、臨床データは、TransactionType の属性、Signature 要素、および AuditRecord 要素を ~~除く~~すべての属性および要素の内容を含む。

注：Signature は特定の時点のデータに適用する。Signature は修正できない。ただし、臨床データが変化した場合は、新しい署名を適用できる。

注：Signature 要素は、その時点で ODM の **要素**内に記述されている内容だけではなく、**entity（構成要素）**のすべての内容に適用する。Signature は（その時点での）内容に適用し、内容の変更には適用しない。

注：紙のフォーム上で変更によりイニシャルを記載し、日付を記入することは、監査証跡を維持する 1 つの方法である。これを 21 CFR Part 11 の電子署名（Signature 要素）と混同すべきではない。

注：[AuditRecord](#) および [ds:Signature](#) と対比すること。

注：CryptobindingManifest 要素は[廃止予定](#)となっている。

3.1.4.1.3.1 SignatureRef

本体：

EMPTY（空）

属性：

SignatureOID [oidref](#) [SignatureDef](#) への参照。

親 :

[Signature](#)

3.1.4.1.4 Annotation

本体 :

([Comment?](#), [Flag*](#))

属性 :

SeqNum	integer		
TransactionType	(Insert Update Remove Upsert Context)	(任意)	
ID	ID	(任意)	Annotations に Annotation 要素を含める場合は、ID 属性が必要である。

親 :

[Association](#), [FormData](#), [ItemData](#), [ItemGroupData](#), [StudyEventData](#), [SubjectData](#), [Annotations](#)

臨床データに関する一般的な注意を記述する。annotation (注釈) にコメントとフラグの両方を記述する場合は、フラグはコメントに関連したものにすべきである。

SeqNum 属性 (小さな正の整数) により、その親の entity (構成要素) 内の annotation (注釈) を一意に識別する。

TransactionType 属性が Remove (削除) でない限り、空の Annotation (コメントもフラグのないもの) は許されない。Update (更新) によって、annotation (注釈) 全体の値が置き換えられる。

3.1.4.1.4.1 Comment

本体 :

[text](#)

属性 :

SponsorOrSite	(Sponsor Site)	(任意)	コメントのソース。
----------------------	------------------	------	-----------

親 :

[Annotation](#)

臨床データに関して (翻訳されていない) コメントを自由に記述する。コメントが Sponsor (試験依頼者) または臨床試験の Site (実施施設) から寄せられる場合がある。

3.1.4.1.4.2 Flag

本体 :

([FlagValue](#), [FlagType](#)?)

属性 :

NONE (無し)

親 :

[Annotation](#)

臨床データに関して機械で処理可能な annotation (注釈) を記述する。

3.1.4.1.4.2.1 FlagValue

本体 :

[text](#)

属性 :

CodeListOID [oidref](#) [CodeList](#) 定義への参照。

親 :

[Flag](#)

フラグの値を記述する。この値の意味は、関連する **FlagType** によって典型的に決まる。実際の値は、参照される **CodeList** のメンバーでなくてはならない。

3.1.4.1.4.2.2 FlagType

本体 :

[name](#)

属性 :

CodeListOID [oidref](#) [CodeList](#) 定義への参照。

親 :

[Flag](#)

フラグ型を記述する。このフラグ型により、そのフラグの目的と意味が決まる。アプリケーションが異なれば、必要なフラグ型が異なることが予想される。実際の値は、参照される **CodeList** のメンバーでなくてはならない。

3.1.4.1.5 InvestigatorRef

本体 :

EMPTY (空)

属性 :

UserOID [oidref](#) [User](#) 定義への参照。

親 :

[SubjectData](#)

3.1.4.1.6 SiteRef

本体 :

EMPTY (空)

属性 :

LocationOID [oidref](#) [Location](#) 定義への参照。

親 :

[SubjectData](#)

3.1.4.2 AuditRecords

本体 :

([AuditRecord*](#))

属性 :

NONE (無し)

親 :

[ReferenceData](#), [ClinicalData](#)

ItemData[TYPE] 要素により参照される AuditRecord 要素を集めたものである。

3.1.4.3 Signatures

本体 :

([Signature*](#))

属性 :

NONE (無し)

親 :

[ReferenceData](#), [ClinicalData](#)

ItemData[TYPE] 要素により参照される Signature 要素を集めたものである。

3.1.4.4 Annotations

本体 :

([Annotation*](#))

属性：

NONE（無し）

親：

[ReferenceData](#), [ClinicalData](#)

ItemData[TYPE] 要素により参照される Annotation 要素を集めたものである。

3.1.5 Association

本体：

([KeySet](#), [KeySet](#), [Annotation](#))

属性：

StudyOID [oidref](#)

MetaDataVersion

nOID [oidref](#)

親：

[ODM](#)

Association により、1つの entity（構成要素）のみではなく、順序付けた entity（構成要素）のペアに annotation（注釈）を付けることが可能になる。1番目と2番目の KeySet を記述することで、注釈を付ける「リンク」の最初と最後を識別する。

StudyOID 属性および MetaDataVersionOID 属性には、その Annotation 内で CodeList の参照を解釈するためのコンテキストを記述する。これらの属性値が各 KeySet の意味に影響を与えることはない。

3.1.5.1 KeySet

本体：

EMPTY（空）

属性：

StudyOID [oidref](#)

SubjectKey [subjectKey](#) (任意)

StudyEventOID [oidref](#) (任意)

StudyEventRepeatKey [repeatKey](#) (任意)

FormOID [oidref](#) (任意)

FormRepeatKey [repeatKey](#) (任意)

ItemGroupOID [oidref](#) (任意)

ItemGroupRepeatKey [repeatKey](#) (任意)

atKey

ItemOID [oidref](#) (任意)

親 :

[Association](#)

KeySet は、study (試験)、subject (被験者)、study event (試験事象) などの 1 つの entity (構成要素) を参照する。特定の entity (構成要素) を規定するために必要とされる属性に必要であり、それ以外はすべて省略しなければならない。([臨床データキー](#) のセクションを参照のこと。)

4 デジタル署名

4.1 ds:Signature

本体 :

(SignedInfo, SignatureValue, KeyInfo?, Object*)

属性 :

xmlns CDATA

Id [ID](#) (任意)

親 :

[ODM](#)

ds:Signature 要素により、デジタル署名付き ODM 文書の完全性が保証される。この要素およびその下位要素は W3C の [XML Digital Signature Specification \(XML デジタル署名仕様書\)](#) を採用している。デジタル署名の意味については、この仕様書を参照のこと。

注 : [Signature](#) 要素は、臨床データセットの完全性と正確性に対して数人が責任を負っていることを表している。それとは対照的に、ds:Signature 要素は、その文書の内容に対する修正が確実に検出できるように、アプリケーションが使用する。

5 要素の索引 (参考)

[Address](#)

[AdminData](#)

[Alias](#)

[Annotation](#)

[Annotations](#)

[ArchiveLayout](#)

[ArchiveLayoutRef](#)

[Association](#)

[AuditRecord](#)

[AuditRecords](#)

[BasicDefinitions](#)

[Certificate](#)

[CheckValue](#)

[City](#)

[ClinicalData](#)

[CodeList](#)

[CodeListItem](#)

[CodeListRef](#)

[Comment](#)

[ConditionDef](#)

[Country](#)

[CryptoBindingManifest](#)

廃止予定 (1.2)

[DateTimeStamp](#)

[Decode](#)

[Description](#)

[DisplayName](#)

[Email](#)

[EnumeratedItem](#)

[ErrorMessage](#)

[ExternalCodeList](#)

[ExternalQuestion](#)

[Fax](#)

[FirstName](#)

[Flag](#)

[FlagType](#)

[FlagValue](#)

[FormalExpression](#)

[FormData](#)

[FormDef](#)

[FormRef](#)

[FullName](#)

[GlobalVariables](#)

[ImputationMethod](#)

[Include](#)

[InvestigatorRef](#)

[ItemData](#)

[ItemDataAny](#)

[ItemDataBase64Binary](#)

[ItemDataBase64Float](#)

[ItemDataBoolean](#)

[ItemDataDate](#)

[ItemDataDatetime](#)

[ItemDataDouble](#)

[ItemDataDurationDatetime](#)

[ItemDataFloat](#)

[ItemDataHexBinary](#)

[ItemDataHexFloat](#)

[ItemDataIncompleteDate](#)

[ItemDataIncompleteDatetime](#)

[ItemDataIncompleteTime](#)

[ItemDataInteger](#)

[ItemDataIntervalDatetime](#)

[ItemDataPartialDate](#)

[ItemDataPartialDatetime](#)

[ItemDataPartialTime](#)

[ItemDataString](#)

[ItemDataTime](#)

[ItemDataURI](#)

[ItemDef](#)

[ItemGroupData](#)

[ItemGroupDef](#)

[ItemGroupRef](#)

[ItemRef](#)

[KeySet](#)

[LastName](#)

[LegalReason](#)

[Location](#)

[LocationRef](#)

[LoginName](#)

[Meaning](#)

[MeasurementUnit](#)

[MeasurementUnitRef](#)

[MetaDataVersion](#)

[MetaDataVersionRef](#)

[MethodDef](#)

[ODM](#)

[Organization](#)

[OtherText](#)

[Pager](#)

[Phone](#)

[Picture](#)

[PostalCode](#)

[Presentation](#)

[Protocol](#)

[ProtocolName](#)

[Question](#)

[RangeCheck](#)

[ReasonForChange](#)

[ReferenceData](#)

[Role](#)

廃止予定 (1.2)

[Signature](#)

[SignatureDef](#)

[SignatureRef](#)

[Signatures](#)

[SiteRef](#)

[SourceID](#)

[StateProv](#)

[StreetName](#)

[Study](#)

[StudyDescription](#)

[StudyEventData](#)
[StudyEventDef](#)
[StudyEventRef](#)
[StudyName](#)
[SubjectData](#)
[Symbol](#)
[TranslatedText](#)
[User](#)
[UserRef](#)
[ds:Signature](#)

6 属性の索引 (参考)

この索引は、本書で使用している属性を、それを使用する要素と合わせてすべて列記している。

AnnotationID	ItemData[TYPE]
Archival	ODM
ArchiveLayoutOID	ArchiveLayoutRef
AsOfDateTime	ODM
AuditRecordID	ItemData[TYPE]
Category	StudyEventDef
Code	ExternalQuestion
CodedValue	CodeListItem
CodeListOID	CodeListRef, FlagType, FlagValue
CollectionExceptionConditionOID	StudyEventRef, FormRef, ItemGroupRef,
ID	ItemRef
Comment	ItemDef, ItemGroupDef
Comparator	RangeCheck
Context	Alias, FormalExpression
CreationDateTime	ODM
DataType	CodeList, ItemDef
Description	MetaDataVersion, ODM
Dictionary	ExternalCodeList, ExternalQuestion
Domain	ItemGroupDef
EditPoint	AuditRecord

EffectiveDate	MetaDataVersionRef
FileOID	ODM
FileType	ODM
FormOID	FormData, FormRef, KeySet
FormRepeatKey	FormData, KeySet
Granularity	ODM
ID	ODM, AuditRecord, Signature, Annotation
Id	ds:Signature
ImageType	Picture
ImputationMethodOID	廃止予定 (1.3)
IsNull	ItemData
IsReferenceData	ItemGroupDef
ItemGroupOID	ItemGroupData, ItemGroupRef, KeySet
ItemGroupRepeatKey	ItemGroupData, KeySet
ItemOID	ItemData, ItemRef, KeySet
KeySequence	ItemRef
Length	ItemDef
LocationOID	LocationRef, SiteRef
LocationType	Location
Mandatory	FormRef, ItemGroupRef, ItemRef, StudyEventRef
MeasurementUnitOID	MeasurementUnitRef, ItemData[TYPE]
MetaDataVersionOID	Association, ClinicalData, Include, MetaDataVersionRef, ReferenceData
MethodOID	ItemRef
Methodology	SignatureDef
Name	Alias, CodeList, ConditionDef, FormDef, ItemDef, ItemGroupDef, Location, MeasurementUnit, MetaDataVersion, MethodDef, StudyEventDef
ODMVersion	ODM
OID	ArchiveLayout, CodeList, ConditionDef, FormDef, ItemDef, ItemGroupDef, Location, MeasurementUnit, MetaDataVersion, MethodDef, Presentation, SignatureDef, Study,

	StudyEventDef, User
OrderNumber	FormRef, ItemGroupRef, ItemRef, StudyEventRef
Origin	ItemDef, ItemGroupDef
Originator	ODM
PdfFileName	ArchiveLayout
PictureFileName	Picture
PresentationOID	ArchiveLayout
PriorFileOID	ODM
Purpose	ItemGroupDef
ref	ExternalCodelist
Repeating	FormDef, ItemGroupDef, StudyEventDef
Role	ItemGroupDef, ItemRef
RoleCodeListOID	ItemRef
SASDatasetName	ItemGroupDef
SASFieldName	ItemDef
SASFormatName	CodeList
SDSVarName	ItemDef
SeqNum	Annotation
SignatureID	ItemData[TYPE]
SignatureOID	SignatureRef
SignificantDigits	ItemDef
SoftHard	RangeCheck
SourceSystem	ODM
SourceSystemVersion	ODM
SponsorOrSite	Comment
StudyEventOID	KeySet, StudyEventData, StudyEventRef
StudyEventRepeatKey	KeySet, StudyEventData
StudyOID	AdminData, Association, ClinicalData, Include, KeySet, MetaDataVersionRef, ReferenceData
SubjectKey	KeySet, SubjectData
TransactionType	Annotation, FormData, ItemData, ItemGroupData, StudyEventData, SubjectData
Type	StudyEventDef, MethodDef
UsedImputationMethod	AuditRecord

UserOID	InvestigatorRef, UserRef
UserType	User
Value	ItemData
Version	ExternalCodeList, ExternalQuestion
xml:lang	Presentation, TranslatedText
xmlns	ds:Signature

7 レビュー期間；ライセンス付与に関する義務、表明および保証；責任範囲

および免責事項

以下の内容は CDISC の IP Policy（知的所有権に関するポリシー）からの抜粋で、全文は http://www.cdisc.org/about/bylaws_pdfs/CDISCIPPolicy-FINAL.pdf で参照できる。

Section 5 より：

5 寄稿以外の必須特許の処理

5.1 レビュー期間／寄稿以外の必須特許に対するライセンス付与義務。

完全な文書として提案された本標準の草稿をそれぞれレビューする機会をすべての当事者に与え、寄稿以外の必須特許を確実に識別するため、本標準の草稿が完成する少なくとも 30 日（暦日）前にすべての当事者に電子メールで通知し、公開レビューのために CDISC ウェブサイト上に本標準の草稿を掲載する。レビュー期間中には、レビュー中の本標準の草稿を作成した CDISC グループ内の各当事者は（このレビューへ加わることを決めた他の関係者ととともに）、以下のことを行う：

(a) 本標準の草稿のレビューを行う；

(b) Section 3 に記載した無償の確約、およびそうでなければ妥当で公平な確約の下で作成された本標準の完成版のすべての実装者に対して当事者がライセンス供与を望まないような本標準の草稿を保護する寄稿以外の必須特許があれば、Section 5.2 に従って開示する；および

(c) (b) 項の下で開示される寄稿以外の必須特許があれば、少なくとも妥当で公平な (RAND) 条項（注：自身の寄稿を保護する必須特許は Section 3 に従って無償でライセンス供与しなければならない）に基づいて開示されるような特許のライセンス供与を当事者が確約する意思があるかどうかを明らかにしたライセンス供与承諾書をレビュー期間終了までに文書にて提出する。

Section 6.2 より

CDISC の特許免責事項。この標準の実装および遵守によって、特許権による保護対象物の使用が必要になる可能性がある。この標準の公表までは、何らかの特許またはそれに関連する何らかの特許権の存在または正当性に関して、CDISC はどのような立場もとらない。CDISC は CDISC 理事会を含めて、この標準を実装するため、あるいは注目を引く特許または特許請求の法的妥当性または範囲に関する問合せを行うために、ライセンスが必要になる可能性がある特許請求を特定する責任を負わない。

Section 9 より：

9.3 表明および保証。

本標準の開発の各当事者は、その当事者（またはその代理人）による寄稿時において、その知識および能力の限り、以下の内容を表明し、保証し、約束するものとみなされる。(a) 妥当な知的財産権を保持しているすべての司法権または管轄内で、すべての関連するライセンスをその寄稿に対して許諾する権利を保持または保有すること、(b) ここに含まれる許諾、承認、合意を行う当事者の能力は無制限であること、(c) 寄稿は、あらゆる寄稿、標準の草稿、標準の完成版、またはその実装の全体または一部に対して、本ポリシーで規定されている制約や要件に矛盾する追加の制約または要件を伴うライセンス付与に関する義務を課さないこと、または、そうした寄稿、標準の完成版、実装の全体または一部に次のいずれかを要求する場合があること：(i) ソースコード形態での公開または配布、(ii) 派生著作物を作成する目的でのライセンス付与（CDISC Intellectual Property Policy（「ポリシー」）の Section 4.2 の規定以外のもの）、(iii) ポリシーの Section 3、Section 5.1、Section 4.2 での規定以外の、無償での配布。任意の当事者または他の関係者により作成された寄稿が、何らかの寄稿、標準の草稿、標準の完成版、または実装の全体もしくは一部に対して、Section 9.3 に記載されたライセンス付与に関する 1 つ以上の義務を課すことを、特定の当事者が知った場合、その当事者はすぐに同様の内容を CDISC 代表者に通知しなければならず、CDISC 代表者もその内容を速やかにすべての当事者に通知しなければならない。

9.4 その他の保証/免責は一切提供されない。

すべての当事者は、CDISC Intellectual Property Policy の Section 9.3 で示されている内容を除き、すべての標準の草稿と完成版、ならびに標準の完成版および草稿に対するすべての寄稿は「現状のまま」提供され、明示的、黙示的、法令上、またはその他の保証は一切行われなことを承認し、さらに当事者、代理人、CDISC 代表者、CDISC 理事会、および CDISC が商品性、権利不侵害、あらゆる特定目的または想定された用途への適合性については一切の保証と、提案、標準の完成版または草稿、もしくは寄稿から生じる以外のその他のあらゆる保証を明確に否認することを承認する。

9.5 責任の範囲。

いかなる場合においても、CDISC またはそのあらゆる構成部分（CDISC 理事会、CDISC 代表、CDISC スタッフ、および CDISC メンバーを含むが、これに限定されない）は、本ポリシーまたは関連する協約から何らかの形で生じる利益の損失、使用権の喪失、直接的、間接的、偶発的、または特殊な損害について、それが契約、不法行為、保証、またはその他のいずれかに基づいているかどうか、また、それらの当事者がそうした損害の可能性を事前に通知していたかどうかに関わらず、他のあらゆる個人または法人に対して一切の責任を負わない。