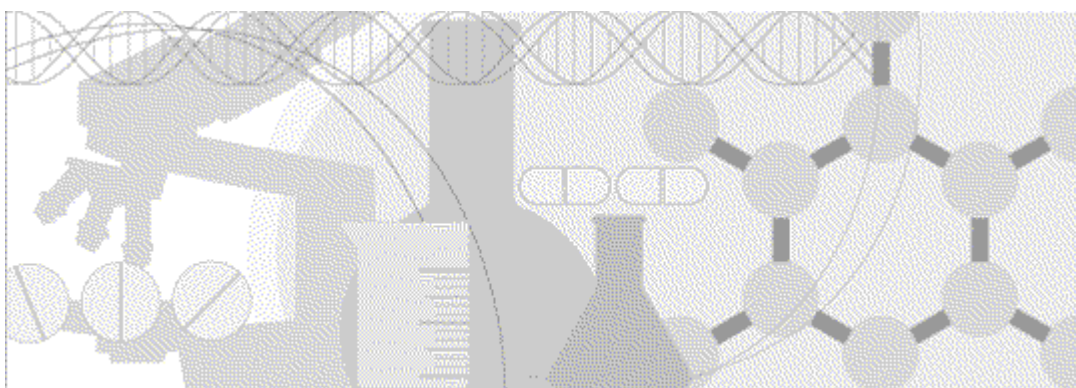


ユーティリティガイド

Oracle Health Sciences InForm 6.0



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

このプログラム（ソフトウェアおよびドキュメントの両方を含む）には専有情報が含まれています。このプログラムは使用および開示に関する制限を含むライセンス契約の下で提供され、著作権および特許、その他の知的財産権と工業所有権に関する法律により保護されています。独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定されている場合を除き、このプログラムのリバース エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

ここに含まれている情報は予告なしに変更される場合があります。当社は、このドキュメントに誤りがないことの保証は致しかねます。誤りを見つけた場合は、書面で当社に報告してください。ライセンス契約で明記的に許諾されている場合を除き、このプログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の通知が適用されます。

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S.

Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software -- Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核または航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかかえる目的で使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、当社は一切責任を負いかねます。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。Oracle は、第三者の Web サイトで提供されるコンテンツに関しては責任を負いかねます。かかるコンテンツの使用に関連する危険はすべて使用者の責任となります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。Oracle は (a) 第三者の製品またはサービスの質、もしくは (b) 製品またはサービスの提供および購入された製品またはサービスに関連する保証義務を含む第三者との契約の履行の責任を一切負いません。Oracle は 第三者との取引に起因する損失または損害には責任を負いません。

Oracle は、Oracle Corporation および／またはその関連会社の登録商標です。その他の名称は、それぞれの所有者の商標である可能性があります。

このドキュメントは、以前 Phase Forward Inc. 社により提供されていた材質、サービス、製品について記載している場合があります。材質、サービス、製品によっては提供を終了したものがあります。本書にかかる記載がある場合、Oracle およびその関連会社は一切その責任を負いかねます。

目次

このガイドについて	vii
このガイドの概要	viii
対象とする読者	viii
関連情報	ix
ドキュメント	ix
サポートが必要な場合	xii
PFConsole ユーティリティ	13
PFConsole ユーティリティの概要	14
コマンドラインからの PFConsole ユーティリティの 実行	15
例	15
PFConsole ユーティリティのスクリプト内での使用	16
MedML および MedML インストーラユーティリティ	17
MedML および MedML スキーマ	18
XML ファイルの作成	18
RSP ファイルの使用方法	21
RefName	22
レポーティングと分析の予約語	24
UUID	24
Global タグ	31
バージョン	41
MedML インストーラユーティリティ	45
MedML インストーラユーティリティの概要	45
MedML インストーラユーティリティによるスタディのセットアップ	45
妥当性チェック	46
MedML インストーラユーティリティの起動	47
MedML インストーラ画面について	47
MedML インストーラユーティリティの使用方法	49
コマンドラインからの MedML インストーラユーティリティの実行	51
MedML インストーラ出力メッセージ	52
エクスターナルデータマッピング	54
エクスターナルデータ形式とのマッピングの定義	54
エクスターナルマッピングターゲット	54
エクスターナルデータマッピングリファレンス	54
MedML スキーマリファレンス	93
Action	93
AttachRuleDepend	94
AttachRuleSet	96
ブラウザ	100
CalculatedControl	101
CheckboxControl	104
CodeTarget (ディクショナリ)	109
ContextItem (ディクショナリ)	110
Controlref	111
DateTimeControl	114
Dictionary (定義)	120
DocBody	122

Documentation	124
Elementref	128
Event	129
Eventref.....	130
ExecutionPlan	131
ExecutionPlanref.....	131
Formref	132
Form	133
FormSet.....	139
GroupControl.....	147
GroupType	150
HelpLink	150
HTMLTemplate	151
Item.....	152
ItemGroup.....	157
ItemGroupref.....	158
Itemref.....	159
ItemSet	161
ItemSetCell.....	164
KeyItemref.....	165
MedMLData	167
PFElement	168
PulldownControl.....	170
QueryGroup	173
RadioControl	174
ReportingGroup.....	179
Resource.....	181
ReviewStage.....	182
ReviewState.....	184
Rightref.....	187
RightsGroup	188
Right	190
Rule	190
Rulearg.....	194
Section	195
Sectionref	199
Sequence.....	200
SequenceType.....	202
SignatureGroup	202
SignCRF	206
SimpleControl	207
Site.....	209
SiteGroup.....	214
Sponsor	215
StudyVersionDoc.....	218
StudyVersionSite.....	218
StudyVersion	221
SVCriticalForm	223
SVCriticalItem	223
SysConfig	224
翻訳.....	229
Translations	231
Unit	231
Unitref.....	235
Update_Form_Section.....	236
Update_Section_Item.....	237

ユーザ	239
UserImage	243
Userref	244
VerbatimType	245
スクリプトオブジェクトリファレンス	247
Conversion オブジェクト	247
実行計画のオブジェクトとメソッド	247
Randomization オブジェクトとメソッド	253
ルールおよび計算のオブジェクトとメソッド	257

InForm データインポートユーティリティ 341

InForm データインポートユーティリティの概要	342
インポートユーティリティの構成	342
インポート方法	342
要注意点	343
XML ファイルのインポート	346
XML ファイルのインポートの概要	346
XML インポートファイルの作成	346
その他の InForm データインポートユーティリティ属性	364
MedML ファイルオプションを使用したインポートの実行	365
データおよびマップファイルを利用するインポート	367
データおよびマップファイルの指定	367
[InForm データおよびマップファイル] オプションを使用したインポートの実行	378
エラーファイルのチェック	380
日付と時間の妥当性チェック	381
コマンドラインからの InForm データインポート ユーティリティの実行	383
データインポートの拡張	386

InForm データエクスポート ユーティリティ 389

InForm データエクスポートユーティリティの概要	390
InForm データエクスポートユーティリティの出力オプション	390
InForm データエクスポートユーティリティの実行	391
CDD へのデータのエクスポート	392
CDD へのデータエクスポートの概要	392
CDD へのデータの移動	392
CDD データ用エクスポートの実行	392
コントロール名/値のペアのエクスポート	395
コントロール名/値のペアのエクスポートの概要	395
出力ファイルの形式	395
関連フォームの出力ファイル形式	396
例	396
コントロール名/値ペア用のエクスポートの実行	397
コマンドラインからの InForm データエクスポートユーティリティの実行	398
例	400
DEL ファイル形式	400

InForm Performance Monitor ユーティリティ 401

InForm Performance Monitor ユーティリティの概要	402
InForm Performance Monitor ユーティリティの起動	403
パフォーマンス統計の取得	404
特定のサブシステムのメッセージの表示	404
特定の InForm サーバのメッセージの表示	405
InForm Performance Monitor ユーティリティの出力オプション	406

InForm Performance Monitor ユーティリティデータの 管理.....	407
InForm Performance Monitor ユーティリティの使用例	408
ルールスクリプトの効率のテスト	408
SQL クエリパフォーマンスのレビュー	408

InForm Report Folder Maintenance ユーティリティ 411

InForm Report Folder Maintenance ユーティリティの 概要	412
複数のスタディまたはスポンサーのフォルダ構造	413
フォルダ構成の初期セットアップ	413
複数スタディのフォルダ構造	414
複数スポンサーと複数スタディのフォルダ構造	416
複数スポンサーまたは複数スタディ用のフォルダ構造のセットアップ	418
レポートパッケージのセットアップ	418
複数スタディまたはスポンサーの新規フォルダの作成	419
レポートフォルダのコピー	421

インポート用 XML ファイルの サンプルデータ 423

インポート用 XML ファイルのサンプルデータの概要	424
スクリーニングおよび症例登録データのインポート	425
新規症例の臨床データのインポート	426
既存の症例の臨床データの更新	427
[エントリの追加] アイテムセットの削除	428
[エントリの追加] アイテムセットの削除取り消し	429
予定外のビジットへのデータ追加	430
症例レコードの移動	431

このガイドについて

この序章の内容

このガイドの概要	viii
関連情報	ix
サポートが必要な場合	xii

このガイドの概要

ユーティリティガイドでは、以下のユーティリティについての情報とその使用手順を説明します。

- PFConsole ユーティリティ
- MedML インストーラユーティリティ
- InForm データインポートユーティリティ
- InForm データエクスポートユーティリティ
- InForm Performance Monitor ユーティリティ
- InForm Report Folder Maintenance ユーティリティ

このガイドには、MedML エlement、および InForm アプリケーション上のデータをインポートおよびエクスポートするために使用されるスクリプトオブジェクト、およびデータインポート XML のサンプルに関する参照情報も含まれています。

対象とする読者

このガイドは、InForm データベースに対するデータの入出力、既存の InForm データの修正、InForm プロセスとユーティリティの監視を行うスタディ設計者およびシステム管理者向けの手引きです。

関連情報

ドキュメント

すべてのドキュメントは Oracle Software Delivery Cloud (<https://edelivery.oracle.com>) および Download Center (<https://extranet.phaseforward.com>) から入手できます。

InForm ドキュメント	
ドキュメント	説明
リリースノート	リリースノートでは、今回のリリースで導入された拡張機能と修正された問題、アップグレードの注意事項、リリース履歴、およびその他の最新情報について説明しています。
既知の問題	<p>既知の問題では、このリリースに含まれる既知の問題に関する詳細な情報とその回避策（存在する場合）を説明しています。</p> <p>注： 既知の問題に関する最新のリストは、Oracle Extranet で入手できます。</p> <p>Extranet にサインインするには、www.phaseforward.com に移動して [Customer Login] をクリックします。電子メールアドレスとパスワードを入力して、[Known Issues] セクションに移動します。製品を選択して、検索条件を入力します。</p>
インストールガイド	『インストールガイド』では、ソフトウェアをインストールし、環境を InForm アプリケーションおよび Cognos 8 Business Intelligence 用に設定する方法について説明します。
スタディとレポート設定ガイド	『スタディとレポート設定ガイド』では、InForm スタディをセットアップし、レポートिंगと分析モジュールをスタディ用に設定するために必要なタスクを実行する方法について説明します。
ユーザガイド	<p>ユーザガイドでは、多言語でのスタディ、ユーザインタフェースのナビゲート方法、また、アプリケーションを使用してスタディ実施時に標準的な業務を遂行する方法などの詳細を含む、InForm アプリケーションの概要について説明しています。</p> <p>このドキュメントは、ドキュメンテーション CD および InForm ユーザインタフェースからも参照できます。</p>

InForm ドキュメント	
ドキュメント	説明
レポーティングと分析ガイド	<p>『レポーティングと分析ガイド』では、レポーティングと分析モジュールの概要について説明します。このガイドでは、レポーティングと分析のインタフェースの概要、アドホックレポーティング機能にアクセスする方法、およびレポートに使用できるスタディ管理と臨床データパッケージについて解説します。また、インストールに含まれる各標準レポートについても詳しく説明します。</p> <p>このドキュメントは、ドキュメンテーション CD から参照できます。</p>
ユーティリティガイド	<p>ユーティリティガイドでは、以下のユーティリティについての情報とその使用手順を説明します。</p> <ul style="list-style-type: none"> • PFConsole ユーティリティ • MedML インストラユーティリティ • InForm データインポートユーティリティ • InForm データエクスポートユーティリティ • InForm Performance Monitor ユーティリティ • InForm Report Folder Maintenance ユーティリティ <p>このガイドには、MedML エlement、および InForm アプリケーション上のデータをインポートおよびエクスポートするために使用されるスクリプトオブジェクト、およびデータインポート XML のサンプルに関する参照情報も含まれています。</p> <p>このドキュメントは、ドキュメンテーション CD から参照できます。</p>
Reporting Database Schema Guide	<p>『Reporting Database Schema Guide』では、レポーティングと分析データベーススキーマについて説明します。</p>
Portal 管理者ガイド	<p>『Portal 管理者ガイド』では、InForm Portal ソフトウェアのセットアップと、InForm Portal アプリケーションの設定および管理の手順について説明します。</p> <p>このドキュメントは、ドキュメンテーション CD および InForm Portal ユーザインターフェイスからも参照できます。</p>

InForm ドキュメント	
ドキュメント	説明
MedML インストーラ ユーティリティ オンラインヘルプ	<p>MedML インストーラユーティリティのオンラインヘルプでは、スタディコンポーネントを定義する XML を InForm データベースにロードするために使用される MedML インストーラユーティリティの使用方法を順を追って説明します。</p> <p>このガイドには、MedML エlement、および InForm アプリケーション上のデータをインポートおよびエクスポートするために使用されるスクリプトオブジェクト、およびデータインポート XML のサンプルに関する参照情報も含まれています。</p> <p>このドキュメントは、ユーザインタフェースからも参照できます。</p>
InForm データ エクスポート ユーティリティ オンラインヘルプ	<p>InForm データエクスポートユーティリティのオンラインヘルプでは、InForm データエクスポートユーティリティの使用方法について、順を追って説明します。InForm データエクスポートユーティリティは、InForm アプリケーションからデータを以下の形式にエクスポートするために使用されます。</p> <ul style="list-style-type: none"> 顧客定義データベース (CDD; Customer-Defined Database) コントロール名/値のペア <p>このドキュメントは、ユーザインタフェースからも参照できます。</p>
InForm データ インポート ユーティリティ オンラインヘルプ	<p>InForm データインポートユーティリティのオンラインヘルプでは、InForm データインポートユーティリティの使用方法について、順を追って説明します。InForm データインポートユーティリティは、InForm アプリケーションにデータをインポートするために使用されます。</p> <p>このドキュメントは、ユーザインタフェースからも参照できます。</p>

サポートが必要な場合

My Oracle Support にアクセスするとサポートを受けることができます。詳細については <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照するか、耳が不自由な方は <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

第 1 章

PFConsole ユーティリティ

この章の内容

PFConsole ユーティリティの概要.....	14
コマンドラインからの PFConsole ユーティリティの実行.....	15
PFConsole ユーティリティのスクリプト内での使用	16

PFConsole ユーティリティの概要

PFConsole ユーティリティによって、1 つの実行画面内にある各ユーティリティ（以下参照）のすべてのアクティビティを一元管理できます:

- InForm データインポートユーティリティ
- InForm データエクスポートユーティリティ
- MedML インストーラユーティリティ

注： コマンドラインからユーティリティを実行する場合は **PFConsole** ユーティリティを使用することを推奨します。コマンドラインウィンドウから個別に実行しないでください。

コマンドラインからの **PFConsole** ユーティリティの実行

以下のコマンドを使用して、PFConsole ユーティリティを開き、InForm ユーティリティを実行します:

```
pfconsole <application> -autorun <parameters>
```

ここで使用されている用語の説明は次のとおりです。

- *pfconsole*: PFConsole ユーティリティを起動します。
- *<application>*: には以下のいずれかを指定します。
 - *pfmminst* — MedML インストーラユーティリティ
 - *pfimport* — InForm データインポートユーティリティ
 - *pfexport* — InForm データエクスポートユーティリティ
- *-autorun* は、PFConsole の必須パラメータです。
- *<parameters>* は個々のユーティリティの変数です。

各ツールで用いる特定のパラメータについては、以下を参照してください。

- コマンドラインからの **MedML** インストーラユーティリティの実行 (51ページ)
- コマンドラインからの **InForm** データインポートユーティリティの実行 (383ページ)
- コマンドラインからの **InForm** データインポートユーティリティの実行 (398ページ)

注: すべてのコマンドラインアプリケーション (**MedML** インストーラユーティリティ、**InForm** データインポートユーティリティ、および **InForm** データエクスポートユーティリティ) は、デフォルトプロダクトロケールを使用します。デフォルトプロダクトロケールは **InForm** のインストール中、または **PFAdmin** コマンドで設定されます。

例

```
pfconsole pfmminst -trial pf206 -verbose -autorun -outfile text.log -xml filename.xml
```

PFConsole ユーティリティのスクリプト内での使用

複数のインポートまたはエクスポートを実行する場合、ユーティリティをバッチモードで実行するスクリプトを生成することができます。アプリケーションが完了するまでスクリプトを一時的に停止してから、次のコマンドに進むようにするには、次のコマンドを使用します。

```
start /wait <pfconsole command line>
```

ここで使用されている用語の説明は次のとおりです。

<pfconsole command line>: PFConsole ユーティリティを実行する際に指定するコマンド文字列です。

詳細については、「[コマンドラインからの PFConsole ユーティリティの実行](#)」(15ページ) を参照してください。

第 2 章

MedML および MedML インストーラ ユーティリティ

この章の内容

MedML および MedML スキーマ	18
MedML インストーラユーティリティ	45
エクスターナルデータマッピング	54
MedML スキーマリファレンス	93
スクリプトオブジェクトリファレンス	247

MedML および MedML スキーマ

XML ファイルの作成

XML ファイルの作成および変更には、メモ帳や TextPad などのテキストエディタを使用できます。

注： Oracle では、InForm インストールメディアの一部として提供されているサンプル XML ファイルを別の名前で保存し、それを編集して使用することを推奨しています。これらのファイルは、下記の場所にあります。¥¥<Installation_Directory>¥InForm ¥Sample_PFST50¥Admin

管理データ用の MedML コンポーネント定義の XML ファイルを作成するには、次の手順を実行します。

- 1 xml バージョン番号を含む最初の行を作成します。この文字列は、次のように小文字にする必要があります。

```
<?xml version="1.0"?>
```

- 2 MedMLData コンポーネントの開始タグと終了タグを追加して、xmlns 属性の値として適切なバージョン文字列を含めます。

```
<MEDMLDATA xmlns="PhaseForward-MedML-Inform4"> </MEDMLDATA>
```

- 3 MedMLData の開始タグと終了タグの間に、以下のリソースタイプの定義を入力します。

- 言語
- ブラウザ
- GIF、JPG、または text リソースファイル
- HTML テンプレート
- レポート
- ドキュメント
- StudyVersionDoc

- 4 スタディデータ定義を、以下の順でインストールします。

- a SysConfig
- b SequenceType
- c Sequence
- d Right
- e GroupType
- f User
- g UserImage
- h Site、該当する場合、各施設にどのユーザが属するかの参照を含む。
- i Sponsor、該当する場合、スポンサーユーザに属するユーザの参照を含む。

- j QueryGroup、各クエリグループのメンバであるユーザの参照を含む。
 - k SignatureGroup、各署名グループのメンバであるユーザの参照を含む。
 - l SiteGroup、各施設へのアクセス権を持つユーザの参照を含む。
 - m RightsGroup、グループを作成する権限の参照とその権限が割り当てられるユーザの参照を含む。
 - n SignCRF
- 5 スタディのプロトコルバージョンを指定するスタディバージョンを作成します。
 - 6 各施設が使用しているスタディバージョンを指定する StudyVersionSite 定義を作成します。

コンポーネントの定義

XML コンポーネント定義は、XML ファイルの基本的な構成単位です。それぞれのコンポーネント定義は、エレメントと属性から構成されます。コンポーネント定義は、互いに階層的な関係にあります（つまり、コンポーネントはチャイルドコンポーネントを持つことができます）。

エレメント

エレメントは、山形括弧（<>）で囲まれたコンポーネント名で構成されています。例：

```
<RIGHTSGROUP GROUPNAME="PF Admin Rights Group">
  <RIGHTREF RIGHT="Activate Site User"/>
  <RIGHTREF RIGHT="Deactivate Site User"/>
  <RIGHTREF RIGHT="Activate Sponsor User"/>
  <RIGHTREF RIGHT="Deactivate Sponsor User"/>
  <RIGHTREF RIGHT="Make user active"/>
  <RIGHTREF RIGHT="Create Sites"/>
  <USERREF USERNAME="admin"/>
</RIGHTSGROUP>
```

コンパウンドコンポーネント（チャイルドを持つコンポーネント）を定義するとき、ペアレントコンポーネントは開始と終了の両方のエレメントを持ち、チャイルドコンポーネントはスラッシュで終わる開始エレメントを持ちます。例：

```
<RIGHTSGROUP GROUPNAME="PF Admin Rights Group">
  <RIGHTREF RIGHT="Activate Site User"/>
  <RIGHTREF RIGHT="Deactivate Site User"/>
  <RIGHTREF RIGHT="Activate Sponsor User"/>
  <RIGHTREF RIGHT="Deactivate Sponsor User"/>
  <RIGHTREF RIGHT="Make user active"/>
  <RIGHTREF RIGHT="Create Sites"/>
  <USERREF USERNAME="admin"/>
</RIGHTSGROUP>
```

チャイルドのないコンポーネントには、終了エレメントは必要ありません。その代わり、コンポーネント定義は、各属性定義のあとに続けてスラッシュで完了し、山形括弧で閉じます。例：

```
<PFELEMENT
  REFNAME="name"
  LABEL="name"
  [LANGUAGE="name"]
  TYPE="STRING|INTEGER|FLOAT"
  VALUE="returned_value"/>
```

属性

属性は、コンポーネントをユニークに識別したり、オンライン上でどのように表示されるかを表します。コンポーネント定義において各属性が記述される場所は、コンポーネント名の後ろから、終了の括弧かスラッシュの前までです。属性のフォーマットは以下の通りです:

ATTRIBUTE_NAME = "attribute_value"、この場合

- ATTRIBUTE_NAME は MedML スキーマで定義された属性の名称。
- attribute_value はコンポーネント定義の属性に与えられた値。ダブルクォーテーション(")で値を囲います。

以下の例は、NA というコンポーネントに対する PFElement 定義の例です。

```
<PFELEMENT REFNAME="NA" LABEL="Not Applicable" TYPE="STRING" VALUE="NAElement"/>
```

REFNAME、**LABEL**、**TYPE**、**VALUE** の4つの必須属性はありますが、**LANGUAGE** というオプション属性は省略されています。**LANGUAGE** 属性は、ラベルがユーザインタフェースに表示されるよう、言語が混在する場合に指定されます。

チャイルドコンポーネント

コンポーネント定義においてチャイルドコンポーネントを使用することで、複数のコントロール定義がネストしているコントロールを定義し、コンパウンドコントロールを作成することができます。チャイルドコンポーネントを使用すると、チャイルドコンポーネント定義の中で過去に定義したコンポーネントを参照することで、複数の場所でコンポーネントを再利用できます。例えば、ドロップダウンリストを定義するために、1セットの選択肢をコンポーネントとして定義し、次にリストコントロールを定義し、そのリストコントロール定義には個々の選択肢をチャイルドコンポーネントとして含めます。

MedML スキーマにおいて、多くのチャイルドコンポーネント名が REF という文字で終わります (Elementref、Controlref など)。これは、以前に定義されたコンポーネント (別のコンポーネントの定義中でネストされている) への参照としての使用であることを表します。

コンパウンドコントロールの定義にチャイルドコンポーネントを含める場合、次の例で示すように、チャイルドコンポーネント定義はペアレントコンポーネントの開始エレメントに続けて、ペアレントコンポーネントの属性の後に記述します。

```
<QUERYGROUP GROUPNAME="CRA Query">
  <USERREF USERNAME="Kevin"/>
  <USERREF USERNAME="George"/>
</QUERYGROUP>
```

コンポーネント定義の順序

1 セットのコンポーネント定義を作成するときは下から順に作成します。チャイルドコンポーネントを定義してから、そのチャイルドコンポーネントを参照するコンポーネントを定義します。

MedML インストーラユーティリティを使用してコンポーネント定義をデータベースにロードする場合、コンポーネント定義が作成される順序が重要です。MedML インストーラユーティリティがコンパウンドコンポーネントを処理するとき、ペアレントコンポー

メント定義で参照するチャイルドコンポーネントがデータベースに存在していなければなりません。チャイルドコンポーネントがデータベースにない場合、MedML インストーラユーティリティはペアレントコンポーネントを作成できません。

「**XML ファイルの作成**」(18ページ)で説明されているコンポーネント順に従っている限り、スタディコンポーネント定義はどのように並べても構いません。また、複数のスタディのコンポーネント定義用 XML ファイルを作成することもできます。スタディのコンポーネント定義を複数のファイルに分ける場合、定められたコンポーネント順でロードしてください。

RSP ファイルの使用方法

XML コンポーネント定義を、実施中のスタディに挿入することができます。以下のファイルタイプのいずれかを挿入できます:

- コンポーネント定義を持つ XML ファイル
- 挿入する XML ファイル名を持つ応答ファイル (.RSP 拡張子)

応答ファイル (RSP 拡張子を持つテキストファイル) には、処理が必要な順に XML ファイル名がリストされます。RSP ファイルは、テキストエディタを使用して作成または変更できます。

例えば以下のような場合に複数のファイルを同時にロードするには、RSP ファイルを使用または編集します。

- スタディからコンポーネントのサブセットをロードする。
- 標準スタディコンポーネントのライブラリから基本コンポーネントをロードする。
- スタディのデプロイ準備をする (RSP ファイルがすべての関連ファイルを参照する必要があります)。
- スタディ運用中に変更する (RSP ファイルには変更済みのスタディコンポーネントのみを含める必要があります)。

RSP ファイルを使用して XML ファイルのパスを指定すると、スタディ用のファイルをすべてロードできます。

Windows のコマンドプロンプトから RSP ファイルを作成するには、XML ファイルを保存したディレクトリに移動し、次のように入力します:

```
dir *.xml /b > RSPfileName.rsp
```

このコマンドにより、コマンドを実行したフォルダに RSP ファイルが作成されます。作成された RSP ファイルには、現在のディレクトリ内のすべての XML ファイルがアルファベット順に含まれます。

RSP ファイルから XML ファイルを除外するには、RSP ファイルからそのファイルを削除するか、パウンド (#) を使用してコメントアウトします。

注: 依存関係がある場合、RSP ファイル内でファイルの順序の並べ替えが必要になることがあります。

RefName

RefName は、コンポーネント定義の識別に使用される名前です。[フォーム定義]画面の、フォームとそのコンポーネントの階層表示では、各コンポーネントが RefName で示されています。RefName は、ルール定義を付けるときに、特定のコントロールへのパスを一意に特定するのにも使用されます。

RefName の使用には、次のルールが適用されます：

- RefName は、最大 63 文字です。
- RefName は、スタディおよびコンポーネントタイプ内で一意である必要があります。たとえば、スタディ内で 2 つのアイテム定義に同じ RefName を割り当てることはできません。ただし CRF とセクションであれば同じ RefName でも構いません。
- RefName は大文字と小文字を区別します。
- RefName を含む定義がスタディデータベースにインストールされた後は、RefName を変更することはできません。これ以降、RefName を新規作成するには、オブジェクトを新規作成する必要があります。
- 次の語を RefName で使用すると、レポートニングと分析データベースのセットアップ中に障害が発生するので、使用しないでください。
 - CD_COUNT
 - AFROWID
 - SUBJECTID
 - SITEID
 - STUDYVERSIONID
 - SUBJECTVISITID
 - SUBJECTVISITREV
 - VISITID
 - VISITINDEX
 - FORMID
 - FORMREV
 - FORMINDEX
 - SUBJECTINITIALS
 - ITEMNEMONIC
 - VISITMNEMONIC
 - FORMMNEMONIC
 - VISITORDER
 - SITENAME
 - SITECOUNTRY

- SECTIONID
- ITEMSETID
- ITEMSETINDEX
- ITEMSETIDX
- DELETEDITEM
- DELETEDFORM
- FORMIDX

レポートिंगと分析の予約語

以下の予約語は、Central Designer アプリケーションのアイテムの Short Question には使用できません。Short Question テキストは、レポートिंगと分析モジュールの Itemset Column Header となります。以下の用語を使用すると、レポートिंगと分析臨床モデルを作成できなくなります。

- 症例イニシャル
- 症例番号
- ビジット略称
- フォーム略称
- 施設略称
- ビジットインデックス
- ビジット順
- ビジット日
- 削除フォーム
- CREATEDBYUSERID
- CREATEDDATETIME
- MODIFIEDBYUSERID
- MODIFIEDDATETIME
- 施設名
- 施設の国名
- SUBJECTID
- SITEID
- SUBJECTVISITID
- VISITID
- VISITINDEX
- FORMID
- FORMREV
- FORMINDEX
- CD_COUNT

UUID

Universal Unique Identifiers (UUID) は、全てのサーバ、データベースおよびスタディにおいて、コンポーネントをユニークに識別できるようにします。

多くの場合、UUID 属性の割り当てはオプションです。しかし、ある目的のため、InForm

アプリケーションは特定の既定の UUID を必要とします。たとえば、症例番号を InForm アプリケーションに自動発行させるのではなく、ユーザが定義できるようにするには、症例番号のデータ入力フィールドを構成するフォームセット、フォーム、セクション、アイテムそしてテキストコントロールの定義に既定の UUID を使用する必要があります。

UUID を変更するには、MedML インストーラを nonstrict モードで使用して変更をインストールする必要があります。詳細については、「**MedML インストーラ画面について**」(47ページ) を参照してください。

注：InForm アプリケーションと **MedML** インストーラユーティリティは、**UUID** の英字を大文字に変換します。

次の場合は、既定の UUID を必要とします。

- 共通フォーム
- ビジット日時
- 症例登録フォーム
- グループタイプ
- 症例 ID フォーム
- 症例イニシャルアイテム
- 症例番号アイテム
- 無作為割付
- 規制ドキュメント
- 繰り返しビジット
- スクリーニングフォーム
- シーケンス
- スタディ完了
- ビジットレポート

共通 CRF

共通 CRF は、複数のビジットで使用可能であり、累積されたすべてのデータを、CRF が含まれる各ビジットで表示します。

共通 CRF を定義するには、StudyVersion 定義の COMMONCRF フォームセットにその CRF を含めます。

注：共通 CRF を定義し、その **CRF** に症例のデータを追加した後は、スタディバージョンの変更中にフォーム タイプを通常の **CRF** に変更しないでください。同様に、フォームに症例データが含まれている場合は、通常の **CRF** を共通 **CRF** に変更しないでください。変更するとデータが失われます。

共通フォームをビジットの最初のフォームとして使用しないでください。フォームが他

のビジットで再利用される場合、ビジット日時コントロールでは各ビジットの個別の日付を認識することができません。

既存の共通 CRF と同じデータを取り込む通常 CRF を作成する必要がある場合、共通 CRF とは異なる RefName を持つ別の FORM 定義として CRF を作成し、それをスタディバージョン定義の中の該当する VISIT フォームセットに追加してください。

共通 CRF を定義するには、CRF が含まれている COMMONCRF フォームセットの定義の中で次の UUID を使用します。

コンポーネント	UUID
共通フォーム	9d6bbc5d-5811-11d2-8065- 00a0c9af7674

ビジット日時

ビジット日時アイテムは、各ビジットの最初のフォームに表示されます。予定もしくは予定外のビジットをセットアップするには、以下の UUID を使用して、ビジット日時を取り込むためのセクション、アイテム、日時コントロールを作成する必要があります。

フォームコンポーネント	UUID
日時コントロール	BD991BC0-B0A4-11D2-80E3-00A0C9AF7674
アイテム	BD991BBF-B0A4-11D2-80E3-00A0C9AF7674
セクション	BD991BBE-B0A4-11D2-80E3-00A0C9AF7674

注： 共通 **CRF** をビジットの最初のフォームとして使用しないでください。フォームが他のビジットで再利用されると、ビジットの日付コントロールでは各ビジットの個別の日付を認識することができません。

症例登録フォーム

コンポーネント	UUID
症例登録フォームセット	d882ce3a-0f42-11d2-a419- 00a0c963e0ac
症例登録フォーム	d882ce3b-0f42-11d2-a419- 00a0c963e0ac
[症例番号] データ入力アイテムを持つ 症例登録フォームセクション	abcfa388-223a-11d2-a426- 00a0c963e0ac

さらに、症例登録フォームに編集可能な症例番号アイテムが含まれている場合は、そのアイテムは、既定の UUID を必要とします。詳細については、「**症例番号アイテム**」(28ページ)を参照してください。

グループタイプ

GROUPTYPE エレメントは InForm アプリケーションで使用されている管理グループの 4 つのタイプ（クエリ、権限、署名、施設）をそれぞれ定義します。次の UUID は、GroupTypes の定義に必要です：

コンポーネント	UUID
クエリ	AC44A6E1-112E-11d2-8BED-0060082DE9D5
権限	FA3C6201-112E-11d2-8BED-0060082DE9D5
署名	002E58C1-112F-11d2-8BED-0060082DE9D5
施設	A4D7B9A1-112E-11d2-8BED-0060082DE9D5

症例 ID フォーム

症例登録が完全に終了したあと、ユーザが症例 ID を変更することができるようにするには、症例 ID フォームを使用します。症例 ID フォームは以下の UUID を必要とします。

コンポーネント	UUID
症例 ID フォームを含むビジット	03B0D5D8-7F2C-11D2-A728-00A0C977C64B
症例 ID フォーム	06702B62-7ED6-11D2-A728-00A0C977C64B
症例 ID セクション	3D25EE4B-7F1B-11D2-A728-00A0C977C64B
次のアイテムのいずれか、または両方： <ul style="list-style-type: none"> 症例登録フォーム上で使用される症例番号アイテム イニシャル変更アイテム 	症例番号アイテム： 3D25EE4C-7F1B-11D2-A728-00A0C977C64B イニシャル変更アイテム： D959FE72-7F1C-11D2-A728-00A0C977C64B
次のテキストボックスコントロールのいずれか、または両方： <ul style="list-style-type: none"> 症例登録フォームで使用される症例番号アイテムがセクションに含まれている場合の症例番号テキストボックスコントロール イニシャル変更アイテムがセクションに含まれている場合のイニシャル変更テキストボックスコントロール 	症例番号コントロール： 433DAFF6-7F1C-11D2-A728-00A0C977C64B イニシャル変更コントロール： 433DAFF7-7F1C-11D2-A728-00A0C977C64B

症例イニシャルアイテム

症例イニシャルアイテム定義は、スクリーニングフォームが必要です。症例イニシャル

アイテムを定義するには、以下の UUID を使用します:

フォームコンポーネント	UUID
症例イニシャルアイテム	aeb64f16-127c-11d2-a41c-00a0c963e0ac
症例イニシャルテキスト コントロール	aeb64f17-127c-11d2-a41c-00a0c963e0ac
生年月日アイテム	96cae359-126c-11d2-a41c- 00a0c963e0ac
生年月日コントロール	40ace712-217c-11d2-a425- 00a0c963e0ac
スクリーニング日アイテム	96cae356-126c-11d2-a41c- 00a0c963e0ac
スクリーニング日コントロール	96cae357-126c-11d2-a41c- 00a0c963e0ac

症例番号アイテム

症例番号アイテム定義は症例登録フォームでのオプションアイテムです。ユーザによる症例番号の編集を許可する場合、以下の UUID を使用して症例番号アイテムを定義します。

フォームコンポーネント	UUID
症例番号アイテム	3d25ee4c-7f1b-11d2-a728- 00a0c977c64b
症例番号テキストコントロール	433daff6-7f1c-11d2-a728-00a0c977c64b
[症例番号] データ入力アイテム を持つ症例登録フォーム セクション	abcfa388-223a-11d2-a426- 00a0c963e0ac

無作為割付

無作為割付番号を生成するように InForm ソフトウェアをセットアップするには、無作為割付番号が表示される CRF 上に既定のコントロール、アイテム、およびセクションを含める必要があります。これらのコンポーネント定義では、以下の UUID を使用します。

フォームコンポーネント	UUID
無作為計算コントロール	DC2EB0BF-4F12-11d2-9319- 00A0C9769A13
DRUGKIT アイテム	52AF1207-4F13-11d2-9319- 00A0C9769A13
DRUGKITSECTION セクション	C0482B37-4F13-11d2-9319- 00A0C9769A13

規制ドキュメント

規制ドキュメント CRF では、規制ドキュメントアイテムが必要です。規制ドキュメントの定義には、以下の UUID が必要です:

コンポーネント	UUID
規制ドキュメントフォームセット	PF_UUID_REGDOCS_FORMSET
規制ドキュメントチェックフォーム	PF_UUID_REGDOCS_FORM
規制ドキュメントセクション	PF_UUID_REGDOCS_SECTION

繰り返しビジット

REPEATING 属性が true にセットされた FORMSET 定義によって、予定外のビジットを定義できます。ビジットをセットアップするには、以下の UUID を使用して、ビジット日時ごとに区別されるセクション、アイテム、および日時コントロールを作成する必要があります。

フォームコンポーネント	UUID
Date/TimeControl	BD991BC0-B0A4-11D2-80E3-00A0C9AF7674
Item	BD991BBF-B0A4-11D2-80E3-00A0C9AF7674
Section	BD991BBE-B0A4-11D2-80E3-00A0C9AF7674

スクリーニング CRF

スクリーニング CRF の定義には、以下の既定の UUID が必要です:

フォームコンポーネント	UUID
スクリーニングフォームセット	D882CE38-0F42-11D2-A419-00A0C963E0AC
[スクリーニング] フォーム	d882ce39-0f42-11d2-a419- 00a0c963e0ac
症例イニシャル、生年月日、スクリーン日を含むスクリーニングフォームセクション	96cae354-126c-11d2-a41c- 00a0c963e0ac

シーケンス

InForm ソフトウェアは、スタディの進行中、症例登録、クエリ、無作為割付およびスクリーニングの番号を自動的に発行します。これらの番号は、シーケンス定義に従って発行されます。必須の UUID が、症例登録、クエリ、無作為割付、およびスクリーニング番号のシーケンスを定義します。

フォームコンポーネント	UUID
症例登録	eb75b898-078b-11d2-a417-00a0c963e0ac
無作為割付	4F4A0246-5009-11d2-931C-00A0C9769A13
スクリーニング	f7f1b3b8-0b5c-11d2-a418-00a0c963e0ac

スタディ完了

スタディ完了 CRF は、症例の最後の訪問日と、スタディを完了したかどうかを記録します。症例がスタディを完了していない場合、スタディ完了 CRF に未完了の理由を記録します。スタディ完了 CRF の定義には次の UUID を使用します。

注： スタディ完了 CRF の UUID は、スタディのバージョンとは関係なく、スタディのすべての症例に適用されます。したがって、新しいスタディバージョンを作成してスタディ完了のメタデータを変更することはできません。

コンポーネント	UUID
スタディ完了 CRF が含まれるビジット	F4699051-69E2-11D2-8FB5-00A0C977C66A
スタディ完了 CRF	7314A6A5-316E-11d2-8F9A-00A0C977C66A
完了ステータスを示すコントロール	PF_SC_COMPLETECTL
スタディ完了または未完了の値および表示テキストを定義するエレメントオブジェクト	次のエレメントオブジェクトは両方とも、スタディ定義に存在する必要があります。 <ul style="list-style-type: none"> PF_SC_STUDY_COMPLETE — 症例がスタディを完了したことを示します。 PF_SC_STUDY_INCOMPLETE — 症例がスタディを完了しなかったことを示します。
症例がスタディを脱落した理由を示すコントロール	PF_SC_REASONCTL

コンポーネント	UUID
のレポートと分析 モジュールで非完了理由の 内部値を列見出しの テキストにマッピングする テキストリソース	<p>PF_SC_REASONCTL_ <i>internal_reason_value</i></p> <p><i>Internal_reason_value</i> は次のいずれかです。</p> <ul style="list-style-type: none"> 脱落理由を指定するコントロールに含まれている シンプルコントロールまたはプルダウンコントロール の要素オブジェクトについて定義された Value プロパティ。 脱落理由を指定するコントロールに含まれている その他のタイプの下位コントロールについて定義さ れた Selection Value プロパティ。 Selection Value プロパティが定義されていない下位 コントロールのデフォルト値。デフォルト値は、 InForm アプリケーションによって割り当てられ、 形式は <code>!pf! control_DBUID_path</code> となります。

ビジットレポート

ビジットレポートフォームにはビジットレポートアイテムが必要です。ビジットレポートの定義には、以下の UUID が必要です。

コンポーネント	UUID
ビジットレポートフォームセット	PF_UUID_VISITREPORT_FORMSET
ビジットレポートフォーム	PF_UUID_VISITREPORT_FORM

Globalタグ

<!-- --> (Comment)

コメント要素は、角括弧でそれを囲んで、XML ファイル中にコメントを挿入できます。コメントを始めるために、コメントテキストの前に<!文字を入れてください。コメントを終えるには、コメントテキストに引き続き-->文字を続けてください。

別の要素の中にコメントを入れることはできません。

例

```
<?xml version="1.0"?>
<MEDMLDATA>
<!-- Define PFElements for Size pulldown -->

<PFELEMENT REFNAME="SMALL" LABEL="Small" TYPE="INTEGER"VALUE="1"/>
<PFELEMENT REFNAME="MEDIUM" LABEL="Medium" TYPE="INTEGER"VALUE="2"/>
<PFELEMENT REFNAME="LARGE" LABEL="Large" TYPE="INTEGER"VALUE="3"/>

<!-- Define SimpleControls -->

<SIMPLECONTROL REFNAME="MALE" NAME="MALE">
```

```

<ELEMENTREF REFNAME="MALE"/>
</SIMPLECONTROL>

<SIMPLECONTROL REFNAME="FEMALE" NAME="FEMALE">
  <ELEMENTREF REFNAME="FEMALE"/>
</SIMPLECONTROL>

```

HTML フォーマットタグ

InForm アプリケーションは次のフォーマットタグをサポートします。スタディプロトコル、CRF ヘルプ、CRF での質問および特記事項、およびルールヘルプなど、どんなテキストベースのスタディコンポーネント定義においても、タグの使用が可能です。

山形括弧 (< および >) は MedML インストーラユーティリティでは使用できないため、MedML ファイルから InForm アプリケーションにデータをインストールする場合は、HTML エスケープ文字を使用する必要があります。

詳細については、「**HTML 特殊文字**」(33ページ) および「**使用禁止文字**」(35ページ)を参照してください。

対象となる書式設定	使用するタグ	HTML タグの代用タグ
太字		
		
改行	
	
テキストを中央揃えに設定	<CENTER>	<CENTER>
	</CENTER>	</CENTER>
斜体テキスト	<I>	<I>
	</I>	</I>
アイテムリスト		
		
番号付リスト		
		
パラグラフ	<P>	<P>
	</P>	</P>
整形済テキスト：たとえば、コンピュータ出力	<PRE>	<PRE>
	</PRE>	</PRE>
取り消し線が引かれたテキスト	<S>	<S>
	</S>	</S>
取り消し線が引かれたテキスト	<STRIKE>	<STRIKE>
	</STRIKE>	</STRIKE>
下付き文字	<SUB>	<SUB>
	</SUB>	</SUB>

対象となる書式設定	使用するタグ	HTML タグの代用タグ
上付き文字	<SUP> </SUP>	^{ }
等幅フォント	<TT> </TT>	<TT> </TT>
下線付きテキスト	<U> </U>	<U> </U>
順序のない (箇条書き) リスト	 	

HTML 特殊文字

HTML 仕様は特殊文字を指定するための多くの文字列を含んでいます。スタディコンポーネント定義ファイルに HTML 特殊文字を含めると、MedML インストーラユーティリティによって文字がデータベースに渡され、InForm アプリケーションのフォームレンダリングコンポーネントによって取得され処理されます。

PDF 出力形式(InForm データエクスポートユーティリティが印刷可能な CRF をエクスポートする形式) では、以下の特殊文字定義がサポートされています。

文字	エンティティ名	数字コード	説明
"	クォーテーション	"	"
&	アンパサンド	&	&
<	小なり	<	<
>	大なり	>	>
	空白	 	
。	逆感嘆符	¡	!
¢	セント	¢	¢
£	ポンド	£	£
¤	一般通貨	¤	¤
¥	円	¥	¥
§	セクション	§	§
¨	ウムラウト(分音記号)	¨	¨
©	copyright	©	©
^a	女性序数	ª	ª
«	左角引用符	«	«

文字	エンティティ名	数字コード	説明
¬	否定記号	¬	¬
-	ソフトハイフン	­	­
®	登録商標	®	®
＝	マクロ記号	¯	¯
²	上付き 2	²	²
¶	段落記号	¶	¶
¸	セディラ	¸	¸
º	男性序数	º	º
»	右角引用符	»	»
¹ / ₄	4 分の 1	¼	¼
³ / ₄	4 分の 3	¾	¾
¿	逆疑問符	¿	¿
Â	大文字 A、サーカムフレックス アクセント	Â	Â
Ã	大文字 A、チルダ	Ã	Ã
Ä	大文字 A、分音記号または ウムラウト	Ä	Ä
Å	大文字 A、リング (オングストローム)	Å	Å
Æ	大文字 AE、二重母音 (リゲチャ)	Æ	Æ
Ç	大文字 C、セディラ	Ç	Ç
Ê	大文字 E、サーカムフレックス アクセント	Ê	Ê
Ë	大文字 E、分音記号またはウムラウト	Ë	Ë
Ì	大文字 I、グレーブアクセント	Ì	Ì
Í	大文字 I、アキュートアクセント	Í	Í
Ï	大文字 I、分音記号またはウムラウト	Ï	Ï
Ð	大文字 Eth、アイスランド語	Ð	Ð
Ñ	大文字 N、チルダ	Ñ	Ñ
Ø	大文字 O、スラッシュ	Ø	Ø
Ù	大文字 U、グレーブアクセント	Ù	Ù
Ú	大文字 U、アキュートアクセント	Ú	Ú

文字	エンティティ名	数字コード	説明
Ü	大文字、分音記号またはウムラウト	Ü	Ü
Ý	大文字 Y、アキュートアクセント	Ý	Ý
Þ	大文字 THORN、アイスランド語	Þ	Þ
ß	小文字 シャープ s、ドイツ語 (sz リゲチャ)	ß	ß
ã	小文字 a、チルダ	ã	ã
ä	小文字 a、分音記号またはウムラウト	ä	ä
å	小文字 a、リング	å	å
æ	小文字 æ 二重母音 (リゲチャ)	æ	æ
ç	小文字 c、セディラ	ç	ç
ð	小文字 eth、アイスランド語	ð	ð
ñ	小文字 n、チルダ	ñ	ñ
ô	小文字 o、サーカムフレックス アクセント	ô	ô
õ	小文字 o、チルダ	õ	õ
ö	小文字 o、分音記号またはウムラウト	ö	ö
ø	小文字 o、スラッシュ	ø	ø
ù	小文字 u、グレーブアクセント	ù	ù
ú	小文字 u、アキュートアクセント	ú	ú
û	小文字 u、サーカムフレックス アクセント	û	û
ü	小文字 u、分音記号またはウムラウト	ü	ü
ý	小文字 y、アキュートアクセント	ý	ý
þ	小文字 thorn、アイスランド語	þ	þ
ÿ	小文字 y、分音記号またはウムラウト	ÿ	ÿ

使用禁止文字

レンダリングおよびその他のフォーマットの問題が発生しないようにするため、以下の特殊文字を使用しないようにしてください。

文字	該当箇所
----	------

文字	該当箇所
二重引用符	<ul style="list-style-type: none"> データ入力テキストボックス フォームタイトル 質問テキスト クエリ回答テキスト クエリテキスト
単一引用符	<ul style="list-style-type: none"> 質問テキスト クエリ回答テキスト クエリテキスト 施設名
アポストロフィ (')	<ul style="list-style-type: none"> フォームタイトル 質問テキスト
\ または \\	これらの文字は使用しないでください。
> または<	<p>すべて。代わりに、次の HTML エスケープ文字を使用してください。</p> <ul style="list-style-type: none"> > -- &#62; または&gt; < -- &#60; または&lt;
上付きおよび下付き フォーマット仕様	プルダウンコントロールで使用するエレメント

また、Microsoft Word アプリケーションは文字を Unicode 形式に変更する可能性があるため、Microsoft Word アプリケーションからテキストボックスやクエリテキストにコピーアンドペーストしないでください。

ロケールコード

次の表は、**TRANSLATION** 属性の定義時に翻訳テキストの指定に使用するロケールコードのリストです。Central Designer アプリケーションで使用されるコードは、カルチャ固有のコードです。次の表のカルチャ固有のコードでは、言語とカルチャの間にハイフンが挿入されています。

詳細については、「**翻訳**」(229ページ)を参照してください。

カルチャと言語コード	カルチャ
af-ZA	アフリカーンス語 (南アフリカ)
sq-AL	アルバニア語 (アルバニア)
ar-DZ	アラビア語 (アルジェリア)

カルチャと言語コード	カルチャ
ar-BH	アラビア語 (バーレーン)
ar-EG	アラビア語 (エジプト)
ar-IQ	アラビア語 (イラク)
ar-JO	アラビア語 (ヨルダン)
ar-KW	アラビア語 (クウェート)
ar-LB	アラビア語 (レバノン)
ar-LY	アラビア語 (リビア)
ar-MA	アラビア語 (モロッコ)
ar-OM	アラビア語 (オマーン)
ar-QA	アラビア語 (カタール)
ar-SA	アラビア語 (サウジアラビア)
ar-SY	アラビア語 (シリア)
ar-TN	アラビア語 (チュニジア)
ar-AE	アラビア語 (アラブ首長国連邦)
ar-YE	アラビア語 (イエメン)
hy-AM	アルメニア語 (アルメニア)
az-Cyrl-AZ	アゼリー語 (アゼルバイジャン、キリルアルファベット)
az-Latn-AZ	アゼリー語 (アゼルバイジャン、ラテン語)
eu-ES	バスク語 (バスク)
be-BY	ベラルーシ語 (ベラルーシ)
bg-BG	ブルガリア語 (ブルガリア)
ca-ES	カタロニア語 (カタロニア)
zh-HK	中国語 (香港、中国)
zh-MO	中国語 (マカオ)
zh-CN	中国語 (中国)
zh-Hans	中国語 (簡素中国語)
zh-SG	中国語 (シンガポール)
zh-TW	中国語 (台湾)
zh-Hant	中国語 (繁体字中国語)
hr-HR	クロアチア語 (クロアチア)
cs-CZ	チェコ語 (チェコ共和国)

カルチャと言語コード	カルチャ
da-DK	デンマーク語 (デンマーク)
dv-MV	ディヴェヒ語 (モルディヴ)
nl-BE	オランダ語 (ベルギー)
nl-NL	オランダ語 (オランダ)
en-AU	英語 (オーストラリア)
en-BZ	英語 (ベリーズ)
en-CA	英語 (カナダ)
en-029	英語 (カリブ海域)
en-IE	英語 (アイルランド)
en-JM	英語 (ジャマイカ)
en-NZ	英語 (ニュージーランド)
en-PH	英語 (フィリピン)
en-ZA	英語 (南アフリカ)
en-TT	英語 (トリニダードトバゴ)
en-GB	英語 (英国)
en-US	英語 (米国)
en-ZW	英語 (ジンバブエ)
et-EE	エストニア語 (エストニア)
fo-FO	フェロー語 (フェロー諸島)
fa-IR	ペルシア語 (イラン)
fi-FI	フィンランド語 (フィンランド)
fr-BE	フランス語 (ベルギー)
fr-CA	フランス語 (カナダ)
fr-FR	フランス語 (フランス)
fr-LU	フランス語 (ルクセンブルグ)
fr-MC	フランス語 (モナコ)
fr-CH	フランス語 (スイス)
gl-ES	ガリシア語 (スペイン)
ka-GE	グルジア語 (グルジア)
de-AT	ドイツ語 (オーストリア)
de-DE	ドイツ語 (ドイツ)

カルチャと言語コード	カルチャ
de-LI	ドイツ語 (リヒテンシュタイン)
de-LU	ドイツ語 (ルクセンブルグ)
de-CH	ドイツ語 (スイス)
el-GR	ギリシア語 (ギリシア)
gu-IN	グジャラート語 (インド)
he-IL	ヘブライ語 (イスラエル)
hi-IN	ヒンディー語 (インド)
hu-HU	ハンガリー語 (ハンガリー)
is-IS	アイスランド語 (アイスランド)
id-ID	インドネシア語 (インドネシア)
it-IT	イタリア語 (イタリア)
it-CH	イタリア語 (スイス)
ja-JP	日本語 (日本)
kn-IN	カナラ語 (インド)
kk-KZ	カザフ語 (カザフスタン)
kok-IN	コンカニ語 (インド)
ko-KR	朝鮮語 (韓国)
ky-KG	キルギスタン語 (キルギスタン)
lv-LV	ラトヴィア語 (ラトヴィア)
lt-LT	リトアニア語 (リトアニア)
mk-MK	マケドニア語 (マケドニア)
ms-BN	マレー語 (ブルネイ)
ms-MY	マレー語 (マレーシア)
mr-IN	マラーティー語 (インド)
mn-MN	モンゴル語 (モンゴル)
nb-NO	ノルウェー語 (ブークモール、ノルウェー)
nn-NO	ノルウェー語 (ニーノシク、ノルウェー)
pl-PL	ポーランド語 (ポーランド)
pt-BR	ポルトガル語 (ブラジル)
pt-PT	ポルトガル語 (ポルトガル)
pa-IN	パンジャブ語 (インド)

カルチャと言語コード	カルチャ
ro-RO	ルーマニア語 (ルーマニア)
ru-RU	ロシア語 (ロシア)
sa-IN	サンスクリット語 (インド)
sr-Cyrl-CS	セルビア語 (セルビア、キリルアルファベット)
sr-Latn-CS	セルビア語 (セルビア、ラテン語)
sk-SK	スロヴァキア語 (スロヴァキア)
sl-SI	スロヴェニア語 (スロヴェニア)
es-AR	スペイン語 (アルゼンチン)
es-BO	スペイン語 (ボリビア)
es-CL	スペイン語 (チリ)
es-CO	スペイン語 (コロンビア)
es-CR	スペイン語 (コスタリカ)
es-DO	スペイン語 (ドミニカ共和国)
es-EC	スペイン語 (エクアドル)
es-SV	スペイン語 (エルサルバドル)
es-GT	スペイン語 (グアテマラ)
es-HN	スペイン語 (ホンデュラス)
es-MX	スペイン語 (メキシコ)
es-NI	スペイン語 (ニカラグア)
es-PA	スペイン語 (パナマ)
es-PY	スペイン語 (パラグアイ)
es-PE	スペイン語 (ペルー)
es-PR	スペイン語 (プエルトリコ)
es-ES	スペイン語 (スペイン)
es-ES_tradnl	スペイン語 (スペイン、伝統スペイン語)
es-UY	スペイン語 (ウルグアイ)
es-VE	スペイン語 (ベネズエラ)
sw-KE	スワヒリ語 (ケニア)
sv-FI	スウェーデン語 (フィンランド)
sv-SE	スウェーデン語 (スウェーデン)
syr-SY	シリア語 (シリア)

カルチャと言語コード	カルチャ
ta-IN	タミル語（インド）
tt-RU	タタール語（ロシア）
te-IN	テルグ語（インド）
th-TH	タイ語（タイ）
tr-TR	トルコ語（トルコ）
uk-UA	ウクライナ語（ウクライナ）
ur-PK	ウルドゥー語（パキスタン）
uz-Cyrl-UZ	ウズベク語（ウズベキスタン、キリルアルファベット）
uz-Latn-UZ	ウズベク語（ウズベキスタン、ラテン語）
vi-VN	ヴェトナム語（ヴェトナム）

バージョン

スタディコンポーネントのバージョン

スタディコンポーネントを作成するとき、定期的にアップデートする必要があります。実装中、フォームの開発が進みスポンサーがレビューを行うのに伴い、スタディコンポーネント定義を修正する必要がある可能性があります。スタディ実施中、スタディプロトコルの変更を反映するため、もしくは質問への回答方法をより明確化して提供するために、スタディコンポーネントを追加するか、または変更する必要がある場合があります。

スタディコンポーネントの新バージョンを作成するには、Central Designer の Incremental Deployment パッケージをインストールするか、新しいコンポーネント定義を含むファイルを MedML インストーラユーティリティにサブミットします。

注： **Central Designer Incremental Deployment** パッケージをインストールして、スタディオブジェクト定義のバージョン管理を行うことをお勧めします。

MedML インストーラユーティリティは、ほとんどのコンポーネントタイプ用の改訂コンポーネント定義をスタディデータベースに作成します。また、ブラウザでフォームを表示するときに、InForm アプリケーションは、自動的に最新のコンポーネント定義を使用します。ただし、以下の例外があります：

- **PFElement** — PFElement 定義を変更するには、SQL ステートメントを使用することによって PF_ELEMENT テーブルから定義を削除する必要があります。または、異なる名前を持つ新しい PFElement を作成して、オリジナルの PFElement を使用したすべての場所で新しい PFElement REFNAME を使用する必要があります。

詳細については、「**PFElement**」（168ページ）を参照してください。

- **StudyVersion** — 改訂版はスタディバージョンのアップデートの機能が提供され、

StudyVersion 定義で **VERSIONDESCRIPTION** 属性を使用することによって、手動でバージョン番号をコントロールする必要があります。

詳細については、「**StudyVersion**」 (221ページ) を参照してください。

- **StudyVersionSite** — 改訂版は StudyVersion 定義をそれが使用されている施設にリンクします。StudyVersionSite 定義で **VERSIONDESCRIPTION** 属性を使用することによって、手動でバージョン番号をコントロールする必要があります。

詳細については、「**StudyVersionSite**」 (218ページ) を参照してください。

- **StudyVersionDoc** — 改訂版は StudyVersion 定義を StudyVersion に伴うドキュメントにリンクします。StudyVersionDoc 定義で **VERSIONDESCRIPTION** 属性を使用することによって、手動でバージョン番号をコントロールする必要があります。

詳細については、「**StudyVersionDoc**」 (218ページ) を参照してください。

注： スタディコンポーネント定義を改訂する場合は、**DataType** プロパティで指定したデータタイプを変更することはできません。

CRF のバージョン

スタディ開始後に CRF を変更した場合、施設はできるだけ多くのスタディ症例で新しいバージョンの CRF を使用します。たとえば、採血を指定した時間に行うなど、時間に深く関わるアイテムがフォームに追加される場合、データはできるだけ多くの症例から収集されます。このデータを収集するために、施設では新しいバージョンのフォームが必要になります。これはフォームの新バージョンを作成するとき、複数のフォームバージョンが同じスタディバージョン下で必要かどうかを考慮する必要があることを示しています。

- CRF の変更により、施設においてフォームへ入力のが完了している症例からデータを集める**必要がない**場合:
 - 必要に応じて、追加コントロールや変更済みコントロール、アイテム、またはセクションなどのフォームコンポーネントを改訂してください。
- CRF の変更により、施設においてフォームへ入力のが完了している症例からデータを集める**必要がある**場合:
 - 1 必要に応じて、追加コントロールや変更済みコントロール、アイテム、またはセクションなどのフォームコンポーネントを改訂してください。
 - 2 フォーム定義には、**ALREFNAME** 属性を指定し、既存フォームの代替定義を作成していることを示してください。

InForm アプリケーションを実行するとき、フォームのオリジナルバージョンと一緒に代替バージョンを提示します。

改訂された CRF を実装するには、フォームを含む新しいスタディバージョンの作成、または変更が適用される各施設の StudyVersionSite コンポーネントのアップデートを行ってください。

詳細については、「Form」および「**StudyVersion**」 (221ページ)、「**StudyVersionSite**」 (218ページ) を参照してください。

スタディプロトコルのバージョン

InForm アプリケーションは、スタディプロトコルの改訂版をサポートし、プロトコル変更の時点と施設の治験審査委員会（IRB; Institutional Review Board）が新バージョンの使用を承認する時点のタイミングの差異を考慮します。プロトコル変更が承認されると、InForm アプリケーションは、プロトコル変更に関連するものがある場合には、以下のタイミングでその新しいフォームの使用を開始します:

- 1 古いスタディバージョンの定義をコピーまたは改訂し、その定義の **VERSIONDESCRIPTION** 属性をアップデートすることによって、新しいまたは変更済みのフォームを含んでいる新しいスタディバージョンの作成を行ったとき。
詳細については、「**StudyVersion**」（221ページ）を参照してください。
- 2 変更を適用する StudyVersionSite コンポーネントそれぞれの **VERSIONDESCRIPTION** 属性をアップデートすることによって、施設で新しい StudyVersion 定義を設定したとき。
プロトコル修正を伴わないフォームの変更に 대해서는、各施設の StudyVersionSite コンポーネントをアップデートしてください。
プロトコル修正を伴う変更に関しては、施設の StudyVersionSite コンポーネントのアップデートは、IRB 承認または他の許可（例えば、安全性向上のための変更により、承認待ちのままで迅速に実装することが必要な場合のスポンサーからの連絡）を受け取ったときに限り行ってください。
詳細については、「**StudyVersionSite**」（218ページ）を参照してください。

MedML インストラユーティリティを使用して、新しいまたは更新した StudyVersion および StudyVersionSite コンポーネントをデータベースにロードしたとき。

ドキュメントのバージョン

InForm アプリケーションは、以下を含むすべてのタイプのドキュメントの改訂版をサポートします:

- スタディプロトコル
- CRF ヘルプ
- システムヘルプ
- サンプル CRB
- ビジットカリキュレーター
- スポンサーによって提供されたドキュメント

新しいドキュメントバージョンを実装するためには:

- 1 ドキュメントを作るファイルをアップデートしてください。
- 2 リンクに変更が生じる場合、または、ドキュメントにファイルを追加または削除を行う場合、ドキュメント定義がされている XML ファイルをアップデートします。
- 3 MedML インストラユーティリティを使用してドキュメント定義の XML ファイルの処理を行い、データベースに新しいまたは更新後のドキュメントファイルを

ロードします。

- 4 CRF ヘルプに変更を行った場合、MedML インストーラユーティリティを使用して、変更されたヘルプテキストが適用される各フォームの定義をデータベースに再びロードします。Central Designer アプリケーションを使用して CRF ヘルプを更新した場合は、変更を反映する Incremental Deployment パッケージを実行します。
- 5 各ドキュメントが関連付けられているスタディバージョンの **VERSIONDESCRIPTION** 属性をアップデートします。
- 6 StudyVersion の **VERSIONDEFINITION** 属性にマッチするように、StudyVersionDoc 定義の **VERSIONDESCRIPTION** 属性をアップデートします。
- 7 各 StudyVersionSite コンポーネントの **VERSIONDESCRIPTION** 属性をアップデートすることによって、新しいドキュメントバージョンを使用する各施設で、新しい StudyVersion 定義を設定してください。

MedML インストーラユーティリティを使用して、更新後の定義を次の順でデータベースにロードしてください:StudyVersion、StudyVersionDoc、StudyVersionSite。

MedML インストーラユーティリティ

MedML インストーラユーティリティの概要

MedML インストーラユーティリティ:

- MedML エlementで生成する XML ファイルの解析
- 以下との照合によるファイルの検証
 - MedML スキーマファイル (MedMLSchema.xsd)
 - データベーススキーマによって強制される制約条件
- ファイルに定義されているスタディコンポーネントの、データベースへのロード

MedML インストーラユーティリティは、以下から実行できます:

- コマンドラインから

詳細については、「**コマンドラインからの MedML インストーラユーティリティの実行**」(51ページ)を参照してください。

- [スタート] メニューから

MedML インストーラユーティリティによるスタディのセットアップ

すべてのスタディコンポーネントを定義したら、MedML インストーラユーティリティを使用して、スタディ内のコンポーネントのスタディ固有の定義をロードします。これらの定義は1セットのXMLファイルの中にあります。

スタディデータベースでスタディを定義する情報を指定するために:

- 1 MedML スキーマまたは Central Designer アプリケーションを使用して、スタディのすべてのコンポーネントを定義した XML ファイルセットを作成します。

注: Central Designer アプリケーションを使用してスタディを設計することをお勧めします。

- 2 その XML ファイルを、MedML インストーラユーティリティを用いて処理します。

妥当性チェック

選択値チェック

MedML インストーラユーティリティは、ラジオボタンまたはチェックボックスのグループコントロールで選択した選択値をチェックします。この確認は選択値の構文と意味を把握し、スタディに反映される前の段階で問題を特定できるようにします。検出されたエラーは MedML インストーラ画面の出力セクションに表示されます。また、出力ファイルを作成することもできます。

MedML インストーラユーティリティ:

- 選択値の属性がすべての子シンプルコントロールに指定されているかどうかチェックします。選択値の属性が不完全な場合、厳格モードと非厳格モードの両方でエラーが発生します。

注： 非厳格モードでは、スタディ設計の目的にのみ、**MedML** 仕様に完全には準拠していない定義をロードできます。スタディの終了後に、ロードする定義が **MedML** 仕様に完全に準拠するように、厳格モードに切り替えます。

- 選択値の属性が、グループコントロールのすべての子コントロールに指定されているかどうかを確認します。選択値の属性が不完全な場合、厳格モードと非厳格モードの両方でエラーが発生します。
- 以下の基準に基づいて、データタイプがすべての子コントロールにわたって一貫しているかどうかを確認します。
 - 子シンプルコントロールがない場合、選択値はすべて文字列になると考えられるため、データタイプの一貫性についての確認は実行されません。
 - 1 つまたは複数の子シンプルコントロールがある場合、シンプルコントロールはすべて同一のデータタイプである必要があります（文字列、整数、または浮動小数など）。また、非シンプルコントロールについても、ユーザ定義された選択値は同一のデータタイプである必要があります。

厳格モードの場合、子コントロール全体のデータタイプが一貫していないと、エラーが発生します。非厳格モードの場合は、警告メッセージが表示されます。

- 選択値が重複していないかどうかを確認します。重複値が存在する場合、厳格モードと非厳格モードの両方でエラーが発生します。
- 非シンプル子コントロールに空の選択値がないかどうかを確認します。空の選択値があり、詳細モードを使用している場合は、厳格モードと非厳格モードの両方で警告メッセージが表示されます。

コードマッピングのチェック

Central Coding アプリケーションを使用して InForm スタディの症例データをコード化する際、スタディ設計者はマッピング定義を作成して、以下を指定する必要があります。

- コード化するアイテム
- コード化されたデータを保持するアイテム
- 上記 2 種類のアイテム間の関係、特定のコードターゲット、コーディング辞書のコンテキストアイテム

コードマッピングの定義をインストールする際、MedML インストーラユーティリティは以下のチェックを実行します。

- Verbatim パスが存在し、完全であるかどうか
- 指定された辞書が有効であるかどうか
- 指定されたコードターゲットおよびコンテキストアイテムが有効かどうか
- コードターゲットおよびコンテキストアイテムに関する確認は、次の通りです。
 - マッピング内で、各名前がユニークであるかどうか
 - パスが存在し、完全であるかどうか
- コードターゲットのパスが、トップレベルフィールドのコントロールまたは計算コントロールを指しているかどうか
- verbatim パスとそのコンテキストアイテムのパスの反復されている部分、または verbatim パスとそのコードターゲットのパスが、それぞれ一致しているかどうか
verbatim およびコードターゲットのコントロール、または verbatim およびコンテキストアイテムのコントロールは、フォームが反復されている場合同一である必要があり、コード化されたデータがアイテムセットに表示されている場合、同一のアイテムセットにある必要があります。
- 指定された verbatim タイプが有効かどうか

MedML インストーラユーティリティの起動

- [スタート]>[すべてのプログラム]>[Oracle]>[InForm 6.0]>[InForm MedML Installer] を選択します。

または

- コマンドラインからユーティリティを実行します。

詳細については、「**コマンドラインからの MedML インストーラの実行**」(51ページ)を参照してください。

MedML インストーラ画面について

定義は下から順に処理されます。コンポーネント定義と XML ファイルが、正しい順番で編成されているかどうかを確認する必要があります。コントロールなどの基本的なコンポーネントは、最初に定義、インポート、および読み込まれる必要があります。

その後に、それらを参照するアイテムが続きます。

フィールド	説明
XML ファイル	追加（または修正）してインストールするファイルの名前。
処理される XML ファイル	新規インストールに含まれるすべてのファイルのリスト。
トライアル名	作業中のトライアルの名前。
追加パス	応答 (RSP) ファイルを使用して XML ファイルのグループを処理する場合、RSP ファイルで参照されている XML ファイルのロケーションであるルートディレクトリの名前をタイプします。
詳細	ビルド中の発生事項に関する詳細レポートを作成します。
解析のみ	XML エラーおよび InForm データベースとの互換性について、データをロードすることなく、入力ファイルをチェックします。
厳格モード	<p>完全なコンポーネント定義のみを許容します。</p> <p>デフォルトではチェックボックスは選択されておらず、非厳格モードになっています。非厳格モードでは、MedML インストーラユーティリティは、すべての依存コンポーネントが揃っていない不完全なコンポーネント定義を許容します。例えば、エレメント定義が事前にデータベースに一部しかロードされていないラジオコントロール定義でも許容されます。</p> <p>注： このオプションは、-notstrict オプションでコマンドラインを使用してユーティリティを起動した場合にのみ利用できます。非厳格モードは、スタディ開発環境のみでの使用を前提としています。稼動環境では、使用しないでください。</p> <p>同様に、一度接続が定義されると、完全な MedML プロセスでのみスタディ定義データをロードすることができます。</p>
オンライン	<p>スタディを設計モードで開始します。デフォルトでは、チェックボックスが選択されています。オンラインモードでは、MedML インストーラユーティリティを起動するときにスタディが開始されていない場合、ユーティリティはスタディを開始し、インストールが完了するまでシャットダウンしません。このモードでは、ユーザはスタディ定義データをインストールした結果をすぐに見ることが可能です。</p> <p>このチェックボックスをオフにした場合、MedML インストーラユーティリティはスタディをオフラインモードで開始します。オフラインモードでは、MedML インストーラユーティリティは、インストールが完了した時点でスタディを停止します。スタディを再開するまでは、ユーザはスタディへの接続はできません。</p> <p>注： このオプションは、-online オプションでコマンドラインを使用してユーティリティを起動した場合にのみ利用できます。</p>

フィールド	説明
出力ファイル	処理するメッセージを含むファイルの場所。

MedML インストーラユーティリティの使用方法

初めて MedML インストーラユーティリティを起動する場合

スタディの XML ファイルについてテストを開始するには、MedML インストーラユーティリティを使用してファイルをスタディデータベースに組み込みます。

MedML インストーラユーティリティを実行するには、次の手順を実行します。

- 1 MedML インストーラユーティリティを起動します。詳細については、「**MedML インストーラユーティリティの起動**」（47ページ）を参照してください。
[MedML インストーラ] ウィンドウが表示されます。
- 2 **[XML ファイル]** フィールドで、処理する XML ファイルを選択するか、ファイルの場所を参照して特定します。

注： 応答 (**RSP**) ファイルに処理するファイルをすべてリストし、**[XML ファイル]** フィールドに応答ファイルの名前を入力するという方法もあります。

- 3 **[追加]** をクリックします。
- 4 すべての XML ファイルに、1～3 のステップを繰り返します。
- 5 **[トライアル名]** ドロップダウンリストからトライアルの名前を選択または入力します。
- 6 または、MedML インストーラ画面の追加のチェックボックスを選択して、MedML インストーラユーティリティにファイルの処理方法を指定します。

詳細については、「**MedML インストーラ画面について**」（47ページ）を参照してください。

- 7 **[実行]** をクリックします。

MedML インストーラユーティリティは XML ファイルを処理し、情報をスタディデータベースにロードします。XML ファイルがすべて処理されると、「**Completed Successfully**」というメッセージが MedML インストーラ出力画面に表示されます。

エラーメッセージの詳細に関しては、「**MedML インストーラ出力メッセージについて**」（52ページ）を参照してください。

進行中のスタディの更新

すでに進行中のスタディのファイルに変更を加えるときは、新しいバージョンのファイルを処理し、新しいスタディバージョンを処理して、変更したファイルを参照しているすべてのオブジェクトの新しいバージョンを取得する必要があります。たとえばセクションに変更を加えた場合、そのセクションのファイル、およびそのセクションへの

参照が含まれているすべてのファイルを再度処理する必要があります。

MedML インストーラユーティリティを使用してすでに進行中のスタディを更新する場合は、変更が反映されるように、PFAdmin ユーティリティでスタディ定義を停止し、再起動する必要があります。

進行中のスタディを更新するには、次の手順に従います。

- 1 **Microsoft Windows** コンポーネントを介して **Internet Information** サーバのスタディの仮想ディレクトリにある [ディレクトリセキュリティ] プロパティを変更することによって、スタディへのアクセスを制限します。

詳細については、『スタディとレポート設定ガイド』を参照してください。

- 2 MedML インストーラユーティリティを起動します。

[MedML インストーラ] ウィンドウが表示されます。

- 3 **[XML ファイル]** フィールドで、変更したファイルを選択します。

- 4 **[追加]** をクリックします。

処理対象の XML ファイルのリストに、その XML ファイルが表示されます。

- 5 **[トライアル名]** ドロップダウンリストからトライアルの名前を選択または入力します。

- 6 または、MedML インストーラ画面の追加のチェックボックスを選択して、MedML インストーラユーティリティにファイルの処理方法を指定します。

詳細については、「**MedML インストーラ画面について**」（47ページ）を参照してください。

- 7 **[実行]** をクリックします。

MedML インストーラユーティリティは XML ファイルを処理し、情報をスタディデータベースにロードします。XML ファイルがすべて処理されると、「**Completed Successfully**」というメッセージが MedML インストーラ出力画面に表示されます。

エラーメッセージの詳細に関しては、「**MedML インストーラ出力メッセージについて**」（52ページ）を参照してください。

- 8 スタディを停止し、再起動します。

ビルドからのXMLファイルの除去

規制当局は、スタディからデータを削除することに対して、厳格な規制を設けています。スタディの一部となった XML は、削除できません。しかし、そのかわりに XML ファイルの他の不必要なファイルへの参照を削除し、その後 MedML インストーラユーティリティを使用して、すべてのファイルを再度処理することができます。削除したファイルのオリジナルのファイル定義はデータベースに残りますが、スタディでは表示されません。

コマンドラインからの MedML インストーラユーティリティの実行

注： MedML インストーラユーティリティを **PFConsole** ユーティリティから実行することをお勧めします。詳細については、「**コマンドラインからの PFConsole ユーティリティの実行** (15ページ) を参照してください。

MedML インストーラユーティリティをコマンドラインから実行するには、次の構文を使用します。

```
PFConsole PFMMinst -trial trialname [-verbose] [-?][-help] [-autorun] [-autorun]
[-outfile output file name] [-parse] -xml [@] XMLFile1 [@] XMLFile2 ...
[@] XMLFileN
```

注： すべてのコマンドラインアプリケーション (MedML インストーラユーティリティ、InForm データインポートユーティリティ、および InForm データエクスポートユーティリティ) は、デフォルトプロダクトロケールを使用します。デフォルトプロダクトロケールは InForm のインストール中、または **PFAdmin** コマンドで設定されます。

コマンドラインパラメータ

パラメータ	変数	説明
PFConsole ユーティリティ		PFConsole ユーティリティを起動します。
PFMMinst		MedML インストーラユーティリティを起動します。
-trial	trialname	ビルド中のスタディの名前です。
-verbose		ビルドの詳細を生成し、ビルド中にそれらを表示するかどうか指定します。
-?		MedML インストーラユーティリティを実行するためのコマンドラインの構文を表示します。
-help		MedML インストーラユーティリティのオンラインヘルプを開きます。
-autorun		コマンドウィンドウで MedML インストーラユーティリティを実行します。PFConsole ユーティリティを使用してコマンドラインから MedML インストーラユーティリティを使用することをお勧めします。詳細については、「 PFConsole ユーティリティ 」 (13ページ) を参照してください。

パラメータ	変数	説明
-notstrict		<p>厳格モードを指定するためのチェックボックスを備えた、MedML インストーラボックスを開きます。デフォルトではチェックボックスはオフで、非厳格モードになっています。非厳格モードでは、MedML インストーラユーティリティは、依存コンポーネントの一部のみしか揃っていない不完全なコンポーネント定義を許容します。例えば、エレメント定義が事前にデータベースに一部しかロードされていないラジオコントロール定義は、許容されます。</p> <p>注： 非厳格モードは、スタディ開発環境でのみ使用してください。非厳格モードでは、稼動スタディ定義をロードしないでください。</p>
-outfile	output filename	出力ファイルにビルドの詳細を保存するかどうかを指定し、保存するパスとファイル名を示します。
-parse		ユーティリティに、データベースにビルドを反映しないように指示します。このモードは、エラーをテストする際に有用です。
-xml	XMLFile	ビルドに含める XML ファイルの名前を指定します。
@<response>		処理される XML ファイルのリストを含む応答ファイル名を指定します。

MedML インストーラ出力メッセージ

MedML インストーラユーティリティは、インストールプロセス中に、次の 3 種類の出力メッセージを生成します。

MedML インストーラ出力メッセージ		
Message	説明	例
情報	ファイルのインストールで深刻な問題を引き起こす可能性は低いものの、知っておくべき状況を示します。	情報: Inserting PFELEMENT 'mestrUnk' (PFELEMENT 'mestrUnk' を挿入しています)

MedML インストーラ出力メッセージ

警告	問題を起こす危険があり、修正の必要がある事象を示していますが、この事象が発生してもファイルのインストールは停止されません。	警告： File “D:\PF408\FLTELT.xml” does not contain any known tags (ファイル “D:\PF408\FLTELT.xml” には既知のタグが含まれていません)
失敗 (エラー)	この事象が起きた場合、ファイルのインストールを停止します。	エラー: 子セクション ‘sctIEInclusion’ を取得できませんでした。 エラー:Item not found (アイテムが見つかりません)

MedML インストーラユーティリティが正常に完了しない場合、以下の方法によってエラーが発生した場所を判断します。

- 出力メッセージをチェックする
- InForm アプリケーションを開始する前に、MedML インストーラユーティリティを起動していたかどうかを確認する
- ビルドが正しいディレクトリにあるかどうかを確認する
- すべてのパラメータが正確に入力されているかどうか確認する
- 入力ファイルへのパスが正確かどうかを確認する
- コンポーネントが正しい順序で処理され、必要なすべてのコンポーネントが XML ファイルまたはデータベースに存在しているかどうかを確認する

注： MedML インストーラユーティリティでは、出力ウィンドウとログファイルにエラーと警告の両方が表示されます。XML ファイルをインストールする前にエラーを修正する必要があります。

エクスターナルデータマッピング

エクスターナルデータ形式とのマッピングの定義

スタディを定義することと関連して、InForm スタディデータベースのスタディコンポーネントからエクスターナルデータフォーマットのマッピングを作成することが必要になる場合があります。マッピング定義の作成後、InForm データエクスポートユーティリティを使用して、選択されたフォーマットでスタディデータをエクスポートすることができます。MedML スキーマエレメントでは以下に対するマッピングをサポートします。

- Clintrial データベース
- 顧客定義済みデータベース(CDD; Customer-Defined Database)

マッピング定義をスタディデータベースにロードするためには、次の手順に従います。

- 1 マッピング定義を XML ファイルとして生成します。
- 2 その XML ファイルを、MedML インストラユーティリティを用いて処理します。

エクスターナルマッピングターゲット

マッピングの作成が可能な外部エンティティは以下の通りです。

- **Central Coding** — Central Coding アプリケーションに提供され、InForm アプリケーションに返されるコードを有効にする情報のマッピングを指定します:
 - コード化ディクショナリ
 - コード化される InForm アプリケーションのコントロール (verbatim)
 - コードターゲットおよびコンテキストアイテムとして使用される InForm アプリケーションのコントロール
- **Clintrial** — Clintrial スタディ定義内で、パネル内のターゲットアイテムのマッピングを指定します。
- **顧客定義済みデータベース (CDD)** — 外部のオラクルデータベースのテーブルカラムのマッピングを指定します。

エクスターナルデータマッピングリファレンス

AssocCDD

目的

顧客定義済みデータベースに対する 2 つのフォームの関連についてマッピングを特定します。関連定義は **TYPE=RELATION** のフォームセットとして定義されます。

構文

```
<ASSOCCDD
```

```

REFNAME="name"
[DESIGNNOTE="text"]
[ACTIVE="true|false"]
TARGETTABLE="name"
ASSOCREFNAME="name"
[LABEL="name"]/>

```

属性

REFNAME="*name*"

関連をマップしたい CDD の RefName。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

ACTIVE="true|false"

コンポーネントが有効であるかどうかを示します。オプションは true または false です。デフォルトは、True になっています。オプション

TARGETTABLE="*name*"

関連コンポーネントがアップデートされる CDD テーブルの名前。関連テーブルは、関連を作る 2 つのフォームの RefName と各フォームが設定されているビジットを含む 4 つのカラムから成ります。必須

ASSOCREFNAME="*name*"

関連を定義するフォームセットの RefName。必須

LABEL="*name*"

関連のためのテキストラベル。ラベルは最大 255 文字です。オプション

例

以下の例では、COMMONCRF フォームセットで AE と CM フォームの間の関連に関するマッピングを説明しています。

```
<EXTERNALMAP>
```

```
  <PATH>
```

```
    <CHAPTERREF REFNAME="COMMONCRF"/>
```

```
  </PATH>
```

```
  <ASSOCCDD REFNAME="CDD_WITH_ASSOC"
```

```
    TARGETTABLE="t_assoc1"
```

```
    ASSOCREFNAME="AECM_ASSOC"/>
```

```
</EXTERNALMAP>
```

CDD

目的

CDD に 1 つのマッピングされたコントロールのターゲットを指定します。

構文

<CDD

REFNAME="*name*"

[DESIGNNOTE="*text*"]

[ACTIVE="*true* | *false*"]

TARGETTABLE="*name*"

TARGETCOLUMN="*name*"

TARGETCOLUMNTYPE="NUMERIC | FLOAT | DATE | SPLITDATE | STRING | TEXT"

[TARGETCOLUMNMAXLENGTH="*length*"]

[TARGETKEYTYPE="PATIENT | PATIENTVISIT | PATIENTTOFORM | PATIENTTOSECTION |

PATIENTTOITEMSET | PATIENTTOITEM | PATIENTTOCONTROL |

PIVOTPATIENT | PIVOTVISIT | PIVOTFORM | PIVOTSECTION"

[PIVOTCOLUMN="*true* | *false*"]

[LABEL="*string*"]/>

属性

REFNAME="*name*"

CDD の RefName。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

ACTIVE="*true* | *false*"

コンポーネントが有効であるかどうかを示します。オプションは true または false です。デフォルトは、True になっています。オプション

TARGETTABLE="*name*"

マッピングされたフォームアイテムの更新に対応する、顧客定義済みデータベースのデータテーブルの名前。ラベルは 30 文字以下です。必須

TARGETCOLUMN="*name*"

マップされたフォームアイテムへの追加あるいは更新に対応する、顧客定義済みデータベースのデータテーブルのカラム名。ラベルは 25 文字以下です。必須

TARGETCOLUMNTYPE="NUMERIC | FLOAT | DATE | SPLITDATE | STRING | TEXT"

ターゲットとなるカラムのデータのタイプ。必須

オプションは以下の通りです。

- **NUMERIC**
- **FLOAT**
- **DATE** — 日付/日時フィールドに入力されるデータ。日付と時間のフィールドをマッピングするとき、InForm ソフトウェアは 4 つのカラムを作成します。
 - **DATE** — 完全な日時。
 - **_DT サフィックス** — 日付コンポーネントのみの日時フィールド。
 - **_TM サフィックス** — 時間コンポーネントのみの日時フィールド。
 - **_STR サフィックス** — 不完全な日時フィールド。
- **SPLITDATE** — 日時コントロールのデータ。6 つのカラムにマップされ、それぞれがコントロールの日付または時間コンポーネントのいずれかを保持しています。各カラムは生成される時点でカラム名を持ち、次のような適切な添え字が付けられます。
_Day、_Mon、_Year、_Hour、_Min、_Sec。
- **STRING** — 255 文字未満。
- **TEXT** — long varchar データ。

注：チェックボックスコントロールなどの複数選択コントロール~データを受け取るカラムの **TARGETCOLUMNTYPE** 属性は、**STRING** に設定する必要があります。

TARGETCOLUMNMAXLENGTH="length"

TARGETCOLUMNTYPE=STRING のターゲットカラムの最大長。カラムは 255 文字以下です。オプション

TARGETKEYTYPE="PATIENT|PATIENTVISIT|PATIENTTOFORM|PATIENTTOSECTION|
PATIENTTOITEMSET|PATIENTTOITEM|PATIENTTOCONTROL|
PIVOTPATIENT|PIVOTVISIT|PIVOTFORM|PIVOTSECTION">

以下のキータイプを用い、ターゲットテーブルのプライマリキーカラムの構成を指定します。以前サブミットされたプライマリキーからプライマリキーのコンポーネントの変更がある度に、>ifsw< は新しい列をターゲットテーブルに挿入します。プライマリキーは以下の DBUID とインデックスから成ります。

- **PATIENT** — PatientID、ItemsetIndex
- **PATIENTVISIT**(デフォルト) — PatientID、VisitID、ItemsetIndex、および VisitIndex
- **PATIENTTOFORM** — PatientID、VisitID、ItemsetIndex、VisitIndex、および FormID
- **PATIENTTOSECTION** — PatientID、VisitID、ItemsetIndex、VisitIndex、FormID、および SectionID
- **PATIENTTOITEMSET** — PatientID、VisitID、ItemsetIndex、VisitIndex、FormID、SectionID、および ItemsetID
- **PATIENTTOITEM** — PatientID、VisitID、ItemsetIndex、VisitIndex、FormID、SectionID、ItemsetID、および ItemID
- **PATIENTTOCONTROL** — PatientID、VisitID、ItemsetIndex、VisitIndex、FormID、SectionID、ItemsetID、ItemID、および 5 つの ControlID。このキータイプを持つ

ターゲットテーブルは、データ選択に使用できるデータラベルも持っています。

以下のキータイプに対して、プライマリキーカラムは PatientID、VisitID、ItemsetIndex、VisitIndex、FormID、SectionID、ItemsetID、ItemID と 5 つの ControllID です。キータイプ選定は、ピボットセットの構成(グループカラム内の 1 つのカラムがピボットカラムと定義される)を決定します。ピボットセット中では、非ピボットカラムにマッピングされるデータエレメントは各行で繰り返されます。また、これらのキータイプを持ったターゲットテーブルは、データ選択に使用可能な **LABEL** 属性の値として指定されるデータラベルも含んでいます。ピボットセットキーは以下の DBUIDs とインデックスから構成されています。

- **PIVOTPATIENT** — PatientID および VisitIndex
- **PIVOTVISIT** — PatientID、VisitID、および VisitIndex
- **PIVOTFORM** — PatientID、VisitID、FormID、および VisitIndex
- **PIVOTSECTION** — PatientID、VisitID、FormID、SectionID、および VisitIndex

注： アイテムセットの中にあるコントロールのマッピングを含むテーブルは、ピボットキータイプのいずれも持つことはできません。

PIVOTCOLUMN="true|false"

TARGETCOLUMN 属性で指定されるカラムがピボットカラムであるかどうかを指定します。オプションは true または false です。**TARGETKEYTYPE** が PIVOTPATIENT、PIVOTVISIT、PIVOTFORM、あるいは PIVOTSECTION である場合、非ピボットカラムにマッピングされるデータエレメントは、ピボットセット中の各行で繰り返されます。

注： ピボットカラムはテーブルの中において最初のカラムである必要があります。

LABEL="*string*"

データアイテムのテキストラベル。PATIENTTOCONTROL か PIVOT キーのいずれかでターゲットテーブルの中のアイテムへのアクセスを可能にします。ラベルは 255 文字以下です。オプション

例

DESCRIBETEXT コントロールを CDD1 CDD のカラムにマッピングする、CDD エレメントの使用例です。

```
<EXTERNALMAP>
```

```
  <PATH>
```

```
    <CHAPTERREF REFNAME="PF_ALL_VISITS"/>
```

```
    <PAGEREF REFNAME="ECG"/>
```

```
    <SECTIONREF REFNAME="CHESTXRAY"/>
```

```
    <ITEMSETREF REFNAME="1"/>
```

```
    <ITEMREF REFNAME="INTERPRET2"/>
```

```
    <CONTROLREF REFNAME="INTERPRETRADIO2"/>
```

```
    <CONTROLREF REFNAME="DESCRIBETEXT"/>
```

```
  </PATH>
```

```
  <CDD REFNAME="CDD1" KEYTYPE="PATIENT" TARGETTABLE="t_ECG"  
    TARGETKEYTYPE="PATIENTVISIT" TARGETCOLUMN="COMMONDAT1"  
    TARGETCOLUMNTYPE="TEXT"/>
```

```
</EXTERNALMAP>
```

Chapterref

目的

マッピング定義の Path エlementで、マッピングされるコントロールの位置として、フォームセットの RefName を指定します。Path Elementによって定義される RefName パスに 1 つの Chapterref Elementを含めてください。

構文

```
<CHAPTERREF  
  REFNAME="name" />
```

属性

REFNAME="*name*"

マッピングされたソースコントロールが存在するフォームセットの RefName。指定されたフォーム/セクション/アイテムセット/アイテムの組み合わせがあるすべてのビジットから、マッピング定義で指定されたターゲットにデータをマップすることを指定するには、特別な RefName である「PF_ALL_VISITS」を RefName として入力してください。必須

例

パス定義における Chapterref Elementでの PF_ALL_VISITS RefName の使用例です。

```
<PATH>  
  <CHAPTERREF REFNAME="PF_ALL_VISITS" />  
  <PAGEREF REFNAME="ECG" />  
  <SECTIONREF REFNAME="LEADECG" />  
  <ITEMSETREF REFNAME="1" />  
  <ITEMREF REFNAME="DATEASSESS" />  
  <CONTROLREF REFNAME="COMMONDATE" />  
</PATH>
```

CodeTarget (マッピング)

目的

InForm スタディでコード化されたデータを持つコントロールから、データのコード化に使用するディクショナリの特定期型のコードターゲットへのマッピングを指定します。CodeTarget エlementは、Verbatim の CodingMap 構造のチャイルドElementであり、一連の Central Coding データマッピングの PATH Elementによって指定されます。

構文

```
<CODETARGET
  NAME="name"
  PATH="path"/>
```

属性

NAME="name"

PATH Elementとそのチャイルドコンポーネントによって指定され、Verbatim のデータをコード化するために使用されるディクショナリコードターゲットの名前。必須

PATH="path"

コード化したデータが保持されるコントロールの RefName パス。必須

制限

MedML インストラユーティリティおよび InForm アプリケーションは、コードターゲットのコントロールパスを定義する Path Elementに次の制限を強制します。

- コードターゲットのコントロールパスは、マッピング内でユニークである必要があります。
- コードターゲットのコントロールは、verbatim およびマッピング内の任意のコードターゲットのコントロールパスとは異なっている必要があります。
- コードターゲットのコントロールパスは、最上位のテキストボックスコントロールまたは計算コントロールを指す必要があります。
- verbatim、コードターゲット、およびコンテキストアイテムのコントロールは、以下のようである必要があります。
 - ビジットが反復されている場合、同一のビジット内に存在する。
 - フォームが反復されている場合、同一フォーム上に存在する。
 - コード化されたデータがアイテムセットに表示されている場合、同一アイテムセット内に存在する。
 - 相互に異なる。

例

次の例は、Central Coding アプリケーションでコード化するように PATH Elementによって指定されている ExternalMap Elementで、InForm スタディの 1 つの verbatim

のマッピングを識別します。

```
<EXTERNALMAP xmlns="MedML-TDE">
  <PATH>
    <CHAPTERREF REFNAME="vstCORE4"/>
    <PAGEREF REFNAME="frmChem"/>
    <SECTIONREF REFNAME="sctChem"/>
    <ITEMSETREF REFNAME="mitsLabInfo"/>
    <ITEMREF REFNAME="mitmAccNo"/>
    <CONTROLREF REFNAME="mcalLAB"/>
  </PATH>
  <CODINGMAP REFNAME="MAPPINGS3" VERBATIMTYPE="MEDPROD">
    <DICTIONARY TYPE="WHODD" VERSION="05Q4" CULTURE="en-US"/>
    <CODETARGET NAME="ATC 1.CODE"
PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabDate.mcalLAB"/>
    <CONTEXTINFORMATION>
      <CONTEXTITEM NAME="Indication"
PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabHi.mcalLAB"/>
      <CONTEXTITEM NAME="Route Of Administration"
PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabTest.mcalLAB"/>
    </CONTEXTINFORMATION>
  </CODINGMAP>
</EXTERNALMAP>
```

CodingMap

目的

コード化される 1 つの verbatim アイテムのマッピングを指定します。

構文

```
<CODINGMAP
  REFNAME="name"
  [VERBATIMTYPE="type"]>
  <DICTIONARY attributes/>
  <CODETARGET+ attributes/>
  <CONTEXTINFORMATION*/>
</CODINGMAP>
```

属性

REFNAME="name"

コード化する verbatim アイテムが含まれたコードマッピングコンポーネントの RefName。必須

VERBATIMTYPE="type"

コード化する verbatim のタイプ。オプション。この値は Central Coding アプリケーションのアイテムタイプに対応しています。

- **AE** — 有害事象
- **DISEASE** — 疾患
- **LABDATA** — ラボデータ
- **MEDPROD** — 医薬品

AE、DISEASE、および LABDATA は MedDRA デictionary の有効な verbatim タイプです。MEDPROD は WHO-DD Dictionary の有効な verbatim タイプです。

子

CodingMap エlement には次のチャイルドElement があります:

- 1 つの Dictionary Element。使用するコード化Dictionary を指定します。
- 少なくとも 1 つの CodeTarget Element。コード化されたデータを持つ InForm コントロールとデータのコード化に使用するDictionary コードターゲット間のマッピングを指定します。
- オプションの ContextInformation Element。1 つまたは複数の ContextItem Element が含まれます。ContextItem Element は、追加のコンテキスト情報を提供する InForm コントロールとデータのコード化に使用するDictionary コンテキストアイテムの間のマッピングを指定します。

例

```
<CODINGMAP REFNAME="MAPPINGS3" VERBATIMTYPE="MEDPROD"> <DICTIONARY
```

```

        TYPE="WHODD"          VERSION="05Q4"          CULTURE="en-US"/>    <CODETARGET
        NAME="ATC
1.CODE"          PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabDate.mcallAB"
/>    <CONTEXTINFORMATION>          <CONTEXTITEM          NAME="Indication"
        PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabHi.mcallAB"/>
<CONTEXTITEM          NAME="Route Of
Administration"          PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabT
est.mcallAB"/>    </CONTEXTINFORMATION></CODINGMAP>

```

ContextInformation

目的

コンテキストアイテムのマッピング定義を囲みます。InForm アプリケーションでコンテキストアイテムと識別されるコントロールは、コード化ディクショナリのコンテキストアイテムにデータを供給します。

構文

```

<CONTEXTINFORMATION>
  <CONTEXTITEM+ attributes/>
</CONTEXTINFORMATION>

```

子

ContextInformation エlement には、1 つまたは複数の ContextItem チャイルドElement があり、それぞれ InForm アプリケーションのコントロールとディクショナリコンテキストアイテムの間のマッピングを指定します。

例

```

<CODINGMAP  REFNAME="MAPPINGS3"  VERBATIMTYPE="MEDPROD">    <DICTIONARY
  TYPE="WHODD"          VERSION="05Q4"          CULTURE="en-US"/>    <CODETARGET
  NAME="ATC
1.CODE"          PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabDate.mcallAB"
/>    <CONTEXTINFORMATION>          <CONTEXTITEM          NAME="Indication"
        PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabHi.mcallAB"/>
<CONTEXTITEM          NAME="Route Of
Administration"          PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabT
est.mcallAB"/>    </CONTEXTINFORMATION></CODINGMAP>

```


ContextItem (マッピング)

目的

InForm スタディでコード化されたデータを持つコントロールから、データのコード化に使用するディクショナリの特定期型のコンテキストアイテムへのマッピングを指定します。ContextItem エlementは、Verbatim の ContextInformation 構造のチャイルドElementであり、一連の Central Coding データマッピングの PATH Elementによって指定されます。

```
<CONTEXTITEM
  NAME="name"
  PATH="path"/>
```

属性

NAME="name"

PATH ElementとそのチャイルドElementによって指定され、Verbatim のデータをコード化するために使用されるディクショナリコンテキストアイテムの名前。必須

PATH="path"

ディクショナリコンテキストアイテムのデータを持つコントロールの RefName パス。必須

制限

MedML インストラユーティリティおよび InForm アプリケーションは、コードターゲットのコントロールパスを定義する Path Elementに次の制限を強制します。

- コードターゲットのコントロールパスは、マッピング内でユニークである必要があります。
- コードターゲットのコントロールは、verbatim およびマッピング内の任意のコードターゲットのコントロールパスとは異なっている必要があります。
- コードターゲットのコントロールパスは、最上位のテキストボックスコントロールまたは計算コントロールを指す必要があります。
- verbatim、コードターゲット、およびコンテキストアイテムのコントロールは、以下のようである必要があります。
 - ビジットが反復されている場合、同一のビジット内に存在する。
 - フォームが反復されている場合、同一フォーム上に存在する。
 - コード化されたデータがアイテムセットに表示されている場合、同一アイテムセット内に存在する。
 - 相互に異なる。

例

```
<EXTERNALMAP xmlns="MedML-TDE">
  <PATH>
```

```

<CHAPTERREF REFNAME="vstCORE4"/>
<PAGEREF REFNAME="frmChem"/>
<SECTIONREF REFNAME="sctChem"/>
<ITEMSETREF REFNAME="mitsLabInfo"/>
<ITEMREF REFNAME="mitmAccNo"/>
<CONTROLREF REFNAME="mcalLAB"/>
</PATH>
<CODINGMAP REFNAME="MAPPINGS3" VERBATIMTYPE="MEDPROD">
  <DICTIONARY TYPE="WHODD" VERSION="05Q4" CULTURE="en-US"/>
  <CODETARGET NAME="ATCCODE"
    PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabDate.mcalLAB"/>
  <CONTEXTINFORMATION>
    <CONTEXTITEM NAME="Indication"
      PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabHi.mcalLAB"/>
    <CONTEXTITEM NAME="Route Of Administration"
      PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabTest.mcalLAB"/>
  </CONTEXTINFORMATION>
</CODINGMAP>
</EXTERNALMAP>

```

ContextPanel

目的

InForm データベースから CIS アプリケーションにエクスポートされるマッピングのセットの Clintrial パネルを定義します。パネル定義は、ContextPanel エlement (すべてのパネルに共通するアイテムを定義する CTItem エlement が関連付けされている)、および 1 つ以上の CTPanel エlement (それぞれに、パネルアイテムを定義する CTItem エlement が関連付けされている) で構成されます。

構文

```

<CONTEXTTPANEL
  REFNAME="name"
  [ACTIVE="true|false"]
  [DESIGNNOTE="text"]>
  <CTITEM* attributes/>
</CONTEXTTPANEL>

```

属性

REFNAME="name"

Clintrial マッピングのセットの RefName。必須

ACTIVE="true|false"

コンポーネントが有効であるかどうかを示します。オプションは true または false です。デフォルトは、True になっています。オプション

DESIGNNOTE="text"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力でき

ます。設計情報のみとして使用され、表示されません。オプション

子

ContextPanel エLEMENTの定義は、1 つ以上の CTItem エLEMENTを含めることができます。各 CTItem エLEMENTは、すべてのパネルに表示されるパネルアイテムを定義します。

例

MEDIKA_MAP と呼ばれるマッピングセットを含んだ ContextPanel 定義の例です。

ContextPanel 定義は 4 つの CTItem エLEMENTを含んでおり、これらはいずれもすべてのパネルに共通なパネルアイテムを定義します。

```
<EXTERNALMAP>
```

```
  <PATH />
```

```
  <CONTEXTPANEL REFNAME="MEDIKA_MAP">
```

```
    <CTITEM REFNAME="PATNUM" ITEMDATATYPE="TEXT" ISREPEAT="false"
    ISREQUIRED="true"
```

```
      DBFORMAT="VARCHAR2(20)" CONTEXTTYPE="1" />
```

```
    <CTITEM REFNAME="VISITID" ITEMDATATYPE="TEXT" ISREPEAT="false"
    ISREQUIRED="true"
```

```
      DBFORMAT="VARCHAR2(20)" CONTEXTTYPE="2" />
```

```
    <CTITEM REFNAME="FORMID" ITEMDATATYPE="TEXT" ISREPEAT="false"
    ISREQUIRED="true"
```

```
      DBFORMAT="VARCHAR2(20)" CONTEXTTYPE="3" />
```

```
    <CTITEM REFNAME="VISITINDEX" ITEMDATATYPE="FLOAT" ISREPEAT="true"
    ISREQUIRED="true"
```

```
      DBFORMAT="NUMBER(5,2)" CONTEXTTYPE="2" />
```

```
  </CONTEXTPANEL>
```

```
</EXTERNALMAP>
```

Controlref

目的

あるタイプのコンポーネントの定義を別のコンポーネントの定義に含めます。Controlref は単に Controlref が含まれているコンポーネントのチャイルドコントロールとして定義され、単独のコンポーネントとしてサブミットされません。

Controlref を含めることができるコンポーネントのタイプ:

- CheckBoxControl
- GroupControl
- Item
- RadioControl
- Path

Controlref を使用することによって含めることができるコンポーネントのタイプ:

- CalculatedControl
- CheckBoxControl
- DateTimeControl
- GroupControl
- PullDownControl
- RadioControl
- SimpleControl
- TextControl

構文

```
<CONTROLREF
  REFNAME="name"
  [ORDER="n"]
  [SELECTIONVALUE="text"]/>
```

属性

REFNAME="*name*"

Controlref によってコンパウンドコントロール (いくつかのコンポーネントを含んでいるコントロール) の定義に含められるコンポーネントの RefName。必須

ORDER="*n*"

コンパウンドコントロール定義における各 Controlref の順序。オプション。順序を指定しない場合、MedML インストラユーティリティは、Controlref が参照するコンポーネントを入力された順番で順序付けします。

SELECTIONVALUE="*text*"

Controlref によって含められるコンパウンドコントロールの値。オプション。値を指定しない場合、ユーザがコンパウンドコントロールの中の 1 つのコントロールを選択すると、文字列 `!pf!` と選択されたコントロールの **DBUID** パスで構成される文字列がデータベースに値として保存されます。たとえば、テキストボックスコントロールに関連付けられた、「その他」というラベルの付いたチェックボックスまたはラジオコントロールコンポーネントの値を入力する際に、この属性が便利です。

注：**Controlref** または **Unitref** 定義を用いてコンパウンドコントロールを定義するとき、すべての下位コントロールが同じ **TYPE** 属性で定義され、同じデータタイプを返すようにしてください。

コンパウンドコントロールを定義するとき、**InForm** アプリケーションは最大 **5** 層のネスティングをサポートしています。**5** 層での設計が可能ですが、パフォーマンスを向上させるためにネストレベルはできるだけ少なくしてください。

Path エlementに **Controlref** 定義を含めるときは、**REFNAME** 属性だけが有効です。

例

この例は、最初のボタンにドロップダウンリストがあり、2 番目のボタンにテキストボックスがあるラジオボタンのリストの作成方法を示しています。ラジオボタンのリストの定義は、**Controlref** としてそれぞれのリストアイテムの定義を含んでいます。このリストは薬物療法の取得の際、使用されます。

- 1 **PFElement** のセットとしてドロップダウンリストを定義し、**Elementref** を用いて、それらを **PullDownControl** 定義に入れます。
- 2

```
<PFELEMENT REFNAME="QD" LABEL="QD" TYPE="STRING" VALUE="QD"/>
<PFELEMENT REFNAME="BID" LABEL="BID" TYPE="STRING" VALUE="BID"/>
<PFELEMENT REFNAME="TID" LABEL="TID" TYPE="STRING" VALUE="TID"/>
<PFELEMENT REFNAME="QID" LABEL="QID" TYPE="STRING" VALUE="QID"/>
<PULLDOWNCONTROL REFNAME="MEDREGIMEN"
  NAME="MEDREGIMEN">
  <ELEMENTREF REFNAME="QD" ORDER="1"/>
  <ELEMENTREF REFNAME="BID" ORDER="2"/>
  <ELEMENTREF REFNAME="TID" ORDER="3"/>
  <ELEMENTREF REFNAME="QID" ORDER="4"/>
</PULLDOWNCONTROL>
```
- 3 **TextControl** としてテキストボックスとそのキャプションを定義します。
- 4

```
<TEXTCONTROL REFNAME="MEDREGTEXT"
  NAME="MEDREGTEXT"
  HEIGHT="1"
  LENGTH="20"
  MAXLENGTH="20"
  DATATYPE="STRING"
  CAPTION="その他(指定する):"
  CAPTIONALIGN="TOP"/>
```
- 5 **Controlref** を使用して、プルダウンコントロールとテキストコントロールを **RadioControl** に含めます。[その他 (指定する)] ラジオボタンに「OTHER」の値を割り当てるために **SELECTIONVALUE** 属性を使用しています。

```

6  <RADIOCONTROL REFNAME="MEDREGGROUP"
    NAME="MEDREGRADIO" LAYOUT="VERTICAL">
    <CONTROLREF REFNAME="MEDREGIMEN" ORDER="1"/>
    <CONTROLREF REFNAME="MEDREGTEXT" ORDER="2"
        SELECTIONVALUE="OTHER"/>
    </RADIOCONTROL>

```

これは、InForm データベースと CDD 間のコンポーネントのマッピング作成において、ソースコントロールを識別するために Controlref 定義が Path エlementの中で使用されている例です。

```
<EXTERNALMAP>
```

```

<PATH>
  <CHAPTERREF REFNAME="PF_ALL_VISITS"/>
  <PAGEREF REFNAME="ECG"/>
  <SECTIONREF REFNAME="CHESTXRAY"/>
  <ITEMSETREF REFNAME="1"/>
  <ITEMREF REFNAME="INTERPRET2"/>
  <CONTROLREF REFNAME="INTERPRETRADIO2"/>
  <CONTROLREF REFNAME="DESCRIBETEXT"/>
</PATH>

```

```

<CDD REFNAME="CDD1" KEYTYPE="PATIENT" TARGETTABLE="t_ECG"
  TARGETKEYTYPE="PATIENTVISIT" TARGETCOLUMN="COMMONDAT1"
  TARGETCOLUMNNTYPE="TEXT"/>

```

```
</EXTERNALMAP>
```

CTItem

目的

Clintrial パネルアイテムへの InForm フォームアイテムのマッピングを定義します。CTItem の属性は、Clintrial ソフトウェアのアイテムの属性について説明します。

構文

```

<CTITEM
  REFNAME="name"
  [DESCRIPTION="text"]
  ITEMDATATYPE="TEXT|FIXED|FLOAT|DATE|DATETIME"
  [SUBSETVALUE="text"]
  [BLOCKKEYVALUE="text"]
  [PAGEKEYVALUE="text"]
  [DATEPART="n"]
  [ISDERIVED="true|false"]
  ISREQUIRED="true|false"
  DBFORMAT="format"
  [SASNAME="name"]
  CONTEXTTYPE="n"

```

```

ISREPEAT="true | false"
[CODELIST="text"]
[CHECKLIST="text"]
[RANGELB="n"]
[RANGEUB="n"]
[KEYORDER="n"]
[COPYWITHPANEL="true | false"]
[LOCKSTATUS="n"]/>

```

属性

REFNAME="*name*"

アイテムの RefName。必須

DESCRIPTION="*text*"

アイテムの説明。オプション

ITEMDATATYPE="TEXT|FIXED|FLOAT|DATE|DATETIME"

アイテムの値のデータタイプ。オプションは以下の通りです。

- テキスト
- Fixed
- Float
- 日
- DateTime

必須

SUBSETVALUE="*text*"

パネルに基づいたサブセットページセクションのサブセットキーの場合に、アイテムがとる値。オプション

BLOCKKEYVALUE="*text*"

Clintrial のブロックキーの値。この値を指定すると、この値がビジットの RefName をブロックキーよりも優先されます。オプション

PAGEKEYVALUE="*text*"

Clintrial のページキーの値。この値を指定すると、この値がフォームの RefName をページキーとして優先されます。オプション

DATEPART="*n*"

Clintrial アイテムにマップする日時コントロールの一部。日時コントロールの一部を個別の Clintrial アイテムにマップする場合にこの属性を使用します。オプションは以下の通りです。

- 0 — 未定義
- 1 — 年

- 2 — 月
- 3 — 日
- 4 — 時間
- 5 — 分
- 6 — 秒

オプション

ISDERIVED="*true|false*"

アイテムの値がパネルに関連付けられている計算文から計算されるかどうかを示します。オプションは true または false です。デフォルトは、False になっています。オプション

ISREQUIRED="*true|false*"

アイテムが必須であるかどうかを示します。オプションは true または false です。必須

DBFORMAT="*format*"

- Clintrial ソフトウェアがオラクルデータベースにアイテムの値を保存するフォーマット。オラクルデータベースフォーマット:VARCHAR2(*n*)
- DATE
- NUMBER(*xx*)
- NUMBER(*xx*,*yy*)
- NUMBER(*xx*,0).*x*

必須

SASNAME="*name*"

データが Clintrial SAS インターフェイスを通して SAS に送信される時のアイテム名。名前は 8 文字以内とし、SAS 命名規則に従わなければなりません。オプション

CONTEXTTYPE="*n*"

アイテムのコンテキストタイプ。コンテキストアイテムは、タイプ 1,2,3,4,5 のパネルで定義されている臨床データテーブルの各レコードに関連付けられます。また、プロトコルの ContextPanel 定義に含まれます。コンテキストタイプは次のとおりです。

- 0 — 非コンテキストアイテム
- 1 — 症例関連コンテキストアイテム
- 2 — ビジット関連コンテキストアイテム
- 3 — ページ関連コンテキストアイテム
- 4 — その他のコンテキストアイテム

必須

ISREPEAT="*true|false*"

ページセクションの中で複数の値を入力できるアイテムであることを示します。

CODELIST="*text*"

アイテムに関連付けられているコードリスト名。**CODELIST** と **CHECKLIST** は同時に使用できません。コードリストは入力した値をコード化します。アイテムの値として、コードリストのコードか値のみが入力されます。オプション

CHECKLIST="*text*"

アイテムに関連付けられているチェックリスト名。**CODELIST** と **CHECKLIST** は同時に使用できません。チェックリストはフィールドの入力候補を参照するために使用される一種のコードリストです。オプション

RANGELB="*n*"

アイテムの値として入力できる最小値。オプション

RANGEUD="*n*"

アイテムの値として入力できる最大値。オプション

KEYORDER="*n*"

アイテムがパネルキーの一部である場合、連続したキーアイテムの中で、アイテムが表示される順序。デフォルトは、0 (キーアイテムではない)になっています。オプション

COPYWITHPANEL="*true|false*"

パネルがコピーされた場合、パネルと共にアイテムをコピーするかどうかを指定します。オプションは true または false です。デフォルトは、True になっています。オプション

LOCKSTATUS="*n*"

アイテムが含まれているプロトコルがロックされるかどうかを示します。

- 0 — アイテムを修正できる
- 1 — アイテムは修正できないが、修正できるようにリセットすることが可能
- 2 — アイテムは修正できず、また修正できるように設定することもできない

デフォルトは 0 です。オプション

ISKEY="*true|false*"

アイテムがパネルのキーの一部であるかどうかを示します。デフォルトは、False になっています。オプション

例

CLIN1 パネルの 3 つのアイテムに関するマッピングの例です。

<EXTERNALMAP>

<PATH />

<CONTEXT PANEL REFNAME="CLIN1">

<CTITEM REFNAME="SUBJECT" ITEM DATATYPE="TEXT"

```
        DBFORMAT="DBF" ISREQUIRED="true" CONTEXTTYPE="1" />
    <CTITEM REFNAME="VISITITEM"ITEMDATATYPE="TEXT"
        DBFORMAT="DBF" ISREQUIRED="true" CONTEXTTYPE="2" />
    <CTITEM REFNAME="PAGEITEM" ITEMDATATYPE="TEXT"
        DBFORMAT="DBF" ISREQUIRED="true" CONTEXTTYPE="3" />
</CONTEXTPANEL>
</EXTERNALMAP>
```

CTItem

目的

Clintrial パネルアイテムへの InForm フォームアイテムのマッピングを定義します。
CTItem の属性は、Clintrial ソフトウェア のアイテムの属性について説明しています。

構文

```
<CTITEM
  REFNAME="name"
  [DESCRIPTION="text"]
  ITEMDATATYPE="TEXT|FIXED|FLOAT|DATE|DATETIME"
  [SUBSETVALUE="text"]
  [BLOCKKEYVALUE="text"]
  [PAGEKEYVALUE="text"]
  [DATEPART="n"]
  [ISDERIVED="true|false"]
  [ISREQUIRED="true|false"]
  DBFORMAT="format"
  [SASNAME="name"]
  CONTEXTTYPE="n"
  [ISREPEAT="true|false"]
  [CODELIST="text"]
  [CHECKLIST="text"]
  [RANGELB="n"]
  [RANGEUB="n"]
  [KEYORDER="n"]
  [COPYWITHPANEL="true|false"]
  [LOCKSTATUS="n"]/>
```

属性

REFNAME="*name*"

アイテムの RefName。必須

DESCRIPTION="*text*"

アイテムの説明。オプション

ITEMDATATYPE="TEXT|FIXED|FLOAT|DATE|DATETIME"

アイテムの値のデータタイプ。オプションは以下の通りです。

- Text
- Fixed
- Float
- Date
- DateTime

必須

SUBSETVALUE="*text*"

パネルに基づいたサブセットページセクションのサブセットキーである場合に、アイテムがとる値。オプション

BLOCKKEYVALUE="*text*"

Clintrial のブロックキーの値。この値を指定すると、この値がビジットの RefName よりもブロックキーとして優先されます。オプション

PAGEKEYVALUE="*text*"

Clintrial のページキーの値。この値を指定すると、この値がフォームの RefName よりもページキーとして優先されます。オプション

DATEPART="*n*"

Clintrial アイテムにマップする日時コントロールの一部。日時コントロールの部分を個別の Clintrial アイテムにマップする場合にこのプロパティを使用します。オプションは以下の通りです。

- 0 — 未定義
- 1 — 年
- 2 — 月
- 3 — 日
- 4 — 時
- 5 — 分
- 6 — 秒

オプション

ISDERIVED="*true|false*"

アイテムの値がパネルに関連付けられている計算文から計算されるかどうかを示します。オプションは true または false です。デフォルトは、False になっています。オプション

ISREQUIRED="*true|false*"

アイテムが必須であるかどうかを示します。オプションは true または false です。必須

DBFORMAT="*format*"

Clintrial ソフトウェア が Oracle データベースにアイテムの値を保存するフォーマット。Oracle データベースフォーマット:

- VARCHAR2(*n*)
- DATE
- NUMBER(*xx*)
- NUMBER(*xx*,*yy*)
- NUMBER(*xx*,0).*x*

必須

SASNAME="*name*"

データが Clintrial SAS インターフェースを通して SAS に送られるときのアイテム名。名前は 8 文字以内とし、SAS 命名規則に従わなければなりません。オプション

CONTEXTTYPE="*n*"

アイテムのコンテキストタイプ。コンテキストアイテムは、タイプ 1、2、3、4、5 のパネルで定義されている臨床データテーブルの各レコードに関連付けられます。また、プロトコルの ContextPanel 定義に含まれます。コンテキストタイプは次のとおりです。

- 0 — 非コンテキストアイテム
- 1 — 症例関連コンテキストアイテム
- 2 — ビジット関連コンテキストアイテム
- 3 — ページ関連コンテキストアイテム
- 4 — その他のコンテキストアイテム

必須

ISREPEAT="*true|false*"

ページセクションの中で複数の値を入力できるアイテムであることを示します。

CODELIST="*text*"

アイテムに関連付けられているコードリスト名。**CODELIST** と **CHECKLIST** は同時に使用できません。コードリストは入力した値をコード化します。アイテムの値として、コードリストのコードか値のみが入力されます。オプション

CHECKLIST="*text*"

アイテムに関連付けられているチェックリスト名。**CODELIST** と **CHECKLIST** は同時に使用できません。チェックリストはフィールドの入力候補を参照するために使用される一種のコードリストです。オプション

RANGELB="*n*"

アイテムの値として入力できる最小値。オプション

RANGEUD="*n*"

アイテムの値として入力できる最大値。オプション

KEYORDER="*n*"

アイテムがパネルキーの一部である場合、連続したキーアイテムの中で、アイテムが表示される順序。デフォルトは、0 (キーアイテムではない) になっています。オプション

COPYWITHPANEL="*true|false*"

パネルがコピーされた場合、パネルと共にアイテムをコピーするかどうかを指定します。オプションは true または false です。デフォルトは、True になっています。オプション

LOCKSTATUS="n"

アイテムが含まれているプロトコルがロックされるかどうかを示します。

- 0 — アイテムを修正できる
- 1 — アイテムは修正できないが、修正できるようにリセットすることが可能
- 2 — アイテムは修正できないし、修正できるようにすることもできない

デフォルトは 0 です。オプション

ISKEY="true|false"

アイテムがパネルのキーの一部であるかどうかを示します。デフォルトは、False になっています。オプション

例

CLIN1 パネルの 3 つのアイテムに関するマッピングの例です。

<EXTERNALMAP>

<PATH />

<CONTEXTPANEL REFNAME="CLIN1">

<CTITEM REFNAME="SUBJECT" ITEMDATATYPE="TEXT"

DBFORMAT="DBF" ISREQUIRED="true" CONTEXTTYPE="1" />

<CTITEM REFNAME="VISITITEM" ITEMDATATYPE="TEXT"

DBFORMAT="DBF" ISREQUIRED="true" CONTEXTTYPE="2" />

<CTITEM REFNAME="PAGEITEM" ITEMDATATYPE="TEXT"

DBFORMAT="DBF" ISREQUIRED="true" CONTEXTTYPE="3" />

</CONTEXTPANEL>

</EXTERNALMAP>

CTPanel

目的

InForm データベースから CIS アプリケーションにエクスポートされるマッピングのセットの Clintrial パネルを定義します。パネル定義は 1 つ以上の CTPanel エレメントから構成されており (CTPanel それぞれに、パネルアイテムを定義する CTItem エレメントが関連付けられている)、すべてのパネルに共通するアイテムは ContextPanel エレメントに定義されています。

構文

<CTPANEL

REFNAME="name"

PANELNAME="name"

[DESCRIPTION="text"]

PANELTYPE="n"

[SUBSETITEM="name"]

[ISPROTECTED="true|false"]

```

[ISVERIFIABLE="true | false"]
[ISDETAILPANEL="true | false"]
[DETAILCTITEM="name"]
[MASTERPANEL="name"]
[MASTERCITEM="name"]
[SASNAME="name"]
[LOCKSTATUS="n"]
[ACTIVE="true | false"]
[DESIGNNOTE="text"]>
  <CTITEM attributes/>
</CONTEXTPANEL>

```

属性

REFNAME="name"

Clintrial マッピングのセットの RefName。必須

PANELNAME="name"

Clintrial のパネルの名前。必須

DESCRIPTION="text"

Clintrial パネルの説明。オプション

PANELTYPE="n"

Clintrial パネルのタイプ:

- 0 — 非症例データ。データは特定の症例に関連しません。
- 1 — 各症例につき 1 レコード。各症例につき、スタディの間に 1 回だけ収集されます。
- 2 — 各症例につき複数のレコード。各症例につき、スタディの間に複数回収集されます。
- 3 — 症例の各ビジットにつき 1 つのレコード。症例の各ビジットにおいて 1 回だけ収集されます。
- 4 — 症例の各ビジットにつき複数のレコード。症例の各ビジットにおいて複数回収集されます。
- 5 — 登録された症例ごとに 1 レコード。

必須

SUBSETITEM="name"

パネルに基づいてサブセットページセクションにサブセットキーとして指定されるアイテム名。サブセットページセクションは Type 0、Type 2、または Type 4 のパネルで、スタディページで複数回現れます (サブセットキーアイテムの各々の値はデータ行を区別する役割があります)。オプション

ISPROTECTED="true | false"

パネルへのアクセス権が Clintrial で制限されるかどうかを示します。オプションは true

または `false` です。デフォルトは、`False` になっています。オプション

ISVERIFIABLE="*true|false*"

確認のため、パネルアイテム内でデータの二度入力が必要かどうかを示します。
デフォルトは、`False` になっています。オプション

ISDETAILPANEL="*true|false*"

マスタ-詳細関係の詳細ページセクションに **CTPanel** 定義が含まれているかどうかを示します。`true` または `false`。マスター-詳細関係とは、スタディページにおける2つのページセクション間の関係を指し、ここでは、一方のページセクション (マスターページセクション) にある各レコードが、他方のページセクション (詳細ページセクション) にある1つまたは複数の関連するレコードを持つことができます。データ入力の際、詳細ページセクションに表示されているレコードは、マスターページセクションで選択されているレコードに関連付けられています。必須

DETAILCTITEM="*name*"

CTPanel 定義が詳細ページセクションの一部である場合、詳細キーアイテムとして識別されるアイテム名。オプション

MASTERPANEL="*name*"

この **CTPanel** 定義がマスタ詳細関係に入れられる際の、マスタパネルの **PANELNAME**。この属性は、**ISDETAILPANEL** 属性の値が `TRUE` である場合にのみ有効です。
オプション

MASTERCTITEM="*name*"

DETAILCTITEM 属性で指定された詳細キーアイテムに対応する関連マスタパネルの上のアイテム名。この属性は、**ISDETAILPANEL** 属性の値が `TRUE` である場合にのみ有効です。オプション

SASNAME="*name*"

Clintrial SAS インターフェースを通して SAS にデータを送る時のパネル名。名前は8文字以内とし、SAS 命名規則に従わなければならない。オプション

LOCKSTATUS="*n*"

パネルが含まれているプロトコルがロックされているかどうかを示します。

- 0 — パネルを修正できる
- 1 — パネルは修正できないが、修正できるようにリセットすることが可能
- 2 — パネルは修正できないし、修正できるようにすることもできない

デフォルトは 0 です。オプション

ACTIVE="*true|false*"

コンポーネントが有効であるかどうかを示します。オプションは `true` または `false` です。
デフォルトは、`True` になっています。オプション

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

子

CTPanel エlementの定義には、1 つの CTItem Element (パネルでアイテムを定義します) が含まれます。

例

以下の例は、MEDIKA_MAP マッピングの **INCLUS** パネルの定義を INCLU1 アイテムの定義を用いて示しています。PATH Elementは、このパネルを InForm スタディ定義における RefName パスにマップします。

```
<EXTERNALMAP>
```

```
  <PATH>
```

```
    <CHAPTERREF REFNAME="PF_ALL_VISITS" />
```

```
    <PAGEREF REFNAME="1" />
```

```
    <SECTIONREF REFNAME="INCLUS" />
```

```
    <ITEMREF REFNAME="INCLU1" />
```

```
    <CONTROLREF REFNAME="INCLU1_c" />
```

```
  </PATH>
```

```
  <CTPANEL REFNAME="MEDIKA_MAP" PANELNAME="INCLUS" DESCRIPTION=""
```

```
    PANELTYPE="1" ISPROTECTED="false" ISDETAILPANEL="false">
```

```
    <CTITEM REFNAME="INCLU1" DESCRIPTION="Inclusion criteria"
```

```
      ITEMDATATYPE="FIXED" ISREQUIRED="false" ISREPEAT="false"
```

```
      DBFORMAT="NUMBER(1)" CONTEXTTYPE="0" ISDERIVED="false"
```

```
      CODELIST="M_YESNO" KEYORDER="0" COPYWITHPANEL="false"
```

```
    LOCKSTATUS="0" />
```

```
  </CTPANEL>
```

```
</EXTERNALMAP>
```

Dictionary (マッピング)

目的

コード化する 1 つの verbatim アイテムのマッピングを指定する CodingMap 構造内のコード化ディクショナリを識別します。

構文

```
<DICTIONARY
  TYPE="text"
  VERSION="text"
  CULTURE="text"/>
```

属性

TYPE="text"

ディクショナリのタイプ (MedDRA、WHO-DD など)。必須

VERSION="text"

ディクショナリのバージョン (8.1、05Q4 など)。必須

CULTURE="text"

言語とカルチャ (en-JP など)。必須

注： **TYPE**、**VERSION**、および **CULTURE** の値の組み合わせでディクショナリを一意に識別します。

例

```
<CODINGMAP REFNAME="MAPPINGS3" VERBATIMTYPE="MEDPROD"> <DICTIONARY
  TYPE="WHODD" VERSION="05Q4" CULTURE="en-US"/> <CODE
  TARGET NAME="ATC
  1.CODE" PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabDate.mcalLAB"/>
  <CONTEXTINFORMATION> <CONTEXTITEM NAME="Indication"
  PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabHi.mcalLAB"/>
  <CONTEXTITEM NAME="Route Of Administration"
  PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabTest.mcalLAB"/>
</CONTEXTINFORMATION></CODINGMAP>
```

ExternalMap

目的

InForm スタディの 1 つのコントロールをマッピングするためのラッパー。ExternalMap エレメントには、以下のエレメントを含めることができます。

- 1 つの PATH エレメント。InForm スタディのソースコントロールを識別するか、または InForm スタディに対応するコントロールがないマッピングターゲットが存

在することを示します。例：

- CDD に作成されたカラムで、CRF に関連するコントロールを持っていないもの。
- Clintrial プロトコルにおいてすべてのパネルに共通のアイテム。
- それぞれのマッピングターゲットを指定する複数のエレメント。

構文

```
<EXTERNALMAP>
  <PATH/>
  <CDD* attributes/>
  <CONTEXTPANEL* attributes/>
  <CTPANEL* attributes/>
  <CODINGMAP* attributes/>
</EXTERNALMAP>
```

子

ExternalMap エレメントには、以下のマッピング定義エレメントを含めることができます。

- Path
- CDD
- ContextPanel
- CTPanel
- CodingMap

例

以下は、CLIN1 マッピングの中ですべてのパネルに共通しているアイテムのマッピング定義をする、明確にされていない PATH エレメントを使用した ExternalMap エレメントの例です。アイテムは共通のため、InForm スタディの特定の RefName パスには対応しません。

```
<EXTERNALMAP>
  <PATH/>
  <CONTEXTPANEL REFNAME="CLIN1">
    <ITEM REFNAME="SUBJECT" ITEMDATATYPE="TEXT"
      ITEMDATAMAXLENGTH="15" CONTEXTTYPE="1"/>
    <ITEM REFNAME="VISITITEM" ITEMDATATYPE="TEXT"
      ITEMDATAMAXLENGTH="15" CONTEXTTYPE="2"/>
    <ITEM REFNAME="PAGEITEM" ITEMDATATYPE="TEXT"
      ITEMDATAMAXLENGTH="15" CONTEXTTYPE="3"/>
  </CONTEXTPANEL>
</EXTERNALMAP>
```

以下の例では、ExternalMap エレメントは PATH エレメントによって指定された InForm スタディのコントロールから、CDD エレメントで指定される CDD のテーブルカラムへのマッピングを定義しています。

```
<EXTERNALMAP>
  <PATH>
```

```

    <CHAPTERREF REFNAME="PF_ALL_VISITS"/>
    <PAGEREF REFNAME="ECG"/>
    <SECTIONREF REFNAME="CHESTXRAY"/>
    <ITEMSETREF REFNAME="1"/>
    <ITEMREF REFNAME="INTERPRET2"/>
    <CONTROLREF REFNAME="INTERPRETRADIO2"/>
    <CONTROLREF REFNAME="DESCRIBETEXT"/>
  </PATH>
  <CDD REFNAME="CDD1" KEYTYPE="PATIENT" TARGETTABLE="t_ECG"
    TARGETKEYTYPE="PATIENTVISIT" TARGETCOLUMN="COMMONDAT1"
    TARGETCOLUMNTYPE="TEXT"/>
</EXTERNALMAP>

```

次の例は、Central Coding アプリケーションでコード化するように PATH エlementによって指定されている ExternalMap Elementで、InForm スタディの 1 つの verbatim のマッピングを識別します。

```

<EXTERNALMAP xmlns="MedML-TDE">
  <PATH>
    <CHAPTERREF REFNAME="vstCORE4"/>
    <PAGEREF REFNAME="frmChem"/>
    <SECTIONREF REFNAME="sctChem"/>
    <ITEMSETREF REFNAME="mitsLabInfo"/>
    <ITEMREF REFNAME="mitmAccNo"/>
    <CONTROLREF REFNAME="mcalLAB"/>
  </PATH>
  <CODINGMAP REFNAME="MAPPINGS3" VERBATIMTYPE="MEDPROD">
    <DICTIONARY TYPE="WHODD" VERSION="05Q4" CULTURE="en-US"/>
    <CODETARGET NAME="ATC 1.CODE"
      PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabDate.mcalLAB"/>
  <CONTEXTINFORMATION>
    <CONTEXTITEM NAME="Indication"
      PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabHi.mcalLAB"/>
    <CONTEXTITEM NAME="Route Of Administration"
      PATH="0.vstCORE4.frmChem.sctChem.mitsLabInfo.mitmLabTest.mcalLAB"/>
  </CONTEXTINFORMATION>
</CODINGMAP>
</EXTERNALMAP>

```

ExternalMapSet

目的

1 つの RefName を持つ 1 セットの外部マッピングを識別します。オプション

構文

```

<EXTERNALMAPSET
  REFNAME="name"
  [DESIGNNOTE="text"]
  [ACTIVE="true|false"]
  TYPE="CDD|CLINFORM|ORACLIN|AUTOCODE|CODINGMAP"

```

```
[AUTOGEN="true|false"]/>
```

属性

REFNAME="*name*"

ExternalMapSet の RefName。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

ACTIVE="*true|false*"

コンポーネントが有効であるかどうかを示します。オプションは true または false です。デフォルトは、True になっています。オプション

TYPE="CDD|CLINFORM|CODINGMAP"

マップされるターゲットエンティティを示します。値は以下の通りです。

- **CDD** — 外部オラクルデータベース内でのテーブルカラム。
- **CLINFORM** — Clintrial スタディ定義におけるパネル中のアイテム。
- **CODINGMAP** — Central Coding アプリケーション。

必須

例

以下の例は MEDIKA-CLINICAL サンプルスタディを定義する MedML の ExternalMapSet エlementを示しています。

```
<EXTERNALMAPSET REFNAME="MEDIKA_MAP"TYPE="CLINFORM" ACTIVE="true"
  DESIGNNOTE="My Design note"/>
```

Itemref

目的

- 定義済みアイテムをセクション、アイテムセット、またはアイテムグループの定義に含めます。セクション、アイテムセット、またはアイテムグループの定義の各アイテムにつき 1 つの Itemref Elementを含めます。
- マッピング定義の Path Elementで、マッピングされるコントロールの位置として、アイテムの RefName を指定します。Path Elementによって定義された RefName パスでは、1 つの Itemref Elementを含めてください。

Itemref は Section または Path のチャイルドとしてのみ定義されます。単独のコンポーネントとしてサブミットされません。

構文

```
<ITEMREF
  REFNAME="name"
```

```
[KEYITEM="true|false"]
[UNIQUEKEY="true|false"]
[ORDER="n"]/>
```

属性

REFNAME="name"

セクションまたはパスの定義に含まれるアイテムの RefName。必須

KEYITEM="true|false"

Itemref によって含められるアイテムが、Itemset 定義のキーアイテムであるかどうかを示します。キーアイテムの値は、アイテムセットの特定のインスタンスをユーザがナビゲートできるようにするため、InForm アプリケーションのドロップダウンリストに表示されます。テキスト、数値、または日付/時間の値を持つアイテムのみがキーアイテムとなります。チェックボックスコントロール、グループコントロール、ラジオコントロール、ドロップダウンリストなどのコンパウンドコントロールを持つアイテムを、キーアイテムにすることはできません。オプションは true または false です。デフォルトは、False になっています。オプション

注： 繰り返しフォームのキーアイテムを指定するには、**KeyItemref** エレメントを使用します。アイテムセットのキーアイテムとして定義されたアイテムは、繰り返しフォームのキーアイテムにすることはできません。

UNIQUEKEY="true|false"

キーアイテムがフォームセット、フォーム、およびアイテムセットの中でユニークであるべきかどうかを true または false で示します。キーアイテムがユニークであると定義されている状態で、アイテムセットの 2 つの行が同じキーアイテム値で、同じビジットとフォームでサブミットされると、InForm アプリケーションは 2 番目のインスタンスの入力を受け付けません。オプションは true または false です。デフォルトは、False になっています。オプション

ORDER="n"

Section 定義の中で、各 Itemref が現れる順序。順序を指定しない場合、MedML インストラユーティリティは、Itemref が参照するアイテムを入力された順番で順序付けします。オプション

注： Path エレメントが Itemref 定義を含む場合、**REFNAME** 属性のみが有効です。

例

以下は、高血圧症の期間（Duration of Hypertension）というセクションの定義（2 つのアイテムを含む）における Itemref の使用例です。詳細については、「**Section**」（195ページ）を参照してください。

```
<SECTION REFNAME="HYPERTENSION"
  TITLE="Duration of Hypertension">
  <ITEMREF REFNAME="HTYESNO" ORDER="1"/>
  <ITEMREF REFNAME="HTDURATION" ORDER="2"/>
```

</SECTION>

以下は、Path エlement での Itemref 定義の使用例です。CLIN アイテムはマップされた TESTTEXT コントロールが存在する位置です。

<PATH>

<CHAPTERREF REFNAME="PF_ALL_VISITS"/>

<PAGEREF REFNAME="LAB"/>

<SECTIONREF REFNAME="LAB"/>

<ITEMSETREF REFNAME="LAB"/>

<ITEMREF REFNAME="CLIN"/>

<CONTROLREF REFNAME="CCGROUP"/>

<CONTROLREF REFNAME="TESTTEXT"/>

</PATH>

以下は、Itemref Element を持つ ItemSet 定義に、3 つのキーアイテムをユニークキーの組み合わせとして指定する例です。

<ITEMSET REFNAME="itsDOSE" ITEMREQUIRED="true"

SDVREQUIRED="true" UNIQUEKEY="true">

<ITEMREF REFNAME="itmDOSEFromDate" ORDER="1"

UNIQUEKEY="true" KEYITEM="true"/>

<ITEMREF REFNAME="itmDOSEToDate" ORDER="2"

UNIQUEKEY="true" KEYITEM="true"/>

<ITEMREF REFNAME="itmDOSEBlisterpack" ORDER="3"

UNIQUEKEY="true" KEYITEM="true"/>

<ITEMREF REFNAME="itmDOSETotalCaps" ORDER="4"/>

<ITEMREF REFNAME="itmDOSENum" ORDER="5"/>

<ITEMREF REFNAME="itmDOSEMissed" ORDER="6"/>

<ITEMREF REFNAME="itmDOSEReasChange" ORDER="7"/>

<ITEMREF REFNAME="Item_InclComments" ORDER="8"/>

</ITEMSET>

ItemSetref

目的

マッピング定義の Path エlementでのマップされたコントロールの位置としてアイテムセットの RefName を指定します。Path Elementによって定義された RefName パスに 1 つの ItemSet Elementを含めます。

構文

```
<ITEMSETREF  
  REFNAME="name"/>
```

属性

REFNAME="name"

マップされたソースコントロールが存在するアイテムセットの RefName。必須

例

この例ではマップされた TESTTEXT コントロールが存在する位置として、LAB アイテムセットを示しています。

```
<PATH>  
  <CHAPTERREF REFNAME="PF_ALL_VISITS"/>  
  <PAGEREF REFNAME="LAB"/>  
  <SECTIONREF REFNAME="LAB"/>  
  <ITEMSETREF REFNAME="LAB"/>  
  <ITEMREF REFNAME="CLIN"/>  
  <CONTROLREF REFNAME="CCGROUP"/>  
  <CONTROLREF REFNAME="TESTTEXT"/>  
</PATH>
```


Pageref

目的

マッピング定義の Path エlement でのマップされたコントロールの位置としてフォームの RefName を指定します。Path Element によって定義された RefName パスに 1 つの Pageref Element を含めます。

構文

```
<PAGEREF
  REFNAME="name"/>
```

属性

REFNAME="*name*"

マップされたソースコントロールが存在するフォームの RefName。必須

例

以下の例は、マッピングされた COMMONDATE コントロールが発生する位置としての ECG フォームを示します。

```
<PATH>
  <CHAPTERREF REFNAME="PF_ALL_VISITS"/>
  <PAGEREF REFNAME="ECG"/>
  <SECTIONREF REFNAME="LEADECG"/>
  <ITEMSETREF REFNAME="1"/>
  <ITEMREF REFNAME="DATEASSESS"/>
  <CONTROLREF REFNAME="COMMONDATE"/>
</PATH>
```

Path

目的

- Path Element はチャイルドコンポーネントなしで単独で使用され、InForm スタディに対応するコントロールがないマッピングターゲットが存在することを示します。例：
 - CDD に作成されたカラムで、CRF に関連コントロールを持っていないもの。
 - Clintrial プロトコルにおいてすべてのパネルに共通のアイテム。

- Path エlementは 1 つ以上のチャイルドコンポーネント（各々が RefName パスの一部となっている RefName を指定する）と共に使用され、InForm スタディでのコントロールを指定します。このコントロールは1つのデータマッピングに対するソースデータを特定します。

Path Elementは、ExternalMap Elementによって作成されるマッピング定義に必要なコンポーネントであり、ExternalMap 定義において最初のチャイルドコンポーネント Elementである必要があります。

構文

```
<PATH>
  <CHAPTERREF attributes/>
  <PAGEREF attributes/>
  <SECTIONREF attributes/>
  <ITEMSETREF attributes/>
  <ITEMREF attributes/>
  <CONTROLREF* attributes/>
</PATH>
```

子

Path Elementのチャイルドコンポーネントはそれぞれオプションですが、InForm スタディのソースコントロールの RefName パスの一部となっている RefName を指定します。

- Chapterref
- Pageref
- Sectionref
- Itemsetref
- Itemref
- Controlref

例

この例は、マッピングにおけるすべての Clintrial パネルに共通のアイテムのマッピング定義において、チャイルドコンポーネントなしで Path Elementを使用する方法を示しています。

```
<EXTERNALMAP>
  <PATH/>
  <CONTEXT PANEL REFNAME="MEDIKA_MAP">
    <CTITEM REFNAME="PATNUM" ITEM DATATYPE="TEXT" ISREPEAT="false"
      ISREQUIRED="true" DBFORMAT="VARCHAR2(20)" CONTEXTTYPE="1" />
    <CTITEM REFNAME="VISITID" ITEM DATATYPE="TEXT" ISREPEAT="false"
      ISREQUIRED="true" DBFORMAT="VARCHAR2(20)" CONTEXTTYPE="2" />
    <CTITEM REFNAME="FORMID" ITEM DATATYPE="TEXT" ISREPEAT="false"
      ISREQUIRED="true" DBFORMAT="VARCHAR2(20)" CONTEXTTYPE="3" />
    <CTITEM REFNAME="VISITINDEX" ITEM DATATYPE="FLOAT" ISREPEAT="true"
      ISREQUIRED="true" DBFORMAT="NUMBER(5,2)" CONTEXTTYPE="2" />
  </CONTEXT PANEL REFNAME="MEDIKA_MAP">
</PATH/>
</EXTERNALMAP>
```

```
</CONTEXT PANEL>
</EXTERNAL MAP>
```

Sectionref

目的

- 定義済みセクションをフォームの定義に含めます。
- フォームのセクションごとに 1 つの Sectionref エlementを含めます。
- マッピング定義の Path Elementでのマップされたコントロールの位置としてセクションの RefName を指定します。
- Path Elementによって定義された RefName パスに 1 つの Sectionref Elementを含めます。

Sectionref は Form または Path Elementのチャイルドとしてのみ定義されます。単独のコンポーネントとしてサブミットされません。

構文

```
<SECTIONREF
  REFNAME="name"
  [ORDER="n"]/>
```

属性

REFNAME="name"

フォームまたはパスの定義に含まれるセクションの RefName。必須

ORDER="n"

フォーム定義における各 Sectionref の順序。順序を指定しない場合、MedML インストラユーティリティは、Sectionref が参照するアイテムを入力された順番で順序付けします。オプション

注： Path Elementが Sectionref 定義を含む場合、**REFNAME** 属性のみが有効です。

例

この例は症例背景フォームの定義を示しています。2 つのセクション、症例背景 (DEM; Demographics) および喫煙履歴 (SH; Smoking History) から構成されます。

```
<FORM REFNAME="DEM" TITLE="Demographics" MNEMONIC="DEM"
  FORMTYPE="CRF">
  <SECTIONREF REFNAME="DEM"/>
  <SECTIONREF REFNAME="SH"/>
</FORM>
```

以下の例は、マッピングされた COMMONDATE コントロールが出現する位置として LEAD ECG セクションを示します。

```
<PATH>
```

```
<CHAPTERREF REFNAME="PF_ALL_VISITS"/>
<PAGEREF REFNAME="ECG"/>
<SECTIONREF REFNAME="LEADEC"/>
<ITEMSETREF REFNAME="1"/>
<ITEMREF REFNAME="DATEASSESS"/>
<CONTROLREF REFNAME="COMMONDATE"/>
</PATH>
```

MedMLスキーマリファレンス

Action

目的

関連付けられているイベントが発生したときのアクションを指定します。

構文

```
<ACTION  
  [NAME="name"]  
  VALUE="Query"  
  [QUERYTYPE="OPEN|CANDIDATE"]  
  [LANGUAGE="name"]  
  [QUERYTEXT="text"]/>
```

属性

NAME="name"

イベントの定義で **Action** を参照するときに使用される名前です。PF_Event_Action のみで有効。

VALUE="Query"

実行タイプ。必須。Query のみ使用可能。

QUERYTYPE="OPEN|CANDIDATE"

クエリを未解決クエリ (OPEN) または仮登録クエリ (CANDIDATE) のどちらで作成するかを指定します。オプション

LANGUAGE="name"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

QUERYTEXT="*text*"

関連付けされたイベントが発生した際に作成されるクエリのテキスト。オプション

例

以下の例は、クエリテキストを「範囲外の値」とし、未解決状態のクエリを生成するアクションを示しています。Action は EScript という実行計画を実行するイベントの定義に含まれています。

```
<EVENT REFNAME="範囲外の値">
  <ACTION NAME="PF_Event_Action"
    VALUE="Query"QUERYTYPE="Open"
    QUERYTEXT="範囲外の値"/>
  <EXECUTIONPLANREF REFNAME="EScript"/>
</EVENT>
```

AttachRuleDepend

目的

汎用ルールと、アイテム、アイテムセット、またはフォームとの間の依存関係を定義します。AttachRuleDepend グループの中の属性とチャイルドコンポーネントは、ルールが実行されるコンテキストに対する依存関係を定義します。

コンテキストは特定の ビジット、フォーム、セクション、アイテムセット/アイテムに対して固有のものにするか、次のいずれかの方法で一般的な範囲を持たせることが可能です。

- **一般的なビジットコンテキスト** — ビジットをルールと関連付けます。
- **一般的なフォームコンテキスト** — そのフォームが定義されているすべてのビジットでのフォーム/セクション/アイテムの組み合わせとルールを関連付けます。
- **一般的なアイテムコンテキスト** — アイテムが定義されるすべての場所でアイテムとルールを関連付けます。

一般的な依存関係を作成するには、AttachRuleSet エLEMENTの **FORMSETNAME**、**FORMNAME**、**SECTIONNAME** と **ITEMNAME** 属性の定義で説明される特別な「*name*」値をご使用ください。

注： コンテキストの各依存性はそのコンテキストと同じスコープを持つ必要があります。例えば、コンテキストが一般のビジットコンテキストとして定義されている場合、そのコンテキストのどの依存性も一般のビジット属性を持つ必要があります。

2 つの繰り返しフォーム間の関係がルールコンテキストの依存関係になる可能性があります。関連付けをコンテキストの依存関係として定義した場合、ルールが実行されるのは、ユーザが、繰り返しフォームの特定のインスタンスとその関連内の他のフォームのとの間の関係を作成または解除するチェックボックスコントロールを選択または選択解除するときです。関連付けに対して定義される依存関係は以下の通りである必要があります。

- 特定ビジットコンテキストに付加されている。
- トリガ依存として定義されている。

構文

```
<ATTACHRULEDEPEND
  FORMSETNAME="name"
  FORMNAME="name"
  SECTIONNAME="name"
  ITEMSETNAME="name"
  ITEMNAME="name"
  [ATTACHTYPE="APPLIED|DEPENDENCY|TRIGGER"]>
```

属性

FORMSETNAME="name"

依存関係を指定するフォームセットの RefName。フォーム、セクション、アイテムがその組み合わせで発生するすべてのフォームセット内の特定のフォーム、セクション、アイテムに対してルールを付加する場合（一般的なフォームコンテキスト）は、**FORMSETNAME** 値として「PF_ALL_VISITS」を使用してください。依存関係として関連を定義するには、関連付ける RefName を **FORMSETNAME** 値として使用します。必須

FORMNAME="name"

依存関係を指定するフォームの RefName。そのアイテムが定義されているすべてのフォームセット、フォーム、およびセクションのアイテムにルールを付加する場合（一般的なアイテムコンテキスト）、または、依存関係として関連付けを定義する場合は、**FORMNAME** 値として「PF_ALL_FORMS」を使用してください。必須

SECTIONNAME="name"

依存関係を指定するセクションの RefName。そのアイテムが定義されているすべてのフォームセット、フォーム、およびセクションのアイテムにルールを付加する場合（一般的なアイテムコンテキスト）、または、依存関係として関連付けを定義する場合は、**SECTIONNAME** 値として「PF_ALL_SECTIONS」を使用してください。必須

ITEMSETNAME="name"

依存関係を指定するアイテムを含むアイテムセットの名前。この名前は ItemSet 定義の RefName 属性に指定されています。アイテムセット内のアイテムへの依存関係に対しては必須。

ITEMNAME="name"

依存関係を指定するアイテムの名前。この名前は Item 定義の RefName 属性に指定されています。必須

ATTACHTYPE="APPLIED|DEPENDENCY|TRIGGER"

使用する依存関係のタイプ。オプションは以下の通りです。

- **APPLIED** — 初期のエントリ後または変更後にアイテムまたはアイテムセットがサブミットされたときにルールが有効になります。
- **DEPENDENCY** — 依存するアイテムまたはアイテムセットがデータを含むときにみにルールが有効になります。
- **TRIGGER** — 依存するアイテムまたはアイテムセットがデータを含まないときもルールが有効になります。依存関係として定義された関連の **ATTACHTYPE** は **TRIGGER** である必要があります。

デフォルトは **APPLIED** です。オプション

AttachRuleSet

目的

ビジット、フォームのアイテムまたはアイテムセットとルールを、または、2 つの関連するフォーム間の関連とルールを関連付けます。初期のエントリ後または変更後にアイテムまたはアイテムセットがサブミットされたときに、そのアイテムまたはアイテムセットに関連付けられているアクティブなルールが実行されます。関連付けに付加されたルールが実行されるのは、ユーザが、繰り返しフォームの特定のインスタンスとその関連内の他のフォームとの間の関係を、作成または解除するチェックボックスコントロールを選択または選択解除するときです。AttachRuleSet グループにある属性とチャイルドコンポーネントはルールの実行時のコンテキストを形成します。

コンテキストは特定の ビジット、フォーム、セクション、アイテムセット/アイテムに対して固有のものにするか、次のいずれかの方法で一般的な範囲を持たせることが可能です。

- **一般的なビジットコンテキスト** — ビジットをルールと関連付けます。
- **一般的なフォームコンテキスト** — そのフォームが定義されているすべてのビジットでのフォーム/セクション/アイテムの組み合わせとルールを関連付けます。
- **一般的なアイテムコンテキスト** — アイテムが定義されるすべての場所でアイテムとルールを関連付けます。

一般的な依存関係を作成するには、AttachRuleSet エLEMENTの **FORMSETNAME**、**FORMNAME**、**SECTIONNAME** と **ITEMNAME** 属性の定義で説明される特別な「*name*」値をご使用ください。

注： コンテキストの各依存性はそのコンテキストと同じスコープを持つ必要があります。例えば、コンテキストが一般のビジットコンテキストとして定義されている場合、そのコンテキストのどの依存性も一般のビジット属性を持つ必要があります。

無作為割付ルールのコンテキストは 1 つだけです。

構文

```
<ATTACHRULESET
  REFNAME="name"
  RULENAME="name"
  FORMSETNAME="name"
  FORMNAME="name"
  SECTIONNAME="name"
  ITEMSETNAME="name"
  ITEMNAME="name"
  [HELPTTEXT="text"]
  [ATTACHTYPE="APPLIED|DEPENDENCY|TRIGGER"]
  [ACTIVE="true|false"]
  <RULEARG* attributes/>
  <ATTACHRULEDEPEND* attributes/>
  <EVENTREF attributes/>
</ATTACHRULESET>
```

属性

REFNAME="name"

AttachRuleSet の名前。各ルールに対してこの名前は一意でなければなりません。必須

RULENAME="name"

アイテムに付加するルールの名前。この名前は Rule 定義の RefName 属性に指定されています。必須

FORMSETNAME="name"

そのアイテムが存在するフォームセットの名前。この名前は FormSet 定義の **REFNAME** 属性に指定されています。フォーム、セクション、アイテムがその組み合わせで発生するすべてのフォームセット内の特定のフォーム、セクション、アイテムに対してルールを付加する場合 (一般的なフォームコンテキスト) は、**FORMSETNAME** 値として「PF_ALL_VISITS」を使用してください。必須

FORMNAME="name"

そのアイテムが存在するフォームの名前。この名前は Form 定義の **REFNAME** 属性に指定されています。そのアイテムが定義されているすべてのフォームセット、フォーム、セクションのアイテムにルールを付加する場合 (一般的なアイテムコンテキスト) は、**FORMNAME** 値として「PF_ALL_FORMS」を使用してください。必須

SECTIONNAME="name"

アイテムが存在するフォームのセクションの名前。この名前は Section 定義の **REFNAME** 属性に指定されています。そのアイテムが定義されているすべてのフォームセット、フォーム、セクションのアイテムにルールを付加する場合 (一般的なアイテムコンテキスト) は、**SECTIONNAME** 値として「PF_ALL_SECTIONS」を使用してください。必須

ITEMSETNAME="*name*"

ルールが付加されているアイテムを含むアイテムセットの名前。この名前は ItemSet 定義の **REFNAME** 属性に指定されています。アイテムセット内のアイテムに付加されているルールに対して必須。

ITEMNAME="*name*"

ルールが付加されているアイテムの名前。この名前は Item 定義の **REFNAME** 属性に指定されています。必須

HELPTEXT="*text*"

ルールによって実行されるデータチェックの説明。ルールが付加されているデータアイテムの CRF ヘルプに、この説明が表示されます (CRF ヘルプが適切にマッピング定義されている場合)。AttachRuleSet エレメントで指定されたヘルプテキストはルールで指定されるヘルプテキストよりも優先されます。オプション

ATTACHTYPE="APPLIED|DEPENDENCY|TRIGGER"

使用する依存関係のタイプ。オプションは以下の通りです。

- **APPLIED** — 初期のエントリ後または変更後にアイテムまたはアイテムセットがサブミットされたときにルールが有効になります。
- **DEPENDENCY** — 依存するアイテムまたはアイテムセットがデータを含むときのみにルールが有効になります。
- **TRIGGER** — 依存するアイテムまたはアイテムセットがデータを含まないときもルールが有効になります。

デフォルトは **APPLIED** です。オプション

ACTIVE="*true|false*"

コンポーネントが有効であるかどうかを示します。オプションは true または false です。デフォルトは、True になっています。オプション

子

AttachRuleSet の定義には、次のエレメントを含めることができます。

- Rulearg エレメント。エレメントそれぞれがルールのパラメータを定義します。
- AttachRuleDepend エレメント。各 AttachRuleDepend エレメントは、依存関係のタイプを示すか、またはルールが依存するアイテムまたは関連付けを示します。
- Eventref エレメント。Eventref エレメントを利用してルールの定義にイベント定義を含めることができます。AttachRuleSet で指定された Eventref はルールで指定される Eventref よりも優先されます。

例

AttachRuleSet 定義で、症例背景フォームの体重 (Weight) アイテムに「Range Checking Rule」と呼ばれるルールの定義をどのように付加するかを以下の例で説明しています。

<MEDMLDATA>

```

<RULE REFNAME="Range Checking Rule"
  DESCRIPTION="Range Checking Rule"
  ENABLED="true"
  SCRIPTTYPE="SERVERRULE"
  SCRIPTFILE="RangeCheck.vbs"
  HELPTEXT="Value of the item have to be in the specified range.">
  <EVENTREF REFNAME="Value Out of Range"/>
</RULE>

<ATTACHRULESET RULENAME="Range Checking Rule"
  REFNAME="attach1"
  FORMSETNAME="Visit1"
  FORMNAME="DEM"
  SECTIONNAME="DEM"
  ITEMNAME="WEIGHT"
  ATTACHTYPE="APPLIED"
  ACTIVE="true">
  <RULEARG NAME="ItemName" VALUE="0.Visit1.DEM.DEM.0.WEIGHT"
TYPE="STRING"/>
  <RULEARG NAME="Min" VALUE="90" TYPE="NUMERIC"/>
  <RULEARG NAME="Max" VALUE="275" TYPE="NUMERIC"/>
</ATTACHRULESET>

<ATTACHRULESET RULENAME="Range Checking Rule"
  REFNAME="attach2"
  FORMSETNAME="Visit1"
  FORMNAME="DEM"
  SECTIONNAME="DEM"
  ITEMNAME="HEIGHT"
  ATTACHTYPE="APPLIED"
  ACTIVE="true">
  <RULEARG NAME="ItemName" VALUE="0.Visit1.DEM.DEM.0.HEIGHT"
TYPE="STRING"/>
  <RULEARG NAME="Min" VALUE="30" TYPE="NUMERIC"/>
  <RULEARG NAME="Max" VALUE="75" TYPE="NUMERIC"/>
</ATTACHRULESET>

<RULE REFNAME="dynatest"
  DESCRIPTION="Dynamic Visit Test"
  ENABLED="true"
  SCRIPTTYPE="SERVERCALCULATION"
  SCRIPTFILE="dynaset.vbs"
  HELPTEXT="Schedule Dynamic Visit"/>

<ATTACHRULESET RULENAME="dynatest"
  REFNAME="attach1"
  FORMSETNAME="Visit1"
  FORMNAME="DEM"
  SECTIONNAME="DEM"
  ITEMNAME="GENDER"
  ATTACHTYPE="APPLIED"

```

```
ACTIVE="true"/>

</MEDMLDATA>
```

例 2

以下の例は、関連ルール定義 (%PFHELPARG% 文字で始まるヘルプ文字列引数を一つ以上定義している) のヘルプテキスト文字列への引数の渡し方を説明しています。たとえば、一般的なヘルプテキスト文字列へ最小値と最大値を渡す場合、以下の引数を定義することができます。

```
<RULEARG NAME="%PFHELPARG%MEASURE" VALUE="Height" TYPE="STRING"/>
<RULEARG NAME="%PFHELPARG%MIN" VALUE="48" TYPE="NUMERIC"/>
<RULEARG NAME="%PFHELPARG%MAX" VALUE="96" TYPE="NUMERIC"/>
```

これらの引数により、以下のルール定義の HELPTEXT 文字列を置き換えます。

```
%PFHELPARG%MEASURE must be between %PFHELPARG%MIN and
%PFHELPARG%MAX.
```

ブラウザ

目的

サポートされているブラウザのタイプを指定します。

注： このコンポーネントは、製品出荷前の **InForm** データベースロードに使用されるデフォルト **XML** ファイルでのみ使用されます。デフォルトブラウザコンポーネント定義は **Oracle** 用に予約されています。

構文

```
<BROWSER
  BROWSERNAME="name"
  BROWSETYPE="type" />
```

属性

BROWSERNAME="*name*"

ブラウザ名。必須

BROWSETYPE="*type*"

ブラウザのタイプ。これは Oracle によって定義された値です。必須

例

以下の例は、Internet Explorer バージョン 9.0 用のブラウザコンポーネント定義に使用される XML コードです。

```
<BROWSER BROWSERNAME="IE090" BROWSETYPE="16897"/>
```

CalculatedControl

目的

1 つ以上の他のコントロールの値に基づく値を持つコントロールを定義します。

構文

```
<CALCULATEDCONTROL  
  REFNAME="name"  
  [DESIGNNOTE="text"]  
  [NAME="name"]  
  [UUID="id"]  
  [CAPTION="text"]  
  [LANGUAGE="name"]  
  [MAXLENGTH="n"]  
  [CAPTIONALIGN="LEFT | RIGHT | TOP | BOTTOM"]  
  [ALIGN="LEFT | CENTER | RIGHT | TOP | MIDDLE | BOTTOM"]  
  [FORMAT="return_format"]  
  [UNITDISPLAYTYPE="ELEMENT">  
    <UNITREF attributes/>  
    <TRANSLATIONS/>  
</CALCULATEDCONTROL>
```

属性

REFNAME="*name*"

他のコントロール定義で計算コントロールを参照するときに使用される名前。必須この名前は計算コントロール間で一意である必要があります。

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

NAME="*name*"

他のコントロール定義で計算コントロールを参照するときに使用される名前。
オプション

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

CAPTION="*text*"

計算された値と一緒に表示されるテキスト。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

MAXLENGTH="*n*"

計算値の最大の長さ。オプション

CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"

計算値に対する相対的なキャプションの位置: LEFT (左)、RIGHT (右)、TOP (上)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"

コントロール内の計算値の位置調整: LEFT (左)、CENTER (中央)、RIGHT (右)、TOP (上)、MIDDLE (中央)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

FORMAT="*return_format*"

計算コントロールによって定義されたデータアイテムに付加されたルールによって返されるデータのフォーマット。計算値の位置を示すには、**%s** を使用してください。

UNITDISPLAYTYPE="ELEMENT"

Unitref 定義を持つテキストコントロールに含まれる単位を表示するために使用されるコントロールのタイプ。「ELEMENT」のみが有効な値です。

注： フォームから計算アイテムを無効化する場合、計算ルールを非アクティブ化するか、もしくは計算に依存するすべてのアイテムのアイテム名を変更する必要があります。計算項目のみ取り除くと、ルールは失敗します。

子

- コントロールに関連付ける単位を指定する 0 または 1 つの Unitref エlement。
- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エlement には 1 つ以上の Translation 定義を含める必要があります。Translation エlement はそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エlement は、スタディコンポーネント定義内の最後のチャイルドElement にする必要があります。CalculatedControl 定義では、**CAPTION** 属性を翻訳できます。

例

以下の例は、ユーザが無作為割付ボタンをクリックすると、結果として生じる無作為割付文字列を表示するのに使用されるコントロールの定義を説明しています。

RANDOMIZATION 計算コントロールは DRUGKIT アイテムの定義に含まれています。DRUGKIT アイテムに付いた無作為割付ルールが実行されると、無作為割付文字列を返します。

注： InForm アプリケーションと MedML インストラユーティリティは、**UUID** の英字を大文字に変換します。

さらに以下の例は、計算コントロールのキャプションの翻訳文字列の定義を示します。

```
<CALCULATEDCONTROL REFNAME="RANDOMIZATION"
  UUID="DC2EB0BF-4F12-11d2-9319-00A0C9769A13"
  LANGUAGE="en-US"
  CAPTION="Drug Kit"
```

```

NAME="RANDOMIZATION"
FORMAT="Drug-kit number %s has been assigned to this patient">
  <TRANSLATIONS>
    <TRANSLATION NAME="CAPTION" LOCALE="fr-FR"
      DISPLAYTEXT="Kit de Drogues"/>
  </TRANSLATIONS>
</CALCULATEDCONTROL>
<ITEM REFNAME="DRUGKIT"
  UUID="52AF1207-4F13-11d2-9319-00A0C9769A13"
  QUESTION="Sequence Number and Information: "
  ITEMREQUIRED="false"
  CALCULATED="true">
  <CONTROLREF REFNAME="RANDOMIZATION" />
</ITEM>

```

CheckboxControl

目的

ユーザが 1 つ以上のオプションを選択できるチェックボックスのリストを定義します。チェックボックスコントロールは、Controlref の使用により挿入された定義済みコンポーネントから構成されています。CheckBoxControl 定義へ以下のタイプのコンポーネントを含めることができます：

- CalculatedControl
- CheckBoxControl
- GroupControl
- PullDownControl
- RadioControl
- SimpleControl
- TextControl

構文

```

<CHECKBOXCONTROL
  REFNAME="name"
  [DESIGNNOTE="text"]
  [NAME="name"]
  [UUID="id"]
  [ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"]
  [LAYOUT="VERTICAL|HORIZONTAL|NOWRAP"]
  [CAPTION="text"]
  [LANGUAGE="name"]
  [CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"]>
  <CONTROLREF+ attributes/>
  <TRANSLATIONS/>
</CHECKBOXCONTROL>

```


属性

REFNAME="*name*"

他のコントロール定義でチェックボックスコントロールを参照するときに使用される名前。必須。この名前はチェックボックスコントロールの中でユニークである必要があります。

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

NAME="*name*"

他のコントロール定義でチェックボックスコントロールを参照するときに使用される名前。オプション

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"

コントロール内の計算値の位置調整: LEFT (左)、CENTER (中央)、RIGHT (右)、TOP (上)、MIDDLE (中央)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

LAYOUT="VERTICAL|HORIZONTAL|NOWRAP"

チェックボックスの向き: 垂直 (Vertical)、水平 (Horizontal)、あるいはノーラップ (Nowrap)。デフォルトは NOWRAP (ノーラップ) です。NOWRAP (ノーラップ) を指定すると、ラジオボタンは水平方向に位置づけられ、ユーザがブラウザウィンドウのサイズを変更しても、ボタンは別の行に改行されません。オプション

CAPTION="*text*"

チェックボックスリストと共に画面に表示されるテキスト。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注: デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"

計算値に対する相対的なキャプションの位置:LEFT (左)、RIGHT (右)、TOP (上)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

子

- 1 つ以上の Controlref 定義。各 Controlref はチェックボックスのリストで 1 つのアイテムを識別する既に定義されたコンポーネントを参照します。

注： **Controlref** 定義を用いてコンパウンドコントロールを定義するとき、すべての下位コントロールが同じ **TYPE** 属性で定義され、同じデータタイプを返すようにしてください。さらに、コンパウンドコントロールを定義するとき、**InForm** アプリケーションは最大 **5** 層のネスティングをサポートしています。**5** 層での設計が可能です。パフォーマンスを向上させるためにネストレベルはできるだけ少なくしてください。

- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エlement には 1 つ以上の Translation 定義を含める必要があります。Translation エlement はそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エlement は、スタディコンポーネント定義内の最後のチャイルドElement にする必要があります。CalculatedControl 定義では、**CAPTION** 属性を翻訳できます。

例 1

この例は、シンプルコントロールに事前に定義された PFElement を含めるという簡単なチェックボックスコントロールの作成を示しています。この例でのチェックボックスコントロールは症例の喫煙習慣のリストです。

- 1 各チェックボックスコンポーネントを PFElement として定義し、シンプルコントロールに含めます。"ND" コンポーネントを一度定義すると、他のコントロールやフォームで再利用できます。
- 2

```
<PFELEMENT REFNAME="CIGARETTE" LABEL="Cigarettes" TYPE="STRING"
VALUE="CIGARETTE"/>
<PFELEMENT REFNAME="PIPE" LABEL="Pipe" TYPE="STRING"
VALUE="PIPE"/>
<PFELEMENT REFNAME="CIGAR" LABEL="Cigars" TYPE="STRING"
VALUE="CIGAR"/>
<PFELEMENT REFNAME="ND" LABEL="Not Done" TYPE="STRING"
VALUE="ND"/> <SIMPLECONTROL REFNAME="CIGARETTE"
NAME="CIGARETTE">
  <ELEMENTREF REFNAME="CIGARETTE"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="PIPE" NAME="PIPE">
  <ELEMENTREF REFNAME="PIPE"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="CIGAR" NAME="CIGAR">
  <ELEMENTREF REFNAME="CIGAR"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="ND" NAME="ND">
  <ELEMENTREF REFNAME="ND"/>
```

- 3 Controlref を使用して各シンプルコントロールを含むチェックボックスを定義します。

```
4 <CHECKBOXCONTROL REFNAME="SMOKING"  
    NAME="SMOKING"  
    LAYOUT="VERTICAL">  
    <CONTROLREF REFNAME="CIGARETTE" ORDER="1"/>  
    <CONTROLREF REFNAME="PIPE" ORDER="2"/>  
    <CONTROLREF REFNAME="CIGAR" ORDER="3"/>  
    <CONTROLREF REFNAME="ND" ORDER="4"/>  
</CHECKBOXCONTROL>
```

例 2

この例は、ネスト化されたコンポーネントを持つチェックボックスコントロールを示しています。サンプルチェックボックスコントロールは筋肉痛についてのデータ収集向けのものです。

- 1 「左」、「右」または「その他」のテキストボックス用のテキストコントロールを定義します。RIGHT テキストコントロールとの位置合わせをコントロールするスペースを追加するために、LEFT テキストコントロールは という HTML 特殊文字を使用しています。

```
2 <TEXTCONTROL REFNAME="LEFT" NAME="LEFT"  
    CAPTION="Left&nbsp;&nbsp;&nbsp;" CAPTIONALIGN="LEFT"  
    HEIGHT="1"  
    LENGTH="32"  
    MAXLENGTH="50"  
    DATATYPE="STRING"/>  
<TEXTCONTROL REFNAME="RIGHT" NAME="RIGHT"  
    CAPTION="Right : "CAPTIONALIGN="LEFT"  
    HEIGHT="1"  
    LENGTH="32"  
    MAXLENGTH="50"  
    DATATYPE="STRING"/>  
<TEXTCONTROL REFNAME="OTHER" NAME="OTHER"  
    CAPTION="Other (describe):"CAPTIONALIGN="TOP"  
    HEIGHT="1"  
    LENGTH="32"  
    MAXLENGTH="50"  
    DATATYPE="STRING"/>
```

- Controlref を使用し、「手」および「足」用のチェックボックスの CheckBoxControl を定義します。各チェックボックスコントロールに左用、右用のテキストコントロールを使用できます。

```
4 <CHECKBOXCONTROL REFNAME="ARMS" NAME="ARMS"
  LAYOUT="VERTICAL"
  CAPTION="Arms (describe):"
  CAPTIONALIGN="LEFT">
  <CONTROLREF REFNAME="LEFT" ORDER="1"/>
  <CONTROLREF REFNAME="RIGHT" ORDER="2"/>
</CHECKBOXCONTROL>
```

```

<CHECKBOXCONTROL REFNAME="LEGS" NAME="LEGS"
  LAYOUT="VERTICAL"
  CAPTION="Legs (describe):"
  CAPTIONALIGN="LEFT">
  <CONTROLREF REFNAME="LEFT" ORDER="1"/>
  <CONTROLREF REFNAME="RIGHT" ORDER="2"/>
</CHECKBOXCONTROL>

```

- 5 「手」用、「足」用チェックボックスコントロールと「その他」のテキストコントロールを含めるために、Controlref を使用するチェックボックスコントロールを定義します。

```

6 <CHECKBOXCONTROL REFNAME="MUSCLES" NAME="MUSCLES"
  LAYOUT="VERTICAL">
  <CONTROLREF REFNAME="ARMS" ORDER="1"/>
  <CONTROLREF REFNAME="LEGS" ORDER="2"/>
  <CONTROLREF REFNAME="OTHER" ORDER="3"/>
</CHECKBOXCONTROL>

```

例 3

以下の例は、チェックボックスコントロールキャプションの翻訳の定義を示しています。

```

<CHECKBOXCONTROL REFNAME="MUSCLES"
  NAME="MUSCLES"
  LANGUAGE="en-US"
  CAPTION="Principal location of pain"
  LAYOUT="VERTICAL">
  <CONTROLREF REFNAME="ARMS" ORDER="1"/>
  <CONTROLREF REFNAME="LEGS" ORDER="2"/>
  <CONTROLREF REFNAME="OTHER" ORDER="3"/>
  <TRANSLATIONS>
    <TRANSLATION NAME="CAPTION" LOCALE="es-ES"
      DISPLAYTEXT="Principal Localizacion del Dolor"/>
    <TRANSLATION NAME="CAPTION" LOCALE="it-IT"
      DISPLAYTEXT="Principali Localita del Dolore"/>
  </TRANSLATIONS>
</CHECKBOXCONTROL>

```

CodeTarget (ディクショナリ)

目的

コード化ディクショナリを定義するエレメントセットの一部。各コード化ディクショナリにはコードターゲットのセットが含まれています。コードターゲットはコード化された後に、データを保持する1つのタイプのアイテムを指定します。

構文

```
< CODETARGET
  NAME="name"/>
```

属性

NAME="name"

コードターゲットの名前。必須

例

```
<DICTIONARY TYPE="WHODD" VERSION="05Q4" CULTURE="en-US">
  <CODETARGET NAME="ATC 1.CODE"/>
  <CODETARGET NAME="ATC 1.TERM"/>
  <CODETARGET NAME="ATC 2.CODE"/>
  <CODETARGET NAME="ATC 2.TERM"/>
  <CODETARGET NAME="ATC 3.CODE"/>
  <CODETARGET NAME="ATC 3.TERM"/>
  <CODETARGET NAME="ATC 4.CODE"/>
  <CODETARGET NAME="ATC 4.TERM"/>
  <CODETARGET NAME="Preferred Name.CODE"/>
  <CODETARGET NAME="Preferred Name.TERM"/>
  <CODETARGET NAME="Ingredients.AddInfo"/>
  <CODETARGET NAME="Trade Name.CODE"/>
  <CODETARGET NAME="Trade Name.TERM"/>
  <CODETARGET NAME="Medicinal Product.CODE"/>
  <CODETARGET NAME="Medicinal Product.TERM"/>
  <CODETARGET NAME="Name Specifier.AddInfo"/>
  <CODETARGET NAME="Country of Sale.AddInfo"/>
  <CODETARGET NAME="MA Holder.AddInfo"/>
  <CODETARGET NAME="MA Holder Country.AddInfo"/>
  <CODETARGET NAME="Company.AddInfo"/>
  <CODETARGET NAME="Company Country.AddInfo"/>
  <CODETARGET NAME="ICH Med Prod ID.AddInfo"/>
  <CODETARGET NAME="Sequence Number 3.AddInfo"/>
  <CODETARGET NAME="Sequence Number 4.AddInfo"/>
  <CODETARGET NAME="MA Number.AddInfo"/>
  <CODETARGET NAME="MA Date.AddInfo"/>
  <CODETARGET NAME="MA Withdrawal Date.AddInfo"/>
  <CODETARGET NAME="Product Type.AddInfo"/>
  <CODETARGET NAME="Product Group.AddInfo"/>
  <CODETARGET NAME="Pharmaceutical Product.AddInfo"/>
```

```

    <CONTEXTITEM NAME="Route Of Administration"/>
    <CONTEXTITEM NAME="Indication"/>
    <VERBATIMTYPE NAME="MEDPROD" />
</DICTIONARY>

```

ContextItem (ディクショナリ)

目的

コード化ディクショナリを定義するエレメントセットの一部。コード化ディクショナリにはコンテキストアイテムのセットを含むことができます。コンテキストアイテムは `verbatim` のコード化を有効にする追加情報を提供します。

構文

```

< CONTEXTITEM
  NAME="name" />

```

属性

NAME="*name*"

以下のようなコンテキストアイテムの名前。

- **投与経路** — 薬剤が投与された経路 (WHO-DD のみ)。
- **適応症** — 薬物の投与が有効であると考えられる病気または障害 (WHO-DD のみ)。

必須

例

```

<DICTIONARY TYPE="WHODD" VERSION="05Q4" CULTURE="en-US">
  <CODETARGET NAME="ATC 1.CODE"/>
  <CODETARGET NAME="ATC 1.TERM"/>
  <CODETARGET NAME="ATC 2.CODE"/>
  <CODETARGET NAME="ATC 2.TERM"/>
  <CODETARGET NAME="ATC 3.CODE"/>
  <CODETARGET NAME="ATC 3.TERM"/>
  <CODETARGET NAME="ATC 4.CODE"/>
  <CODETARGET NAME="ATC 4.TERM"/>
  <CODETARGET NAME="Preferred Name.CODE"/>
  <CODETARGET NAME="Preferred Name.TERM"/>
  <CODETARGET NAME="Ingredients.AddInfo"/>
  <CODETARGET NAME="Trade Name.CODE"/>
  <CODETARGET NAME="Trade Name.TERM"/>
  <CODETARGET NAME="Medicinal Product.CODE"/>
  <CODETARGET NAME="Medicinal Product.TERM"/>
  <CODETARGET NAME="Name Specifier.AddInfo"/>
  <CODETARGET NAME="Country of Sale.AddInfo"/>
  <CODETARGET NAME="MA Holder.AddInfo"/>
  <CODETARGET NAME="MA Holder Country.AddInfo"/>
  <CODETARGET NAME="Company.AddInfo"/>

```

```

<CODETARGET NAME="Company Country.AddInfo"/>
<CODETARGET NAME="ICH Med Prod ID.AddInfo"/>
<CODETARGET NAME="Sequence Number 3.AddInfo"/>
<CODETARGET NAME="Sequence Number 4.AddInfo"/>
<CODETARGET NAME="MA Number.AddInfo"/>
<CODETARGET NAME="MA Date.AddInfo"/>
<CODETARGET NAME="MA Withdrawal Date.AddInfo"/>
<CODETARGET NAME="Product Type.AddInfo"/>
<CODETARGET NAME="Product Group.AddInfo"/>
<CODETARGET NAME="Pharmaceutical Product.AddInfo"/>
<CONTEXTITEM NAME="Route Of Administration"/>
<CONTEXTITEM NAME="Indication"/>
<VERBATIMTYPE NAME="MEDPROD" />
</DICTIONARY>

```

Controlref

目的

あるタイプのコンポーネントの定義を別のコンポーネントの定義に含めます。Controlref は単に Controlref が含まれているコンポーネントのチャイルドコントロールとして定義され、単独のコンポーネントとしてサブミットされません。

Controlref を含めることができるコンポーネントのタイプ:

- CheckBoxControl
- GroupControl
- Item
- RadioControl
- Path

Controlref を使用することによって含めることができるコンポーネントのタイプ:

- CalculatedControl
- CheckBoxControl
- DateTimeControl
- GroupControl
- PullDownControl
- RadioControl
- SimpleControl
- TextControl

構文

```

<CONTROLREF
  REFNAME="name"
  [ORDER="n"]

```

```
[SELECTIONVALUE="text"]/>
```

属性

REFNAME="name"

Controlref によってコンパウンドコントロール (いくつかのコンポーネントを含んでいるコントロール) の定義に含められるコンポーネントの RefName。必須

ORDER="n"

コンパウンドコントロール定義における各 Controlref の順序。オプション。順序を指定しない場合、MedML インストラユーティリティは、Controlref が参照するコンポーネントを入力された順番で順序付けします。

SELECTIONVALUE="text"

Controlref によって含められるコンパウンドコントロールの値。オプション。値を指定しない場合、ユーザがコンパウンドコントロールの中の 1 つのコントロールを選択すると、文字列 lpf と選択されたコントロールの DBUID パスで構成される文字列がデータベースに値として保存されます。たとえば、テキストボックスコントロールに関連付けられた、「その他」というラベルの付いたチェックボックスまたはラジオコントロールコンポーネントの値を入力する際に、この属性が便利です。

注：Controlref または Unitref 定義を用いてコンパウンドコントロールを定義するとき、すべての下位コントロールが同じ **TYPE** 属性で定義され、同じデータタイプを返すようにしてください。

コンパウンドコントロールを定義するとき、InForm アプリケーションは最大 **5** 層のネスティングをサポートしています。**5** 層での設計が可能ですが、パフォーマンスを向上させるためにネストレベルはできるだけ少なくしてください。

Path エlement に **Controlref** 定義を含めるときは、**REFNAME** 属性だけが有効です。

例

この例は、最初のボタンにドロップダウンリストがあり、2 番目のボタンにテキストボックスがあるラジオボタンのリストの作成方法を示しています。ラジオボタンのリストの定義は、Controlref としてそれぞれのリストアイテムの定義を含んでいます。このリストは薬物療法の取得の際、使用されます。

- 1 PFElement のセットとしてドロップダウンリストを定義し、Elementref を用いて、それらを PullDownControl 定義に入れます。
- 2


```
<PFELEMENT REFNAME="QD" LABEL="QD" TYPE="STRING" VALUE="QD"/>
<PFELEMENT REFNAME="BID" LABEL="BID" TYPE="STRING" VALUE="BID"/>
<PFELEMENT REFNAME="TID" LABEL="TID" TYPE="STRING" VALUE="TID"/>
<PFELEMENT REFNAME="QID" LABEL="QID" TYPE="STRING" VALUE="QID"/>
<PULLDOWNCONTROL REFNAME="MEDREGIMEN"
  NAME="MEDREGIMEN">
  <ELEMENTREF REFNAME="QD" ORDER="1"/>
  <ELEMENTREF REFNAME="BID" ORDER="2"/>
  <ELEMENTREF REFNAME="TID" ORDER="3"/>
```



```
<ELEMENTREF REFNAME="QID" ORDER="4"/>
</PULLDOWNCONTROL>
```

3 TextControl としてテキストボックスとそのキャプションを定義します。

```
4 <TEXTCONTROL REFNAME="MEDREGTEXT"
  NAME="MEDREGTEXT"
  HEIGHT="1"
  LENGTH="20"
  MAXLENGTH="20"
  DATATYPE="STRING"
  CAPTION="その他(指定する):"
  CAPTIONALIGN="TOP"/>
```

5 Controlref を使用して、プルダウンコントロールとテキストコントロールを RadioControl に含めます。[その他 (指定する)] ラジオボタンに「OTHER」の値を割り当てるために **SELECTIONVALUE** 属性を使用しています。

```
6 <RADIOCONTROL REFNAME="MEDREGGROUP"
  NAME="MEDREGRADIO" LAYOUT="VERTICAL">
  <CONTROLREF REFNAME="MEDREGIMEN" ORDER="1"/>
  <CONTROLREF REFNAME="MEDREGTEXT" ORDER="2"
    SELECTIONVALUE="OTHER"/>
</RADIOCONTROL>
```

これは、InForm データベースと CDD 間のコンポーネントのマッピング作成において、ソースコントロールを識別するために Controlref 定義が Path エlement の中で使用されている例です。

```
<EXTERNALMAP>
```

```
<PATH>
  <CHAPTERREF REFNAME="PF_ALL_VISITS"/>
  <PAGEREF REFNAME="ECG"/>
  <SECTIONREF REFNAME="CHESTXRAY"/>
  <ITEMSETREF REFNAME="1"/>
  <ITEMREF REFNAME="INTERPRET2"/>
  <CONTROLREF REFNAME="INTERPRETRADIO2"/>
  <CONTROLREF REFNAME="DESCRIBETEXT"/>
</PATH>
```

```
<CDD REFNAME="CDD1" KEYTYPE="PATIENT" TARGETTABLE="t_ECG"
  TARGETKEYTYPE="PATIENTVISIT" TARGETCOLUMN="COMMONDAT1"
  TARGETCOLUMNTYPE="TEXT"/>
```

```
</EXTERNALMAP>
```

DateTimeControl

目的

次に示す日付や時間データを 1 つ以上を含むドロップダウンリストを定義します。
日、月、年、時間、分または秒。

構文

```
<DATETIMECONTROL
  REFNAME="name"
  [DESIGNNOTE="text"]
  [NAME="name"]
  [UUID="id"]
  [LANGUAGE="name"]
  [CAPTION="text"]
  [CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"]
  [ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"]
  STARTYEAR="year"
  ENDYEAR="year"
  [DISPLAYMONTH="true|false"]
  [DISPLAYDAY="true|false"]
  [DISPLAYYEAR="true|false"]
  [DISPLAYHOUR="true|false"]
  [DISPLAYMINUTE="true|false"]
  [DISPLAYSECOND="true|false"]
  [REQUIREMONTH="true|false"]
  [REQUIREDAY="true|false"]
  [REQUIREYEAR="true|false"]
  [REQUIREHOUR="true|false"]
  [REQUIREMINUTE="true|false"]
  [REQUIRESECOND="true|false"]
  [UNKNOWNMONTH="true|false"]
  [UNKNOWNDAY="true|false"]
  [UNKNOWNYEAR="true|false"]
  [UNKNOWNHOUR="true|false"]
  [UNKNOWNMINUTE="true|false"]
  [UNKNOWNSECOND="true|false"]
  [CHECKCONSISTENT="true|false"]
  [TEXTFORMAT="true|false"]>
  <TRANSLATIONS/>
</DATETIMECONTROL>
```

属性

REFNAME="name"

他のコントロール定義で **DateTimeControl** を参照するときに使用される名前。この名前は **DateTimeControl** 定義の中でユニークである必要があります。必須

DESIGNNOTE="text"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

他のコントロール定義で **DateTimeControl** を参照するときに使用される名前。オプション

UUID="id"

Universally Unique Identifier。すべてのスタディ、スタディデータベースおよびマシン間でコンポーネントをユニークに識別する文字列。オプション

注： < 繰り返しフォームセットを作成するとき（予定外のビジットを識別するほとんどの場合）、フォームセットの最初のフォームは以下の **UUID** を含む **DateTimeControl** 定義を含んでいる必要があります。ï

BD991BC0-B0A4-11D2-80E3-00A0C9AF7674

この **DateTimeControl** 定義は、例セクションで説明されています。

LANGUAGE="name"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合（**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合）は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

CAPTION="text"

チェックボックスリストと共に画面に表示されるテキスト。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"

計算値に対する相対的なキャプションの位置: **LEFT**（左）、**RIGHT**（右）、**TOP**（上）、もしくは **BOTTOM**（下）。デフォルトは **LEFT**（左）です。オプション

ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"

コントロール内の計算値の位置調整: **LEFT**（左）、**CENTER**（中央）、**RIGHT**（右）、**TOP**（上）、**MIDDLE**（中央）、もしくは **BOTTOM**（下）。デフォルトは **LEFT**（左）です。オプション

STARTYEAR="year"

年のドロップダウンリストに表示される最初の年。**DateTimeControl** で日を定義する場合には必須です。

ENDYEAR="year"

年のドロップダウンリストに表示される最後の年。DateTimeControl で日を定義する場合には必須です。

DISPLAYMONTH="true|false"

コントロールに「月」用のドロップダウンリストを含めるかどうかを指定します。デフォルトは、True になっています。オプション

DISPLAYDAY="true|false"

コントロールに「日」用のドロップダウンリストを含めるかどうかを指定します。デフォルトは、True になっています。オプション

DISPLAYYEAR="true|false"

コントロールに「年」用のドロップダウンリストを含めるかどうかを指定します。デフォルトは、True になっています。オプション

DISPLAYHOUR="true|false"

コントロールに「時間」用のドロップダウンリストを含めるかどうかを指定します。デフォルトは、False になっています。オプション

DISPLAYMINUTE="true|false"

コントロールに「分」用のドロップダウンリストを含めるかどうかを指定します。デフォルトは、False になっています。オプション

DISPLAYSECOND="true|false"

コントロールに「秒」用のドロップダウンリストを含めるかどうかを指定します。デフォルトは、False になっています。オプション

REQUIREMONTH="true|false"

「月」が入力必須かどうかを示します。この属性は、**DISPLAYMONTH** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

REQUIREDAY="true|false"

「日」が入力必須かどうかを示します。この属性は、**DISPLAYDAY** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

REQUIREYEAR="true|false"

「年」が入力必須かどうかを示します。この属性は、**DISPLAYYEAR** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

REQUIREHOUR="true|false"

「月」が入力必須かどうかを示します。この属性は、**DISPLAYHOUR** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

REQUIREMINUTE="true|false"

「月」が入力必須かどうかを示します。この属性は、**DISPLAYMINUTE** 属性に true を

指定する場合にのみ有効です。デフォルトは、False になっています。オプション

REQUIRESECOND="true|false"

「月」が入力必須かどうかを示します。この属性は、**DISPLAYSECOND** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

UNKNOWNMONTH="true|false"

ユーザが「月」を不明として入力できるかどうかを示します。true の場合、ドロップダウンリストに「UNK」が含まれます。この属性は、**DISPLAYMONTH** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

UNKNOWNDAY="true|false"

ユーザが「日」を不明として入力できるかどうかを示します。true の場合、ドロップダウンリストに「UNK」が含まれます。この属性は、**DISPLAYDAY** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

UNKNOWNYEAR="true|false"

ユーザが「年」を不明として入力できるかどうかを示します。true の場合、ドロップダウンリストに「UNK」が含まれます。この属性は、**DISPLAYYEAR** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

UNKNOWNHOUR="true|false"

ユーザが「時」を不明として入力できるかどうかを示します。true の場合、ドロップダウンリストに「UNK」が含まれます。この属性は、**DISPLAYHOUR** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

UNKNOWNMINUTE="true|false"

ユーザが「時」を不明として入力できるかどうかを示します。true の場合、ドロップダウンリストに「UNK」が含まれます。この属性は、**DISPLAYMINUTE** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

UNKNOWNSECOND="true|false"

ユーザが「時」を不明として入力できるかどうかを示します。true の場合、ドロップダウンリストに「UNK」が含まれます。この属性は、**DISPLAYSECOND** 属性に true を指定する場合にのみ有効です。デフォルトは、False になっています。オプション

CHECKCONSISTENT="true|false"

日時コンポーネントに入力された内容の整合性を InForm アプリケーションがチェックするかどうか指定します。デフォルトは、True になっています。オプション

CHECKCONSISTENT の値が true の場合:

- 日付または時間コンポーネントの値が入力されると、コントロールに表示されるすべての上位コンポーネントにも数値が入力される必要があります。たとえば、月/日/年の日時コントロールに日の値を入力する場合は、月と年の入力も必要になります。日時コントロールが日付と時刻コンポーネントの両方を含んでいるときに時間の値を入力する場合は、日付部分の入力も必要になります。
- InForm アプリケーションは空白のコントロールと UNK 値を持つコントロールを

同様に処理します。たとえば、月に対しては数値、年に対しては UNK をそれぞれ入力すると、エントリはエラーを生成します。

- 一貫性チェックはオプションと必須の日時コンポーネントに適用されます。たとえば、コントロールの時刻部分がオプションの場合、時間を入力せずに分を入力すると、InForm ソフトウェアはエラーを生成します。
- 日時コントロールの年コンポーネントが表示されない場合は、一貫性チェックが無効となっています。

TEXTFORMAT="true|false"

コントロールの時刻部分を一連のデフォルトのドロップダウンリストではなく、テキストボックスのセットとして表示するかどうかを指定します。デフォルトは、False になっています。オプション

子

スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エlement には 1 つ以上の Translation 定義を含める必要があります。Translation エlement はそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。DateTimeControl 定義では、**CAPTION** 属性を翻訳できます。

例 1

以下の例は、症例の生年月日のドロップダウンリストを作成する DateTimeControl 定義を示しています。月、日、年は必須です。コントロールは Controlref コンポーネントを使用することにより、Item 定義に含まれます。また、この例はアイテムから CRF ヘルプエントリへのリンクを指定する HelpLink コンポーネントを含んでいます。

```
<DATETIMECONTROL REFNAME="dob" NAME="dob"
  UUID="40aee712-217c-11d2-a425-00a0c963e0ac"
  STARTYEAR="1932"
  ENDYEAR="1998"
  DISPLAYDAY="true" DISPLAYMONTH="true" DISPLAYYEAR="true"
  REQUIREMONTH="true" REQUIREDAY="true" REQUIREYEAR="true"/>

<ITEM REFNAME="DEMDOB" QUESTION="Date of Birth:">
  <CONTROLREF REFNAME="dob"/>
  <HELPLINK DOCREFNAME="Study"
    BODYREFNAME="DEMHELP"
    BOOKMARK="Item2"/>
</ITEM>
```

例 2

この例は、すべてのビジットの最初のフォームと、繰り返しフォームセットに必要な「ビジット日時 (Date of Visit)」セクションの定義を示しています。このセクション定義は、必要な UUID を持つ DateTimeControl を使用しています。コントロールには、ビジットの月、日、年、時、および分が含まれています。月と年は必須です。

```
<DATETIMECONTROL REFNAME="DOV" NAME="DOV"
```

```

    UUID="BD991BC0-B0A4-11D2-80E3- 00A0C9AF7674"
    STARTYEAR="1997" ENDYEAR="2004"
    DISPLAYDAY="true" DISPLAYMONTH="true"
    DISPLAYYEAR="true" DISPLAYHOUR="true"
    DISPLAYMINUTE="true"
    REQUIREMONTH="true" REQUIREYEAR="true"/>

<ITEM REFNAME="DOV" QUESTION="Date and time of visit:"
    UUID="BD991BBF-B0A4-11D2-80E3-00A0C9AF7674">
    <CONTROLREF REFNAME="DOV"/>
</ITEM>

<SECTION REFNAME="DOV" TITLE="Date Of Visit"
    UUID="BD991BBE-B0A4-11D2-80E3-00A0C9AF7674">
    <ITEMREF REFNAME="DOV" ORDER="1"/>
</SECTION>

```

例 3

以下の例は、DateTimeControl 定義におけるキャプションの翻訳を示しています。

```

<DATETIMECONTROL REFNAME="startdate" NAME="startdate"
    LANGUAGE="en-US"
    STARTYEAR="1990"
    ENDYEAR="2010"
    CAPTION="Enter Month/Day/Year"
    DISPLAYDAY="true" DISPLAYMONTH="true" DISPLAYYEAR="true"
    REQUIREMONTH="true" REQUIREDAY="true" REQUIREYEAR="true">
    <TRANSLATIONS>
        <TRANSLATION NAME="CAPTION" LOCALE="fr-FR"
            DISPLAYTEXT="Entrez Jour / Mois / Annee"/>
    </TRANSLATIONS>
</DATETIMECONTROL>

```

Dictionary (定義)

目的

コード化ディクショナリを定義します。

構文

```
<DICTIONARY  
  TYPE="text"  
  VERSION="text"  
  CULTURE="text">  
  <CODETARGET+ attributes/>  
  <CONTEXTITEM* attributes/>  
  <VERBATIMTYPE* attributes/>  
</DICTIONARY>
```

属性

TYPE="text"

ディクショナリのタイプ (MedDRA、WHO-DD など)。必須

VERSION="text"

ディクショナリのバージョン (8.1、05Q4 など)。必須

CULTURE="text"

言語とカルチャ (en-JP など)。必須

注: **TYPE**、**VERSION**、および **CULTURE** の値の組み合わせでディクショナリを一意に識別します。

子

- ディクショナリに定義された特定のコードターゲットを識別する 1 つ以上の CodeTarget エlement。
- ディクショナリに定義された特定のコンテキストアイテムを識別する 0 以上の ContextItem エlement。
- ディクショナリの有効な verbatim タイプを識別する 0 以上の VerbatimType エlement。verbatim タイプは Central Coding アイテムタイプに対応します。

制限

- コードターゲット名、コンテキストアイテム名、および verbatim タイプ名はディクショナリ内で一意である必要があります。
- 少なくとも 1 つのコードターゲットを指定する必要があります。

例

```
<DICTIONARY TYPE="WHODD" VERSION="05Q4" CULTURE="en-US">
  <CODETARGET NAME="ATC 1.CODE"/>
  <CODETARGET NAME="ATC 1.TERM"/>
  <CODETARGET NAME="ATC 2.CODE"/>
  <CODETARGET NAME="ATC 2.TERM"/>
  <CODETARGET NAME="ATC 3.CODE"/>
  <CODETARGET NAME="ATC 3.TERM"/>
  <CODETARGET NAME="ATC 4.CODE"/>
  <CODETARGET NAME="ATC 4.TERM"/>
  <CODETARGET NAME="Preferred Name.CODE"/>
  <CODETARGET NAME="Preferred Name.TERM"/>
  <CODETARGET NAME="Ingredients.AddInfo"/>
  <CODETARGET NAME="Trade Name.CODE"/>
  <CODETARGET NAME="Trade Name.TERM"/>
  <CODETARGET NAME="Medicinal Product.CODE"/>
  <CODETARGET NAME="Medicinal Product.TERM"/>
  <CODETARGET NAME="Name Specifier.AddInfo"/>
  <CODETARGET NAME="Country of Sale.AddInfo"/>
  <CODETARGET NAME="MA Holder.AddInfo"/>
  <CODETARGET NAME="MA Holder Country.AddInfo"/>
  <CODETARGET NAME="Company.AddInfo"/>
  <CODETARGET NAME="Company Country.AddInfo"/>
  <CODETARGET NAME="ICH Med Prod ID.AddInfo"/>
  <CODETARGET NAME="Sequence Number 3.AddInfo"/>
  <CODETARGET NAME="Sequence Number 4.AddInfo"/>
  <CODETARGET NAME="MA Number.AddInfo"/>
  <CODETARGET NAME="MA Date.AddInfo"/>
  <CODETARGET NAME="MA Withdrawal Date.AddInfo"/>
  <CODETARGET NAME="Product Type.AddInfo"/>
  <CODETARGET NAME="Product Group.AddInfo"/>
  <CODETARGET NAME="Pharmaceutical Product.AddInfo"/>
  <CONTEXTITEM NAME="Route Of Administration"/>
  <CONTEXTITEM NAME="Indication"/>
  <VERBATIMTYPE NAME="MEDPROD" />
```

</DICTIONARY>

DocBody

目的

主としてスタディプロトコルのセクション、CRF アイテムヘルプ、オンラインヘルプ トピックを 1 つの HTM ファイルで定義します。DocBody エレメントは Documentation 定義に含めることで定義されます。単独で成り立ちません。

構文

```
<DOCBODY
  REFNAME="name"
  FILENAME="name"
  FILENAMEID="id"
  [ORDER="n"]
  [LINKS="n"]
  [LANGUAGE="name"]>
  <TRANSLATIONS/>
</DOCBODY>
```

属性

REFNAME="name"

リンク指定にある DocBody ファイルを参照するための RefName。必須

FILENAME="name"

DocBody で定義する HTM ファイルのパスとファイル名。**TRANSLATIONS** 属性を使用すると、翻訳済みのものを含んだファイルのパスとファイル名を指定できます。DocBody エレメントによって表される翻訳済みバージョンのコンテンツを含むファイルのパスとファイル名を指定できます。

注： パスは **MedML** インストーラユーティリティを実行する場所からの相対パスにする必要があります。

FILENAMEID="id"

<同期>処理のみが使用する識別コード。

ORDER="n"

ユーザが、ドキュメントまたはヘルプウィンドウで、ページコントロールをクリックしながら順番に読み進めているときに、DocBody ファイルが表示される順番を示す数。デフォルトは Documentation 定義内で DocBody が出現する順番です。オプション

LINKS="n"

DocBody ファイルから他の DocBody ファイルへのリンク件数。この件数には、同じファイル内のブックマークへのリンクは含まれません。DocBody ファイルが他の DocBody ファイルへのリンクを含んでいる場合のみ必要です。

LANGUAGE="name"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない

場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合（**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合）は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

子

翻訳済みファイルセットを指定する 0 以上の Translations 定義。Translation エレメントには 1 つ以上の Translation 定義を含める必要があります。各 Translation エレメントには 1 つのロケールの言語に翻訳されたファイルのパスとファイル名が含まれています。DocBody 定義では、主としてスタディプロトコルのセクション、CRF アイテムヘルプ、オンラインヘルプトピックを表す翻訳済み HTM ファイルのパスとファイル名を含めることができます。

例

以下の例は、オンラインヘルプの 1 つのセクションにトピックをロードするための DocBody エレメントの用例を示しています。さらに、翻訳済みバージョンのトピックのロード方法も示しています。

```
<DOCUMENTATION REFNAME="AboutInForm"
  DOCNAME="About InForm software"
  DOCTYPE="HELP"
  HELPTEXT="Click here for an introduction to InForm software"
  TOC="..\XMLBase\Help\TocAboutInform.htm"
  TOCLINKS="6"
  INDEX="..\XMLBase\Help\HelpIndex.htm"
  INDEXLINKS="220">
  <DOCBODY REFNAME="WELCOME" FILENAME="..\XMLBase\Help\Welcome.htm"
    LINKS="5"/>
    <TRANSLATIONS>
      <TRANSLATION NAME="FILENAME" LOCALE="jp-JP"
        DISPLAYTEXT="..\XMLBase\Help\Welcome_jp-JP.htm"/>
      </TRANSLATIONS>
    <DOCBODY REFNAME="FEATURES" FILENAME="..\XMLBase\Help\InFormFeatures.htm"
      LINKS="4"/>
      <TRANSLATIONS>
        <TRANSLATION NAME="FILENAME" LOCALE="jp-JP"
          DISPLAYTEXT="..\XMLBase\Help\InFormFeatures_jp-JP.htm"/>
        </TRANSLATIONS>
      <DOCBODY REFNAME="ROADMAP" FILENAME="..\XMLBase\Help\RoadMap.htm"
        LINKS="5"/>
        <TRANSLATIONS>
          <TRANSLATION NAME="FILENAME" LOCALE="jp-JP"
            DISPLAYTEXT="..\XMLBase\Help\RoadMap_jp-JP.htm"/>
          </TRANSLATIONS>
      <DOCBODY REFNAME="RIGHTS" FILENAME="..\XMLBase\Help\Rights.htm" LINKS="5"/>
      <TRANSLATIONS>
        <TRANSLATION NAME="FILENAME" LOCALE="jp-JP"
          DISPLAYTEXT="..\XMLBase\Help\Rights_jp-JP.htm"/>
        </TRANSLATIONS>
      <DOCBODY REFNAME="DATETIME" FILENAME="..\XMLBase\Help\DateTime.htm"
        LINKS="4"/>
      <TRANSLATIONS>
        <TRANSLATION NAME="FILENAME" LOCALE="jp-JP"
          DISPLAYTEXT="..\XMLBase\Help\DateTime_jp-JP.htm"/>
        </TRANSLATIONS>
    </DOCBODY>
  </DOCUMENTATION>
```

Documentation

目的

ドキュメントウィンドウまたはヘルプウィンドウに表示するドキュメントを定義します。それぞれのドキュメントは、ドキュメントウィンドウまたはヘルプウィンドウで 1 つのタブの中に表示されます。1 つのドキュメントに、DocBody エlementで定義された複数のドキュメントを含めることができます。InForm アプリケーションでは、以下のものが Documentation Elementによって表示されています。

- スタディプロトコル
- CRF ヘルプ
- スポンサードキュメント
- ビジットカルキュレイター
- サンプル CRB

ドキュメントをドキュメントウィンドウやヘルプウィンドウで参照できるように、Documentation 定義は StudyVersionDoc Elementの StudyVersion に関連付ける必要があります。

構文

```
<DOCUMENTATION
  REFNAME="name"
  DOCNAME="name"
  DOCTYPE="BOOKMARKDOC|BOOKMARKFAQDOC|VISITDOC|CRBDOC|HELP"
  [HELPTXT="text"]
  [UUID="id"]
  [LANGUAGE="name"]
  TOC="filename"
  [TOCLINKS="n"]
  [INDEX="filename"]
  [INDEXLINKS="n"]>
  <DOCBODY+ attributes/>
  <TRANSLATIONS/>
</DOCUMENTATION>
```

属性

REFNAME="name"

StudyVersionDoc 定義でドキュメントを参照するときに使用される名前。必須

DOCNAME="name"

ドキュメントウィンドウまたはヘルプウィンドウでドキュメントを表すエレメントに表示する名前。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。必須

DOCTYPE="BOOKMARKDOC|BOOKMARKFAQDOC|VISITDOC|CRBDOC|HELP"

ドキュメントのタイプ。必須

- **BOOKMARKDOC**—他のドキュメントからリンクできるドキュメント。スタディプロトコルに利用されます。
- **BOOKMARKFAQDOC**—他のドキュメントからリンクできるドキュメント。CRF ヘルプで使用されます。
- **VISITDOC** — ビジットカルキュレイター画面。
- **CRBDOC** — スタディのための空白の CRF フォームサンプルセット。

HELPTTEXT="text"

ドキュメントウィンドウやヘルプウィンドウのドキュメントのタブの上にカーソルを動かすと表示されるフローティングヘルプのテキスト。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

UUID="id"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション。

LANGUAGE="name"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

TOC="filename"

ドキュメントの目次として使用される HTM ファイルのパスとファイル名。

BOOKMARKDOC、**BOOKMARKFAQDOC**、および **HELP** のドキュメントタイプには必須です。**TRANSLATIONS** 属性を使用すると、翻訳済みのものを含んだファイルのパスとファイル名を指定できます。翻訳済みの目次を含むファイルのパスとファイル名を指定できます。

注： パスは **MedML** インストーラユーティリティを実行する場所からの相対パスにする必要があります。

TOCLINKS="n"

ドキュメントの目次ファイルから他のファイルへのリンクの件数。

INDEX="filename"

ドキュメントのインデックスとして利用される HTM ファイルのパスおよびファイル名。ドキュメントタイプが **HELP** の場合、必要です。他では不要です。**TRANSLATIONS** 属性を使用すると、翻訳済みのものを含んだファイルのパスとファイル名を指定できます。インデックス。

注： パスは **MedML** インストーラユーティリティを実行する場所からの相対パスにする必要があります。

INDEXLINKS="n"

ドキュメントのインデックスファイルから他のファイルへのリンクの件数。

子

- 0 以上の DocBody エレメント。各 DocBody エレメントは、1 つの HTM ファイルを定義しています (この HTM ファイルは一般的にスタディプロトコルのセクション、CRF の CRF アイテムヘルプのセット、またはオンラインシステムヘルプトピック)。**VISITDOC** や **CRBDOC** タイプを含んだドキュメント定義は DocBody エレメントを含みません。
- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エレメントには 1 つ以上の Translation 定義を含める必要があります。Translation エレメントはそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エレメントは、スタディコンポーネント定義内の最後のチャイルドエレメントにする必要があります。Documentation 定義では、**DOCNAME** と **HELPTEXT** 属性を翻訳し、翻訳された目次と索引のパスとファイル名を指定できます。属性を翻訳する場合は、スタディコンポーネント定義の一部であるすべての翻訳可能な属性の翻訳定義を含めます。

例

以下のドキュメント定義セットはそれぞれのドキュメントタイプを説明しています。プロトコル、スタディ、およびヘルプドキュメントには、DocBody エレメントで定義される複数の HTM ファイルが含まれています。プロトコルとスタディドキュメントには目次があります。ヘルプドキュメントには目次と索引の両方があります。Documentation

定義では、DOCNAME、HELPTXT、TOC、および INDEX 属性を翻訳できます。

```
<DOCUMENTATION REFNAME="Protocol"
  DOCNAME="Protocol Guide"
  DOCTYPE="BOOKMARKDOC"
  LANGUAGE="en-US"
  HELPTXT="Click here to view Protocol Guide"
  TOC=".\\StudyDoc\\TOC.htm"
  TOCLINKS="25">
  <DOCBODY REFNAME= "OBJECT" FILENAME=".\\StudyDoc\\Objectives.htm" LINKS="0"/>
  <DOCBODY REFNAME= "DESIGN" FILENAME=".\\StudyDoc\\StudyDesign.htm" LINKS="9"/>
  <DOCBODY REFNAME= "DROPOUT" FILENAME=".\\StudyDoc\\DropoutDrug.htm"
  LINKS="10"/>
  <DOCBODY REFNAME= "INCLEXCL" FILENAME=".\\StudyDoc\\InclExclCriteria.htm"
  LINKS="3"/>
  <DOCBODY REFNAME= "TESCHEDULE" FILENAME=".\\StudyDoc\\TESchedule.htm"
  LINKS="0"/>
  <DOCBODY REFNAME= "MEDRESTRICT" FILENAME=".\\StudyDoc\\MedRestrict.htm"
  LINKS="1"/>
  <DOCBODY REFNAME= "PLACEBO" FILENAME=".\\StudyDoc\\PlaceboRunIn.htm"
  LINKS="2"/>
  <DOCBODY REFNAME= "WEEK-4" FILENAME=".\\StudyDoc\\Week-4.htm" LINKS="1"/>
  <DOCBODY REFNAME= "WEEK-2" FILENAME=".\\StudyDoc\\Week-2.htm" LINKS="2"/>
  <DOCBODY REFNAME= "WEEK0" FILENAME=".\\StudyDoc\\Week0.htm" LINKS="7"/>
  <DOCBODY REFNAME= "TREATMENT" FILENAME=".\\StudyDoc\\TreatmentPhase.htm"
  LINKS="5"/>
  <DOCBODY REFNAME= "WEEK1" FILENAME=".\\StudyDoc\\Week1.htm" LINKS="1"/>
  <DOCBODY REFNAME= "WEEK2" FILENAME=".\\StudyDoc\\Week2.htm" LINKS="2"/>
  <DOCBODY REFNAME= "WEEK4" FILENAME=".\\StudyDoc\\Week4.htm" LINKS="2"/>
  <DOCBODY REFNAME= "WEEK5" FILENAME=".\\StudyDoc\\Week5.htm" LINKS="5"/>
  <DOCBODY REFNAME= "WEEK8" FILENAME=".\\StudyDoc\\Week8.htm" LINKS="1"/>
</DOCUMENTATION>

<DOCUMENTATION REFNAME="Study"
  DOCNAME="CRF Help"
  DOCTYPE="BOOKMARKFAQDOC"
  LANGUAGE="en-US"
  HELPTXT="Click here to view information on completing CRFs"
  TOC=".\\StudyGuide\\SG_TOC.htm"
  TOCLINKS="10">
  <DOCBODY REFNAME="SCREENHELP" FILENAME=".\\StudyGuide\\ScreeningLog.htm"/>
  <DOCBODY REFNAME="ELIGHELP" FILENAME=".\\StudyGuide\\Eligibility.htm"/>
  <DOCBODY REFNAME="SSHELP" FILENAME=".\\StudyGuide\\Signs_Symptoms.htm"/>
  <DOCBODY REFNAME="DEMHELP" FILENAME=".\\StudyGuide\\Demographics.htm"/>
  <DOCBODY REFNAME="VSHelp" FILENAME=".\\StudyGuide\\VitalSigns_BP.htm"/>
  <DOCBODY REFNAME="CHESTHELP" FILENAME=".\\StudyGuide\\ECG_Chest_XRay.htm"/>
  <DOCBODY REFNAME="PEHELP" FILENAME=".\\StudyGuide\\Physical_Exam.htm"/>
  <DOCBODY REFNAME="PEW8HELP"
  FILENAME=".\\StudyGuide\\Week8 Physical Exam.htm"/>
  <DOCBODY REFNAME="CMHELP" FILENAME=".\\StudyGuide\\Con_Meds.htm"/>
  <DOCBODY REFNAME="SCHELP" FILENAME=".\\StudyGuide\\Study_Completion.htm"/>
</DOCUMENTATION>

<DOCUMENTATION REFNAME="Visit"
  DOCNAME="Visit Calculator"
  DOCTYPE="VISITDOC"
  LANGUAGE="en-US"
  HELPTXT="Click here to view Patient Visit Calculator"
</DOCUMENTATION>

<DOCUMENTATION REFNAME="CRB"
  DOCNAME="Sample Book"
  DOCTYPE="CRBDOC"
  LANGUAGE="en-US"
  HELPTXT="Click here to view Sample Case Book forms"
</DOCUMENTATION>
```

Elementref

目的

以下のタイプのフォームコンポーネントの定義に PFElement の定義を含めます。

- PullDownControl
- SimpleControl

Elementref は、これらのコンポーネントのうちの 1 つのチャイルドとしてのみ定義されます。単独コンポーネントとして使用できません。

構文

```
<ELEMENTREF
  REFNAME="name"
  [ORDER="n"]/>
```

属性

REFNAME="*name*"

PullDownControl または SimpleControl に含まれる際の PFElement の RefName。必須

ORDER="*n*"

各 PFElement が PullDownControl または SimpleControl に表示される順序。順序を指定しない場合、MedML インストラユーティリティは PFElement を入力された順に並べます。オプション

例

この例は前もって作成された 3 つの PFElement を取り入れるための Elementref の使い方について示しています。PFElement は「FRAME」という PullDownControl において、症例の体格の選択肢を定義しています。この例では、リストは表示される順序で並べられています。リストを並べ替える場合は、Elementref 定義の Order 属性を変更します。

- 1 PFElement としてリストアイテムを作成します。各コンポーネントのテキストラベルは文字列であるが、コンポーネントは整数値として定義され、データベースには割り当てられた整数値が入ります。
- 2 <PFELEMENT REFNAME="SMALL" LABEL="Small" TYPE="INTEGER" VALUE="1"/>
<PFELEMENT REFNAME="MEDIUM" LABEL="Medium" TYPE="INTEGER" VALUE="2"/>
<PFELEMENT REFNAME="LARGE" LABEL="Large" TYPE="INTEGER" VALUE="3"/>
- 3 Elementrefs の使用によって、FRAME PullDownControl 定義に PFElement 定義を含めます。
- 4 <PULLDOWNCONTROL REFNAME="FRAMEPULLDOWN" NAME="FRAME">


```

<ELEMENTREF REFNAME="SMALL" ORDER="1"/>
<ELEMENTREF REFNAME="MEDIUM" ORDER="2"/>
<ELEMENTREF REFNAME="LARGE" ORDER="3"/>
</PULLDOWNCONTROL>

```

Event

目的

ルールが実行される際に実行するイベントを定義します。イベントの定義には、実行計画への参照、関連付けられているルールを満たしていない場合に作成されるクエリのテキスト、またはそのクエリへの回答が関連付けられているルールを満たす場合に閉じられるクエリのテキストを含めることができます。

構文

```

<EVENT
  REFNAME="name"
  [DESIGNNOTE="text"]
  [UUID="id"]>
  <ACTION* attributes/>
  <EXECUTIONPLANREF* attributes/>
</EVENT>

```

属性

REFNAME="*name*"

ルールの定義でイベントを参照するときに使用される名前。この名前はイベント間でユニークである必要があります。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。必須

子

イベントの定義には 0 以上の Action エlement を指定できます。各々の Action Element は、イベントが発生した際に生成されるクエリを定義します。さらに、イベントの定義には 0 以上の ExecutionPlanref Element を指定できます。各 ExecutionPlanref は定義済み実行計画を参照します。イベントが実行されると、すべての実行計画が実行されます。

例

以下の例は、クエリテキストを「範囲外の値 (Value Out of Range)」とし、未解決状態のクエリを生成するイベントを示しています。イベントが実行されるときに EPScript と

呼ばれる実行計画もまた実行されます。

```
<EVENT REFNAME="Value Out of Range">
  <ACTION NAME="PF_Event_Action"
    VALUE="Query"
    QUERYTYPE="Open"
    QUERYTEXT="Value Out of Range"/>
  <EXECUTIONPLANREF REFNAME="EPScript"/>
</EVENT>
```

Eventref

目的

ルールの定義にイベントの定義を含めます。Eventref は Rule のチャイルドとしてのみ定義されます。単独のコンポーネントとしてサブミットされません。

構文

```
<EVENTREF
  REFNAME="name" />
```

属性

REFNAME="*name*"

Eventref によって Rule 定義に含められるイベントの RefName。必須

例

以下の例で、Eventref タグは、「Height Numeric Rule」という Rule 定義に「Value Out of Range」という Event 定義を含めています。

```
<RULE REFNAME="Height Numeric Rule"
  DESCRIPTION="Height Numeric Rule Description"
  ENABLED="true"
  SCRIPTTYPE="SERVERRULE"
  SCRIPTFILE="height.vbs"
  HELPTEXT="Height must have a numeric value between 30 AND 75.">
  <EVENTREF REFNAME="Value Out of Range"/>
</RULE>
```

ExecutionPlan

目的

以下の処理を実行できるスクリプトを定義します。

- E-メール送信
- Windows ログファイルへのログエントリ

構文

```
<EXECUTIONPLAN
  REFNAME="name"
  [DESIGNNOTE="text"]
  SCRIPTTEXT="text"
  SCRIPTFILE="filename" />
```

属性

REFNAME="*name*"

イベントの定義で ExecutionPlan を参照するときに使用される名前。この名前は ExecutionPlans の中でユニークである必要があります。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

SCRIPTTEXT="*text*"

スクリプトのテキスト。Visual Basic スクリプトのみがサポートされます。**SCRIPTTEXT** 属性または **SCRIPTFILE** 属性のいずれかが必要です。

SCRIPTFILE="*filename*"

スクリプトのテキストを含んでいるファイルの名前。Visual Basic スクリプトのみがサポートされます。**SCRIPTTEXT** 属性または **SCRIPTFILE** 属性のいずれかが必要です。

例

この例は、実行計画を含むイベントが発生するときに Windows ログにメッセージを記録する実行計画を示しています。

```
<EXECUTIONPLAN REFNAME="EPScript"
  SCRIPTTEXT="Global.LogMessage(&quot;Query generated&quot;)" />
```

ExecutionPlanref

目的

イベントの定義に実行計画の定義を含めます。ExecutionPlanref は Event のチャイルドと

してのみ定義されます。単独のコンポーネントとしてサブミットされません。

構文

```
<EXECUTIONPLANREF
  REFNAME="name" />
```

属性

REFNAME="*name*"

ExecutionPlanref によって、Event 定義に含められる実行計画の RefName。必須

例

この例では、ExecutionPlanref エlementによって、「Value Out of Range」イベントの定義に「EPScript」実行計画の定義が含まれています。

```
<EXECUTIONPLAN REFNAME="EPScript"
  SCRIPT*TYPE="SERVERRULE"
  SCRIPT*TEXT="Global.LogMessage("Query generated")"/>
<EVENT REFNAME="Value Out of Range">
  <ACTION NAME="PF_Event_Action"
    VALUE="Query" QUERYTYPE="Open"
    QUERY*TEXT="Value Out of Range"/>
  <EXECUTIONPLANREF REFNAME="EPScript"/>
</EVENT>
```

Formref

目的

定義済みフォームをフォームセットの定義に含めます。Formref はビジットのチャイルドとしてのみ定義されます。単独のコンポーネントとしてサブミットされません。

構文

```
<FORMREF
  REFNAME="name"
  [DYNAMIC="true|false"]
  [ORDER="n"]
  [ALTREFNAME="name"]
  [VISITREFNAME="name"/>
```

属性

REFNAME="*name*"

Formref によってフォームセットの定義に含められるフォームの RefName。必須

DYNAMIC="true|false"

別のフォーム中のデータアイテムの値に基づき、ダイナミックにビジットにフォームが

割り当てられるかどうかを示します。オプションは true または false です。デフォルトは、False になっています。

ORDER="*n*"

FormSet 定義における各フォームの順序。順序を指定しない場合、MedML インストラユーティリティは、Formref が参照するフォームを入力された順番で順序付けします。オプション

ALTREFNAME="*name*"

スタディバージョンが変わる場合に、スタディに参加している症例に提示される古いフォームの新バージョンを特定します。この属性は、特殊な目的を持っているフォームの中でのみ使用してください。このようなフォームの使用例としては、家族歴の一部といった、特定のビジットに関する、時間依存でないデータの追加コンポーネントの集合があります。オプション

VISITREFNAME="*name*"

フォーム間の関連を適用するフォームセットの RefName。FormRef が、Relationship タイプのフォームセット定義に含まれている場合、この属性を使用してください。

例

以下の例は、FormSet エlement第-4 週ビジットを構成するフォームセットに、症例背景 (DEM) や高血圧既往歴 (HH) フォームをどのようにグループ化しているかを示しています。詳細は、「**FormSet**」 (139ページ) を参照してください。

```
<FORMSET REFNAME="Visit1" TITLE="Week -4" TYPE="Visit"
  SCHEDULED="true" ORDER="1">
  <FORMREF REFNAME="DEM"/>
  <FORMREF REFNAME="HH"/>
  <FORMREF REFNAME="DEM" ORDER="3" ALTREFNAME="SMOKE"/>
  <FORMREF REFNAME="Preg" ORDER="4" DYNAMIC="true"/>
</FORMSET>
```

以下の例は、AE と CM のフォーム間の関連を設定するために Formref Elementを使用している例です。

```
<FORMSET REFNAME="AE_CM_RELATION" TYPE="Relationship"
  <FORMREF REFNAME="AE" VISITREFNAME="Visit2"/>
  <FORMREF REFNAME="CM" VISITREFNAME="Visit2"/>
</FORMSET>
```

Form

目的

フォームの定義を作成します。フォーム は Sectionref Elementを含む、事前に定義されたセクションのグループから成ります。

構文

```
<FORM
```

```
REFNAME="name"
[DESIGNNOTE="text"]
[UUID="id"]
TITLE="text"
MNEMONIC="name"
[NOTE="text"]
[LANGUAGE="name"]
[TYPE="CRF|ENROLLMENT|REGDOC|VISITREPORT|CUSTOM|CUSTOMTRIAL|CUSTOMRULES|
      CUSTOMADMIN|CUSTOMAUTH"]
[QUESTIONWIDTH="n"]
[CONTROLWIDTH="n"]
[REPEATING="true|false"]
[UNIQUEKEY="true|false"]>
  <SECTIONREF+ attributes/>
  <HELPLINK attributes/>
  <KEYITEMREF attributes/>
  <TRANSLATIONS/>
</FORM>
```

属性

REFNAME="name"

フォームセットの定義でフォームを参照するときに使用される名前。この名前はすべてのフォームの中で一意でなければなりません。必須

DESIGNNOTE="text"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

UUID="id"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。必須

TITLE="text"

画面上、フォームの先頭に表示されるフォームタイトルのテキストです。

TRANSLATIONS 属性を使用して、属性値を翻訳することができます。必須

MNEMONIC="name"

フォームタイトルの略称。ユーザがフォームへ移動する際にクリックするタブコントロールの上に表示されます。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。必須

NOTE="text"

フォームタイトルのすぐ下に表示されるノート (注意書きなど) のテキストです。

TRANSLATIONS 属性を使用して、属性値を翻訳することができます。オプション

LANGUAGE="name"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エlementがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

TYPE="CRF|ENROLLMENT|REGDOC|VISITREPORT|CUSTOM|CUSTOMTRIAL|CUSTOMRULES|CUSTOMADMIN|CUSTOMAUTH"

フォームタイプ。オプションは以下の通りです。

- **CRF** — 症例報告書フォーム。CRF が同じビジットの中で複数のインスタンスを持つことができる場合、REPEATING 属性を使用してください。
- **ENROLLMENT** — スクリーニングフォームまたは症例登録フォーム。
- **REGDOC** — 規制ドキュメントページで使用されるフォーム。
- **VISITREPORT** — ビジットレポートページで使用されるフォーム。
- **CUSTOM** — 未使用。
- **CUSTOMTRIAL** InForm 機能に必要なページで使用されるフォーム。
- **CUSTOMRULES** — ルール管理ページで使用されるフォーム。
- **CUSTOMADMIN** — ユーザ、施設、システム構成などの管理ページで使用されるフォーム。
- **CUSTOMAUTH** — パスワード管理ページで使用されるフォーム。

デフォルトは CRF です。オプション

QUESTIONWIDTH="n"

質問項目カラムによって占められるフォームの割合。デフォルトは 50 です。
オプション

CONTROLWIDTH="n"

コントロールカラムによって占められるフォームの割合。デフォルトは 50 です。
オプション

REPEATING="true|false"

フォームを繰り返すかどうかを示します。繰り返しフォームでは、同じビジット内に多数のインスタンスを作成できます。また、繰り返しフォームは関連性を持たせることができます。オプションは true または false です。デフォルトは、False になっています。
オプション

UNIQUEKEY="true|false"

フォームに 1 つ以上の複合キー (スタディ設計でグループの一意性によって定義されたアイテム) が含まれているかどうかを示します。このプロパティは、繰り返しフォームのみに適用されます。デフォルトは、False になっています。オプション

フォームに複合キーが含まれており、そのフォームから 2 つインスタンスが同じ複合キーアイテムで同じビジットにサブミットされた場合、2 番目のインスタンスは受け付けられません。アイテムセット内のアイテムは、繰り返しフォームのキーアイテムになることができません。

True を指定した場合

- キーアイテムのすべての組み合わせが一意でなければなりません。

- キーアイテムの UNIQUEKEY 属性は設定によって上書きされません。これらは個別に一意でなければなりません。そのため、ITEMSET と ITEMREF の両エレメントに対して True を指定しても意味はありません。
- 繰り返しフォームの一意のキーは、同一のフォームバージョン内でのみチェックされます。
- 繰り返しフォームの一意のキーをチェックする際、削除された繰り返しフォームのインスタンスは考慮されません。

削除された繰り返しフォームのインスタンスは、削除を解除するとキーアイテムが一意でなくなる場合、解除できません。

子

- 1 つ以上の Sectionref 定義。各 Sectionref は定義済みのセクションを参照します。
- CRF ヘルプドキュメントの位置を示す 0 または 1 つの HelpLink 定義。
- を示す、0 または 1 つ以上の KeyItemref リファレンス。KeyItemref 定義は、繰り返しフォームの定義でのみ有効です。
- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エレメントには 1 つ以上の Translation 定義を含める必要があります。Translation エレメントはそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エレメントは、スタディコンポーネント定義内の最後のチャイルドエレメントにする必要があります。Form 定義では、**MNEMONIC**、**NOTE**、**TITLE** の各属性を翻訳できます。属性を翻訳する場合は、スタディコンポーネント定義の一部であるすべての翻訳可能な属性の翻訳定義を含めます。

例 1

この例は症例背景フォームの定義を示しています。2 つのセクション、症例背景 (DEM; Demographics) および喫煙履歴 (SH; Smoking History) から構成されます。例では、個々のコンポーネントおよびコントロールが既に定義されていると仮定します。以下の定義方法を示します。

- コンポーネントとコントロールを含んでいるアイテム
- アイテムを含んでいるセクション
- セクションを含んでいるフォーム

1 症例背景 (Demographics) セクション用のアイテムを定義します。

```
<ITEM REFNAME="GENDER" QUESTION="Gender: ">
  <CONTROLREF REFNAME="GENDERRADIO"/>
</ITEM>
<ITEM REFNAME="HEIGHT" QUESTION="Height: ">
  <CONTROLREF REFNAME="HEIGHTTEXT"/>
</ITEM>
<ITEM REFNAME="WEIGHT" QUESTION="Weight: ">
  <CONTROLREF REFNAME="NUMTEXT"/>
</ITEM>
<ITEM REFNAME="FRAME" QUESTION="Frame Size: ">
  <CONTROLREF REFNAME="FRAMEPULLDOWN"/>
</ITEM>
<ITEM REFNAME="RACE" QUESTION="Race: ">
  <CONTROLREF REFNAME="RACEGROUP"/>
```

```
</ITEM>
```

2 喫煙経歴 (Smoking History) セクション用のアイテムを定義します。

```
<ITEM REFNAME="SMOKE" QUESTION="Has the subject ever smoked? ">
  <CONTROLREF REFNAME="SMOKERADIO"/>
</ITEM>
<ITEM REFNAME="EVERSMOKED" QUESTION="If the subject has ever smoked, has the subject
quit smoking? ">
  <CONTROLREF REFNAME="SMOKERADIO"/>
</ITEM>
<ITEM REFNAME="WHATSMOKED" QUESTION="If the subject currently smokes, the subject
smokes: ">
  <CONTROLREF REFNAME="SMOKECHECKBOX"/>
</ITEM>
```

3 症例背景 (Demographics) および喫煙経歴 (Smoking History) セクションを定義します。

```
<SECTION REFNAME="DEM" TITLE="Demographics">
  <ITEMREF REFNAME="GENDER" ORDER="1"/>
  <ITEMREF REFNAME="HEIGHT" ORDER="2"/>
  <ITEMREF REFNAME="WEIGHT" ORDER="3"/>
  <ITEMREF REFNAME="FRAME" ORDER="4"/>
  <ITEMREF REFNAME="RACE" ORDER="5"/>
</SECTION>

<SECTION REFNAME="SH" TITLE="Smoking History">
  <ITEMREF REFNAME="SMOKE" ORDER="1"/>
  <ITEMREF REFNAME="EVERSMOKED" ORDER="2"/>
  <ITEMREF REFNAME="WHATSMOKED" ORDER="3"/>
</SECTION>
```

4 症例背景フォームを定義

```
<FORM REFNAME="DEM" TITLE="Demographics" MNEMONIC="DEM" TYPE="CRF">
  <SECTIONREF REFNAME="DEM"/>
  <SECTIONREF REFNAME="SH"/>
</FORM>
```

例 2

この例では、TITLE、MNEMONIC、および NOTE 属性の翻訳を含んだフォームを示しています。

```
<FORM REFNAME="DEM" TITLE="Demographics" MNEMONIC="DEM"
NOTE="Use this form to record demographic data and smoking history."
TYPE="CRF">
  <SECTIONREF REFNAME="DEM"/><SECTIONREF REFNAME="SH"/>
  <TRANSLATIONS>
    <TRANSLATION NAME="TITLE" LOCALE="da-DK" DISPLAYTEXT="Demografi"/>
    <TRANSLATION NAME="TITLE" LOCALE="sy-SE" DISPLAYTEXT="Demografi"/>
    <TRANSLATION NAME="MNEMONIC" LOCALE="da-DK" DISPLAYTEXT="DEM"/>
    <TRANSLATION NAME="MNEMONIC" LOCALE="sy-SE" DISPLAYTEXT="DEM"/>
    <TRANSLATION NAME="NOTE" LOCALE="da-DK"
      DISPLAYTEXT="Verwenden Sie dieses Formular zur Erfassung von demografischen
Daten und Rauchen Geschichte"/>
    <TRANSLATION NAME="NOTE" LOCALE="sy-SE"
      DISPLAYTEXT="Använd detta formulär för att registrera demografiska data
och rökning historia"/>
  </TRANSLATIONS>
</FORM>
```

FormSet

目的

CRF を以下のいずれかにグループ化できます。

- ビジット
- 共通 CRF
- スクリーニング CRF
- 症例登録 CRF
- モニタリング CRF
- ステータスレポート
- 関連 CRF

構文

```
<FORMSET
  REFNAME="name"
  [DESIGNNOTE="text"]
  TYPE="VISIT|ENROLLMENT|SCREENING|COMMONCRF|
        MONITOR|STATUS|REGDOCS|VISITREPORTS|RELATION"
  [UUID="id"]
  TITLE="text"
  MNEMONIC="name"
  [LANGUAGE="name"]
  [STARTHOUR="n"]
  [ORDER="n"]
  [SCHEDULED="true|false"]
  [UNSCHEDULED="true|false"]
  [DYNAMIC="true|false"]
  [OPTIONAL="true|false"]
  [REPEATING="true|false"]
  [HELPTEXT="text"]>
  <FORMREF+ attributes/>
  <TRANSLATIONS/>
</FORMSET>
```

属性

REFNAME="*name*"

スタディバージョンの定義でフォームセットを参照するときに使用される名前。この名前はすべてのフォームセットの中で一意でなければなりません。すべてのスタディでは、Conflict という RefName が必要です。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

TYPE="VISIT|ENROLLMENT|SCREENING|COMMONCRF|MONITOR|STATUS|REGDOCS|VISITREPORTS|RELATION"

フォームセットのタイプ。オプションは以下の通りです。

- **VISIT** — 症例ビジットのセット。
- **ENROLLMENT** — 症例登録 CRF のセット。
- **SCREENING** — スクリーニング CRF のセット。
- **COMMONCRF** — 複数のビジットに存在し、累積データを含む CRF のセット。
たとえば、併用薬フォームや有害事象フォーム。共通 CRF を定義するには、**COMMONCRF** フォームセットおよび共通フォームが表示されるそれぞれの **VISIT** フォームセットの両方に、Form 定義を含めてください。

必須

警告: 一度 **COMMONCRF** フォームセットに共通 CRF を含めることで共通 CRF を定義し、いずれかの症例のデータが入力された場合、**StudyVersion** 定義の **COMMONCRF** フォームセットから共通 CRF を削除して、フォームを通常の CRF 状態に戻すことはできません。同様に、症例のデータが入力された後に、**COMMONCRF** フォームセットに追加して、通常の **VISIT** フォームセットにある CRF を共通 CRF に変更することはできません。これらの方法でスタディバージョンを変更しようとする、症例データを失うことになります。

既存の共通 CRF と同じデータを取り込む通常の CRF を作成する必要がある場合、共通 CRF とは異なる **REFNAME** を持つ別のフォーム定義として CRF を作成し、スタディバージョンの中の該当する **VISIT** フォームセットに追加してください。

- **MONITOR** — モニタリングフォームセット: 規制ドキュメントチェックリストおよびビジットレポート。
- **STATUS** — サポートされていません。
- **RELATION** — 1 組の関連フォームを定義します。フォーム間の関連を設定した場合、1 つのフォームを表示すると、ある画面から他の画面にナビゲートできるのみならず、関連付けられたフォームも同様に表示されます。
- **REGDOCS** — サポートされていません。
- **VISITREPORTS** — サポートされていません。

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。

注：いくつかのタイプのフォームセットは特定の **UUID** を必要とします。これらのタイプのフォームセットを作成する場合、以下で特定されているように **UUID** を正確に使用しなければなりません。

注：InForm アプリケーションと MedML インストラユーティリティは、**UUID** の英字を大文字に変換します。

コンポーネント	UUID
症例登録	d882ce3a-0f42-11d2-a419- 00a0c963e0ac
スクリーニング	D882CE38-0F42-11D2-A419-00A0C963E0AC
共通	9d6bbc5d-5811-11d2-8065- 00a0c9af7674
競合	PF_UUID_CONFLICT_FORMSET

TITLE="*text*"

フォームセットのタイトル。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。必須

MNEMONIC="*name*"

ビジットタイトルの略称。症例ビジットを定義するフォームセットの **REFNAME** と同じにすることができます。この略称はビジットウィンドウの一番上のタイムラインの中に表示されます。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。必須

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

STARTHOUR="*n*"

スタディの開始からの時間 (たとえばビジット 1 より # 時間)。ビジットカルキュレーターにおいてビジットの日時を提案する場合に使用されます。オプション。

ORDER="*n*"

ビジットスケジュールにおけるフォームセットの順序を示す数値。値を指定しない場合、MedML インストーラユーティリティは、フォームセットを StudyVersion 定義に入力された順番で順序付けします。オプション

SCHEDULED="*true|false*"

ビジットが予定されているかどうかを示します。つまり、ビジットの日時のスケジューリングがスタディプロトコルで指定されているかどうかを示します。オプションは true または false です。デフォルトは、True になっています。オプション

UNSCHEDULED="*true|false*"

ビジットが予定外かどうかを示します。つまり、ビジットの日時のスケジューリングがスタディプロトコルで指定されているかどうかを示します。オプションは true または false です。デフォルトは、False になっています。オプション

DYNAMIC="*true|false*"

フォーム中のデータアイテムの値に基づき、ダイナミックにビジットが割り当てられるかどうかを示します。オプションは true または false です。デフォルトは、False になっています。

OPTIONAL="*true|false*"

ビジットがオプションかどうかを示します。内部使用向けのみです。デフォルトは、False

になっています。

REPEATING="true|false"

予定外のビジットと共に使用されます。複数回の予定外のビジットが作成可能かどうかを示します。オプションは true または false です。デフォルトは、False になっています。
オプション

REPEATING フォームセットを作成する場合、以下のルールが適用されます。

- 繰り返しフォームセットの最初のフォームは共通 CRF ではあってはなりません。
- 繰り返しフォームセットの最初のフォームには、DateTimeControl 定義 (ビジットの月、日、年、時、分を含み、少なくとも年と月の入力が必要) を持つ 1 つのアイテムからなる Section 定義が必要です。
- Section、Item、および DateTimeControl 定義には、以下の特定の必須 UUID が含まれている必要があります。

注：InForm ソフトウェアと MedML インストーラユーティリティは、UUID の英字を大文字に変換します。

コンポーネント	UUID
セクション	BD991BBE-B0A4-11D2-80E3-00A0C9AF7674
アイテム	BD991BBF-B0A4-11D2-80E3-00A0C9AF7674
日時コントロール	BD991BC0-B0A4-11D2-80E3-00A0C9AF7674

HELPTEXT="text"

フォームセットに関連付けられたフローティングヘルプのテキスト。オプション

子

- 1 つ以上の Formref 定義。各 Formref エLEMENTは定義済みのフォームを参照します。
- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エLEMENTには 1 つ以上の Translation 定義を含める必要があります。Translation エLEMENTはそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エLEMENTは、スタディコンポーネント定義内の最後のチャイルドELEMENTにする必要があります。FormSet 定義では、MNEMONIC と **TITLE** 属性を翻訳できます。属性を翻訳する場合は、スタディコンポーネント定義の一部であるすべての翻訳可能な属性の翻訳定義を含めます。

例 1

この例は、スタディの 2 週目のビジット フォームセットを示しています。フォームセットは以下の 3 つのフォームから構成されます。

- バイタルサイン/血圧 (VS/BP; Vital Signs/Blood Pressure)
- 有害事象 (AE; Adverse Experience)
- コンプライアンス (CMP; Compliance)

この例のために、各フォームのコンポーネント、コントロール、アイテム、およびセクションは事前に定義されていると仮定します。

注：他のコンポーネントと異なり、フォームセットは再使用できません。したがって、スタディバージョンの定義内にこれらを定義しています。

- 1 バイタルサイン/血圧 (Vital Signs/Blood Pressure) フォームを定義します。バイタルサイン (VS) セクションおよび血圧 (BP) セクションから構成されます。

```
<FORM REFNAME="VSBP" TITLE="Vital Signs/Blood Pressure" MNEMONIC="VS/BP"
  TYPE="CRF">
  <SECTIONREF REFNAME="VS"/>
  <SECTIONREF REFNAME="BP"/>
</FORM>
```

- 2 有害事象 (AE) という 1 つの繰り返しセクションを持つ有害事象 (Adverse Experience) CRF を定義します。

```
<FORM REFNAME="AE" TITLE="Adverse Experience" MNEMONIC="AE" TYPE="CRF">
```



```
<SECTIONREF REFNAME="AE"/>
</FORM>
```

- 3 コンプライアンス CRF を定義します。症例コンプライアンス情報 (PCI) および薬剤キット (DK) セクションから構成されます。

```
<FORM REFNAME="CMP" TITLE="Subject Compliance" MNEMONIC="CMP"
  TYPE="CRF">
  <SECTIONREF REFNAME="PCI"/>
  <SECTIONREF REFNAME="DK"/>
</FORM>
```

- 4 スタディバージョンの定義内で、Formref エlementを使用した 3 つの Form 定義を含む、第 2 週 (Week 2) ビジットのフォームセットを定義します。

```
<STUDYVERSION
  VERSION="1"
  STUDYNAME="Hypertension Study"
  PROTOCOL="Protocol XYZZY">
  <FORMSET REFNAME="Visit1" TITLE="Week -4" UUID="03b0d5d8-
7f2c-11d2-a728-00a0c977c64b"
    LANGUAGE="English" TYPE="Visit"
    SCHEDULED="true" ORDER="1">
    <FORMREF REFNAME="DEM"/>
    <FORMREF REFNAME="FH"/>
    <FORMREF REFNAME="DOC"/>
  </FORMSET>
  <FORMSET REFNAME="Visit2" TITLE="Week -2"
    TYPE="Visit" SCHEDULED="true">
    <FORMREF REFNAME="VS"/>
  </FORMSET>
  <FORMSET REFNAME="Visit3" TITLE="Week 0"
    TYPE="Visit" SCHEDULED="true">
    <FORMREF REFNAME="VS"/>
  </FORMSET>
  <FORMSET REFNAME="WK2" TITLE="Week 2" UUID="F4699051-69E2-11d2-8FB5-
00A0C977C66A">
    TYPE="VISIT"
    SCHEDULED="true">
    <FORMREF REFNAME="VSBP"/>
    <FORMREF REFNAME="AE"/>
    <FORMREF REFNAME="CMP"/>
  </FORMSET>
  <FORMSET REFNAME="screen" TITLE="Screening Log"
    UUID="d882ce38-0f42-11d2-a419- 00a0c963e0ac"
    TYPE="SCREENING" SCHEDULED="false">
    <FORMREF REFNAME="screen"/>
  </FORMSET>
  <FORMSET REFNAME="enroll" TITLE="Enrollment"
    UUID="d882ce3a-0f42-11d2-a419- 00a0c963e0ac"
    TYPE="ENROLLMENT" SCHEDULED="false">
    <FORMREF REFNAME="enroll"/>
  </FORMSET>
</STUDYVERSION>
```

例 2

この例は、スタディバージョンに共通 CRF を含める方法を示しています。この例において、AE CRF は共通 CRF で、CommonCRF フォームセット、および第 -2 週 (Week -2)、0 週 (Week 0) および予定外のビジットのビジットフォームセットに含まれます。CommonCRF フォームセットでは、この例で指定されている UUID と CommonCRF TYPE 属性が必須です。

さらに、この例には、CommonCRF フォームセットの中に AE と CM の CRF をリンクする CMtoAE という関連が含まれています。

```
<STUDYVERSION
  VERSION="2"
  STUDYNAME="Hypertension Study"
```

```

    PROTOCOL="Protocol XYZZY">
<FORMSET REFNAME="CommonCRF" TITLE="Common CRF"
    UUID="9d6bbc5d-5811-11d2-8065-00a0c9af7674" TYPE="COMMONCRF"
    SCHEDULED="false">
    <FORMREF REFNAME="AE" />
</FORMSET>
<FORMSET REFNAME="Visit1" TITLE="Week -4" LANGUAGE="English"
    UUID="03b0d5d8-7f2c-11d2-a728- 00a0c977c64b"
    TYPE="Visit"
    SCHEDULED="true" ORDER="1">
    <FORMREF REFNAME="DEM" ORDER="3"/>
    <FORMREF REFNAME="inclusion" ORDER="1"/>
    <FORMREF REFNAME="exclusion" ORDER="2"/>
</FORMSET>
<FORMSET REFNAME="Visit2" TITLE="Week -2" TYPE="Visit"
    SCHEDULED="true">
    <FORMREF REFNAME="VS"/>
    <FORMREF REFNAME="AE" />
</FORMSET>
<FORMSET REFNAME="Visit3" TITLE="Week 0" TYPE="Visit"
    SCHEDULED="true">
    <FORMREF REFNAME="VS"/>
    <FORMREF REFNAME="AE" />
</FORMSET>
<FORMSET REFNAME="Visit4" TITLE="Week 2" TYPE="Visit"
    SCHEDULED="true" UUID="F4699051-69E2-11d2- 8FB5-00A0C977C66A">
    <FORMREF REFNAME="VS1" ORDER="1"/>
    <FORMREF REFNAME="VS2" ORDER="2"/>
    <FORMREF REFNAME="VS3" ORDER="3"/>
    <FORMREF REFNAME="VS4" ORDER="4"/>
    <FORMREF REFNAME="VS5" ORDER="5"/>
    <FORMREF REFNAME="VS6" ORDER="6"/>
    <FORMREF REFNAME="VS7" ORDER="7"/>
    <FORMREF REFNAME="VS8" ORDER="8"/>
    <FORMREF REFNAME="VS9" ORDER="9"/>
    <FORMREF REFNAME="VS10" ORDER="10"/>
    <FORMREF REFNAME="VS11" ORDER="11"/>
    <FORMREF REFNAME="VS12" ORDER="12"/>
</FORMSET>
<FORMSET REFNAME="Visit5" TITLE="Unsched" TYPE="Visit"
    SCHEDULED="false" REPEATING="true">
    <FORMREF REFNAME="DEM" ORDER="1"/>
    <FORMREF REFNAME="VS" ORDER="2"/>
    <FORMREF REFNAME="AE" />
</FORMSET>
<FORMSET REFNAME="conflict" TITLE="Conflict" MNEMONIC="Conflict"
    UUID="PF_UUID_CONFLICT_FORMSET"
    TYPE="Visit" UNSCHEDULED="false" REPEATING="true"
    DYNAMIC="true">
    <FORMREF REFNAME="SYS_CONFLICT"/>
</FORMSET>
<FORMSET REFNAME="CMtoAE" TYPE="RELATION"
    TITLE="ConMed to AE relationship" MNEMONIC="AECMRelation"
    STARTHOUR="0" ORDER="17">
    <FORMREF REFNAME="AE" ORDER="1" VISITREFNAME="CommonCRF"/>
    <FORMREF REFNAME="CM" ORDER="2" VISITREFNAME="CommonCRF"/>
</FORMSET>
</STUDYVERSION>

```

例 3

この例では、TITLE と MNEMONIC 属性が翻訳されているフォームセット定義を示しています。

```

<FORMSET REFNAME="Visit5" TITLE="Unsched" MNEMONIC="UNSC"
    TYPE="Visit" LANGUAGE="en-US" SCHEDULED="false" REPEATING="true">
    <FORMREF REFNAME="DEM" ORDER="1"/>
    <FORMREF REFNAME="VS" ORDER="2"/>
    <FORMREF REFNAME="AE" />
    <TRANSLATIONS>
        <TRANSLATION NAME="TITLE" LOCALE="nl-NL" DISPLAYTEXT="Ongepland"/>
        <TRANSLATION NAME="MNEMONIC" LOCALE="nl-NL" DISPLAYTEXT="ONGEPL"/>
    </TRANSLATIONS>

```

</FORMSET>

GroupControl

目的

一連のコンポーネントを、ネスト化されたコントロールまたはアイテム定義に含めるために、単一のコンポーネントとして処理できます。GroupControl 定義には以下の種類のコンポーネントを含めることができます。

- CalculatedControl
- CheckBoxControl
- GroupControl
- PullDownControl
- RadioControl
- SimpleControl
- TextControl

構文

```
<GROUPCONTROL
  REFNAME="name"
  [NAME="name"]
  [UUID="id"]
  [DESIGNNOTE="text"]
  [ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"]
  [LAYOUT="VERTICAL|HORIZONTAL|NOWRAP"]
  [CAPTION="text"]
  [LANGUAGE="name"]
  [CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"]
  [ISCOLLAPSIBLE="true|false">
    <CONTROLREF+ attributes/>
  <TRANSLATIONS/>
</GROUPCONTROL>
```

属性

REFNAME="*name*"

他のコントロール定義でグループコントロールを参照するときに使用される名前。この名前はグループコントロール間で一意でなければなりません。必須

NAME="*name*"

他のコントロール定義でグループコントロールを参照するときに使用される名前。
オプション

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"

グループコントロールの中で Controlref に付属しているコンポーネントの位置調整。
LEFT (左)、CENTER (中央)、RIGHT (右)、TOP (上)、MIDDLE (中央)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

LAYOUT="VERTICAL|HORIZONTAL|NOWRAP"

チェックボックスの向き: 垂直(Vertical)、水平(Horizontal)、あるいはノーラップ(Nowrap)。NOWRAP (ノーラップ) を指定すると、ラジオボタンは水平方向に位置づけられ、ユーザがブラウザウィンドウのサイズを変更しても、ボタンは別の行に改行されません。デフォルトは NOWRAP (ノーラップ) です。オプション

CAPTION="*text*"

グループと共に画面に表示されるテキスト。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注: デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"

計算値に対する相対的なキャプションの位置:LEFT (左)、RIGHT (右)、TOP (上)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

ISCOLLAPSIBLE="true|false"

コントロールがダイナミックコントロールかどうかを示します。ダイナミックコントロールとは、ペアレントコントロールの選択によって表示が決定されるコントロールです。

- **True** — グループコントロールのペアレントコントロールが選択されるまでコントロールが表示されません。
- **False** — コントロールは常に表示されます。

デフォルトは、False になっています。オプション

子

- 1 つ以上の Controlref 定義。各 Controlref はグループ内の 1 つのアイテムを識別する事前に定義されたコンポーネントを参照します。

注： Controlref や Unitref 定義を使用して複雑なコントロールを定義する場合、すべての従属するコントロールが同じ**タイプ**の属性で定義され、同じデータタイプを返すことを確認してください。

さらに、コンパウンドコントロールを定義するとき、**InForm** ソフトウェア は最大 **5** 層のネスティングをサポートしています。**5** 層での設計が可能です、パフォーマンスを向上させるためにネストレベルはできるだけ少なくしてください。

- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エレメントには 1 つ以上の Translation 定義を含める必要があります。Translation エレメントはそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エレメントは、スタディコンポーネント定義内の最後のチャイルドエレメントにする必要があります。GroupControl 定義では、**CAPTION** 属性を翻訳できます。

例

この例の GroupControl は、血圧測定の結果を得るために、2 つのテキストコントロールを使用しています。さらに、この例ではグループコントロールの **CAPTION** 属性の翻訳を示しています。

- 1 最初のテキストボックス用に 1 つのテキストコントロールを定義し、スラッシュでテキストボックスを区切り第 2 テキストボックスと単位用に第 2 テキストコントロールを定義しています。

```
<TEXTCONTROL REFNAME="BP1" NAME="BP1"
  CAPTION="/" CAPTIONALIGN="RIGHT"
  HEIGHT="1"
  LENGTH="5" MAXLENGTH="5"
  DATATYPE="INTEGER"/>
<TEXTCONTROL REFNAME="BP2" NAME="BP2"
  CAPTION="mm Hg" CAPTIONALIGN="RIGHT"
  HEIGHT="1"
  LENGTH="5" MAXLENGTH="5"
```

```

        DATATYPE="INTEGER"/>
2  2 つのテキストコントロールを含めるために Controlref を使用してグループコントロールを定義します。

<GROUPCONTROL REFNAME="BPGRP" NAME="BPGRP"
  LANGUAGE="en-US" CAPTION="Diastolic/Systolic" LAYOUT="HORIZONTAL">
  <CONTROLREF REFNAME="BP1" ORDER="1"/>
  <CONTROLREF REFNAME="BP2" ORDER="2"/>
  <TRANSLATIONS>
    <TRANSLATION NAME="CAPTION" LOCALE="es-ES"
      DISPLAYTEXT="Diastólica/Sistólica"/>
  </TRANSLATIONS>
</GROUPCONTROL>

```

GroupType

目的

InForm アプリケーションでサポートされているグループタイプをロードします。サポートされているタイプは、アイテムグループ、クエリ、レポート、および署名です。これらの GroupType が、Oracle の内部にある XML を使用して InForm システムのインストール時にロードされます。この XML への変更を行わないでください。

HelpLink

目的

フォームのアイテムから CRF ヘルプドキュメントへのリンクを定義します。リンクを設定するには:

- HelpLink 定義において、ユーザが CRF 上の下線を引いたアイテムテキストをクリックしたときに見ることができる CRF ヘルプドキュメントの位置を指定します。
- HelpLink 定義を Item 定義のチャイルドとして含めます。

構文

```

<HELPLINK
  DOCREFNAME="name"
  BODYREFNAME="name"
  [BOOKMARK="name"]/>

```

属性

DOCREFNAME="*name*"

CRF ヘルプドキュメントを定義する Documentation コンポーネントの RefName。必須

BODYREFNAME="*name*"

HelpLink が指す情報を含む HTML ファイルを定義する DocBody コンポーネントの RefName。必須

BOOKMARK="*name*"

BODYREFNAME 属性で参照される HTM ファイルの中の HTML ブックマークの

名前。ユーザが CRF 上の下線が引かれたアイテムをクリックした時に、特定アイテムヘルプテキストのドキュメントウィンドウをオープンする場合、必要です。それ以外の場合は、オプションです。

例

以下の例はCRFから特定のCRFヘルプドキュメントへのリンクをセットアップする方法を示しています。

- 1 表示するヘルプテキストを含む HTML ファイルを作成します。HTML ブックマークで CRF がリンクするテキストのそれぞれのセグメントの始まりを指定します。以下の例は、HTM ファイルからのブックマークを正しいフォーマットで示しています。

[illegible]

- 2 CRF ヘルプのための Documentation コンポーネントと、HTML ファイルのための DocBody コンポーネント、およびファイルの中のブックマークを示す HelpLink を定義します。例：

```
<HELPLINK DOCREFNAME="Study"
BODYREFNAME="DEMHELP"
BOOKMARK="Item2"/>
```

- 3 CRF フォームを定義する XML ファイルでは、適切なアイテムのチャイルドとして HelpLink を含めます。

```
<ITEM REFNAME="DEMDOB" QUESTION="Date of Birth:">
  <CONTROLREF REFNAME="dob"/>
  <HELPLINK DOCREFNAME="Study"
    BODYREFNAME="DEMHHELP"
    BOOKMARK="Item2"/>
</ITEM>
```

HTMLTemplate

目的

フォームを表示するとき InForm アプリケーションによって内部的に使用される HTML コードセットを定義します。

注：このコンポーネントは、製品出荷前の **<inf>** データベースのロードに使用されるデフォルトの **XML** ファイルでのみ使用されます。デフォルトの **HTMLTemplate** 定義は **Oracle** による使用のために予約されています。

構文

```
<HTMLTEMPLATE
  TEMPLATENAME="name"
  BROWSERNAME | BROWSER="name"
  RESOURCEUUID="id"
  [DESCRIPTION="text"]
  [UUID="id" />
```

属性

TEMPLATENAME="*name*"

HTML テンプレートの名前。必須

BROWSERNAME | **BROWSER**="*name*"

テンプレートを使用するブラウザの名前。BROWSERNAME か BROWSER 属性のどちらかを使用してください。この名前はテンプレートが使用されているブラウザを定義する Browser コンポーネントの BROWSERNAME 属性と一致します。必須

RESOURCEUUID="*id*"

テンプレート用 Universally Unique Identifier; すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを識別する文字列です。必須

DESCRIPTION="*text*"

HTML テンプレートの説明。オプション

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

例

以下の例では、生成された XML コードが症例登録 CRF の表示で使用する HTML コードをロードするために使用されていることを示しています。

```
<HTMLTEMPLATE
  TEMPLATENAME="PF_HTML_ENROLLOVERFRAMESET"
  BROWSERNAME="NS0000"
  RESOURCEUUID="PF_RESOURCE_NS0000_ENROLLOVERFRAMESET"/>
```

Item

目的

定義済みコントロールを CRF 上のデータポイントに組み入れます。Item は質問文や、データ入力のための 1 つ以上のコントロールから成ります。

構文

```
<ITEM
  REFNAME="name"
  [DESIGNNOTE="text"]
  QUESTION="text"
  [LABEL="text"]
  [UUID="id"]
  [LANGUAGE="name"]
  [CALCULATED="true|false"]
  [ITEMREQUIRED="true|false"]
  [SDVREQUIRED="true|false"]
  [SDVCritical="true|false"]
  [DISPLAYOVERRIDE="READONLY|EDITABLE|HIDDEN"]>
  <CONTROLREF attributes/>
  <HELP LINK attributes/>
  <TRANSLATIONS/>
</ITEM>
```


属性

REFNAME="*name*"

他のコントロール定義で Item を参照するときに使用される名前。この名前はすべてのアイテムの中で一意でなければなりません。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

QUESTION="*text*"

データ項目に対しての、データ入力方法説明や、質問文のテキスト。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。必須

注： フォーマット (たとえば太字、イタリック体、あるいは質問文のテキスト中の色の表示) を埋め込んだ **HTML** エレメントを使用できます。

LABEL="*text*"

アイテムセットのカラムヘッダーでアイテムの質問文として表示される質問文 (短い質問文)。ItemSet の場合、この属性の使用を推奨します。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。**QUESTION** 属性で指定されたテキストがデフォルトです。オプション

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

注： 予定外ビジットなどの繰り返しフォームセットを作成する場合、フォームセット中の最初のフォームには **DateTimeControl** 定義のアイテムを含む必要があります。アイテム定義には以下の **UUID** を含む必要があります。

BD991BBF-B0A4-11D2-80E3- 00A0C9AF7674**LANGUAGE**="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

CALCULATED="true|false"

アイテムに CalculatedControl を含めるかどうかを指定します。オプションは true または false です。デフォルトは、False になっています。Calculated 属性に true を指定すると、アイテム番号は付与されず、ユーザ入力が必要なアイテムと区別されます。オプション

ITEMREQUIRED="true|false"

入力フォームでアイテムが必須かどうかを指定します。オプションは true または false です。デフォルトは、True になっています。False を指定した場合、サブミット後にアイテムは CRF 上でグレー表示され、入力が必須でないことが示されます。ItemSet ではサポートされていません。オプション

SDVREQUIRED="true|false"

アイテムに SDV が必要かどうかを指定します。オプションは true または false です。デフォルトは、True になっています。false を指定した場合、アイテムは CRF の SV ビュー上でグレーで表示され、SDV が必須ではないことが示されます。オプション

SDVCRITICAL="true|false"

アイテムが SDV 対象としてマークされているかどうかを示します。オプションは true または false です。デフォルトは、False になっています。

アイテムセット内と外の両方に存在するアイテムの SDVCRITICAL 属性を true に設定すると、アイテムセット外に存在するアイテムのインスタンスのみが SDV 対象になります。

DISPLAYOVERRIDE="READONLY|EDITABLE|HIDDEN"

ユーザの権限グループの権限によって判別し、デフォルトアイテム表示モードよりも優先されます。オプションは以下の通りです。

- **READONLY** — ユーザは参照することはできるが、権限グループにかかわらず、アイテムにデータを入力することができません。
- **EDITABLE** — ユーザは権限グループにかかわらずアイテムのデータを入力また編集することができます。
- **HIDDEN** — ユーザは権限グループにかかわらず、アイテムを参照することができません。CRF において、アイテム行は番号を付け替えられ、アイテムはフォーム上に表示されません。

注： アイテムがアイテムグループの中にある場合、アイテムグループのディスプレイ設定は **DISPLAYOVERRIDE** 設定より優先されます。

子

- 1 つの Controlref 定義。各 Controlref は、データ入力オプションを指定する定義済みコンポーネントを参照します。
- ユーザが下線付きアイテム説明をクリックするとき、ドキュメントウィンドウに表示される CRF ヘルプドキュメントの位置を示す 0 または 1 つの HelpLink 定義。
- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 また

は 1 つの Translations 定義。Translation エlement には 1 つ以上の Translation 定義を含める必要があります。Translation エlement はそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エlement は、スタディコンポーネント定義内の最後のチャイルドElement にする必要があります。Item 定義では、**LABEL** と **QUESTION** 属性を翻訳できます。属性を翻訳する場合は、スタディコンポーネント定義の一部であるすべての翻訳可能な属性の翻訳定義を含めます。属性を翻訳する場合は、スタディコンポーネント定義の一部であるすべての翻訳可能な属性の翻訳定義を含めます。

例 1

この例は、症例の喫煙習慣に関する情報を入手するアイテムの作成順序を示しています。さらに、この例では、**QUESTION** と **LABEL** の属性の翻訳も示しています。

- 1 各チェックボックスの PFELEMENT 定義を作成します。一般に使われるコンポーネント、たとえば、「未実施 (Not Done)」コンポーネントなどは他のところで定義しておいて、参照利用することができます。

```
<PFELEMENT REFNAME="CIGARETTE" LABEL="Cigarettes" TYPE="STRING" VALUE="cigarette"/>
<PFELEMENT REFNAME="PIPE" LABEL="Pipe" TYPE="STRING" VALUE="pipe"/>
<PFELEMENT REFNAME="CIGAR" LABEL="Cigars" TYPE="STRING" VALUE="cigar"/>
<PFELEMENT REFNAME="ND" LABEL="Not Done" TYPE="STRING" VALUE="nd"/>
```

- 2 Elementref エlement を使用してシンプルコントロールに各 PFELEMENT 定義を含めます。

```
<SIMPLECONTROL REFNAME="CIGARETTE" NAME="WHATSMOKED">
  <ELEMENTREF REFNAME="CIGARETTE"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="PIPE" NAME="WHATSMOKED">
  <ELEMENTREF REFNAME="PIPE"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="CIGAR" NAME="WHATSMOKED">
  <ELEMENTREF REFNAME="CIGAR"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="ND" NAME="WHATSMOKED">
  <ELEMENTREF REFNAME="ND"/>
</SIMPLECONTROL>
```

- 3 Controlref エlement を使用してチェックボックスコントロールの中で SimpleControl 定義を使用します。

```
<CHECKBOXCONTROL REFNAME="SMOKECHECKBOX"
  NAME="SMOKECHECKBOX" LAYOUT="VERTICAL">
  <CONTROLREF REFNAME="CIGARETTE" ORDER="1"/>
  <CONTROLREF REFNAME="PIPE" ORDER="2"/>
  <CONTROLREF REFNAME="CIGAR" ORDER="3"/>
  <CONTROLREF REFNAME="ND" ORDER="4"/>
</CHECKBOXCONTROL>
```

- 4 Controlref エlement を使用して、Item 定義の中で CheckBoxControl 定義を使用します。

```
<ITEM REFNAME="WHATSMOKED" QUESTION="If the subject currently smokes, the subject
smokes: ">
  <CONTROLREF REFNAME="SMOKECHECKBOX"/>
</ITEM>
```

- 5 アイテムの CRF ヘルプを示す HelpLink 定義を追加します。

```
<ITEM REFNAME="WHATSMOKED" QUESTION="If the subject currently smokes, the subject
smokes:"
LABEL="Smoking materials" LANGUAGE="en-US">
  <CONTROLREF REFNAME="SMOKECHECKBOX"/>
  <HELPLINK DOCREFNAME="Study" BODYREFNAME="DEMHELP" BOOKMARK="Item9"/>
  <TRANSLATIONS>
    <TRANSLATION NAME="QUESTION" LOCALE="es-ES"
      DISPLAYTEXT="Si el paciente fuma actualmente, el paciente fuma"/>
    <TRANSLATION NAME="LABEL" LOCALE="es-ES" DISPLAYTEXT="Materiales de
```

```

Fumar"/>
<TRANSLATION NAME="QUESTION" LOCALE="fr-FR" DISPLAYTEXT="Si le patient fume
actuellement, le patient fume"/>
<TRANSLATION NAME="LABEL" LOCALE="fr-FR" DISPLAYTEXT="Fumeur"/>
</TRANSLATIONS>
</ITEM>

```

例 2

この例は、繰り返しフォームセットの最初のフォームで必要とされるビジット日時 (Date of Visit) セクションの定義を示しています。このセクションの定義は、DateTimeControl を使用した Item 定義を使用します。Section、Item および DateTimeControl 定義はすべて特定の必要な UUID を含んでいます。

```

<DATETIMECONTROL REFNAME="DOV" NAME="DOV"
  UUID="BD991BC0-B0A4-11D2-80E3-00A0C9AF7674"
  STARTYEAR="1997" ENDYEAR="2004"
  DISPLAYDAY="true" DISPLAYMONTH="true"
  DISPLAYYEAR="true" DISPLAYHOUR="true"
  DISPLAYMINUTE="true"
  REQUIREMONTH="true" REQUIREYEAR="true"/>
<ITEM REFNAME="DOV" QUESTION="Date and time of visit:"
  UUID="BD991BBF-B0A4-11D2-80E3-00A0C9AF7674">
  <CONTROLREF REFNAME="DOV"/>
</ITEM>
<SECTION REFNAME="DOV" TITLE="Date Of Visit"
  UUID="BD991BBE-B0A4-11D2-80E3-00A0C9AF7674">
  <ITEMREF REFNAME="DOV" ORDER="1"/>
</SECTION>

```

ItemGroup

目的

権限グループでの表示オプションを設定する目的で定義済みアイテムをグループ化します。

構文

```
<ITEMGROUP  
  GROUPNAME="name"  
  [GROUPDESCRIPTION="name"]  
  <ITEMREF attributes/>  
</ITEMGROUP>
```

属性

GROUPNAME="*name*"

ItemGroupRef 定義でアイテムグループを参照するときに使用される名前。この名前はすべてのアイテムグループの中で一意でなければなりません。必須

GROUPDESCRIPTION="*name*"

アイテムグループの説明。オプション

子

ItemGroup には、いくつでも Itemref を含めることができます。各 Itemref エレメントは、アイテムグループに含められる 1 つのアイテムを指定します。

例

以下の例は、CRC_Hidden ItemGroup の中に RACE と GENDER アイテムを含んでいます。

```
<ITEMGROUP  
  GROUPNAME="CRC_Hidden"  
  GROUPDESCRIPTION="Items hidden from CRC">  
  <ITEMREF REFNAME="GENDER"/>  
  <ITEMREF REFNAME="RACE"/>  
</ITEMGROUP>
```

ItemGroupref

目的

定義済みアイテムグループを権限グループの定義に含めます。ItemGroupref は権限グループのチャイルドとしてのみ定義されます。単独のコンポーネントとしてサブミットされません。

構文

```
<ITEMGROUPREF
  REFNAME="name"
  DISPLAYOVERRIDE="READONLY|EDITABLE|HIDDEN"/>
```

属性

REFNAME="*name*"

RightsGroup 定義でアイテムグループを参照するときに使用される名前。この名前はすべてのアイテムグループの中でユニークである必要があります。必須

DISPLAYOVERRIDE="READONLY|EDITABLE|HIDDEN"

アイテムグループ内のアイテムのアイテムレベル表示モードよりも優先されます。アイテムレベル表示モードは、Item 定義の DISPLAYOVERRIDE 属性で設定されます。値は以下の通りです。

- **READONLY** — ユーザは参照することはできるが、権限グループやアイテムレベル表示モードにかかわらず、アイテムにデータを入力することができません。
- **EDITABLE** — ユーザは権限グループやアイテムレベル表示モードにかかわらずアイテムのデータを入力また編集することができます。
- **HIDDEN** — ユーザは権限グループやアイテムレベル表示モードにかかわらず、アイテムを参照することができません。CRF において、アイテム行は番号を付け替えられ、アイテムはフォーム上に表示されません。

必須

例

以下の例では、CRC RG 権限グループの中に CRC_Hidden アイテムグループを挿入しています。

```
<ITEMGROUP
  GROUPNAME="CRC_Hidden"
  GROUPDESCRIPTION="Items hidden from CRC">
  <ITEMREF REFNAME="GENDER"/>
  <ITEMREF REFNAME="RACE"/>
</ITEMGROUP>

<RIGHTSGROUP GROUPNAME="CRC RG">
  <RIGHTREF RIGHT="Print" />
  <RIGHTREF RIGHT="Custom Reports" />
```

```

<RIGHTREF RIGHT="Canned Reports" />
<RIGHTREF RIGHT="Enroll Patients" />
<RIGHTREF RIGHT="View CRF" />
<RIGHTREF RIGHT="Enter Data into a CRF" />
<RIGHTREF RIGHT="Edit Data on a CRF" />
<RIGHTREF RIGHT="Enter Comments into a CRF" />
<RIGHTREF RIGHT="Mark a CRF as Ready for SV" />
<RIGHTREF RIGHT="Mark and unmark a CRB as Ready for SV" />
<RIGHTREF RIGHT="Answer Query" />
<RIGHTREF RIGHT="View Connections" />
<RIGHTREF RIGHT="View Default Connection" />
<RIGHTREF RIGHT="Synchronize New Data" />
<USERREF USERNAME="crc" />
<ITEMGROUPREF REFNAME="CRC_Hidden" DISPLAYOVERRIDE="HIDDEN"/>
</RIGHTSGROUP>

```

Itemref

目的

- 定義済みアイテムをセクション、アイテムセット、またはアイテムグループの定義に含めます。セクション、アイテムセット、またはアイテムグループの定義の各アイテムにつき 1 つの Itemref エlementを含めます。
- マッピング定義の PATH エlementで、マッピングされるコントロールの位置として、アイテムの RefName を指定します。PATH エlementによって定義された RefName パスでは、1 つの Itemref エlementを含めてください。

Itemref は Section または Path のチャイルドとしてのみ定義されます。単独のコンポーネントとしてサブミットされません。

構文

```

<ITEMREF
  REFNAME="name"
  [KEYITEM="true|false"]
  [UNIQUEKEY="true|false"]
  [ORDER="n"]/>

```

属性

REFNAME="*name*"

指定アイテムの RefName。必須

KEYITEM="*true|false*"

指定アイテムが、Itemset 定義のキーアイテムであるかどうかを示します。テキスト、数値、または日付/時間の値を持つアイテムのみがキーアイテムとなります。チェックボックスコントロールやグループコントロール、ラジオコントロール、ドロップダウンリストなどのコンパウンドコントロールを持つアイテムをキーアイテムにすることはできません。オプションは true または false です。デフォルトは、False になっています。

オプション

注： **KeyItemref** エレメントを使用して、繰り返しフォームのキーアイテムを指定できます。また、アイテムセットでキーアイテムと定義されるアイテムは、繰り返しフォームのキーアイテムに定義することはできません。

UNIQUEKEY="true|false"

キーアイテムがビジットおよびフォーム、アイテムセット内で一意であるべきかどうかを示します。キーアイテムが一意であると定義されている状態で、アイテムセットの 2 つの行が同じキーアイテム値で、同じビジットとフォームでサブミットされると、InForm アプリケーションは 2 番目のインスタンスの入力を受け付けません。オプションは true または false です。デフォルトは、False になっています。オプション

ORDER="n"

Section 定義の中で、各 Itemref が現れる順序。順序を指定しない場合、MedML インストーラユーティリティは、Itemref が参照するアイテムが入力された順番で順序付けします。オプション

注： **PATH** エレメントが **Itemref** 定義を含む場合、**REFNAME** 属性のみが有効です。

例

この例は、高血圧症の期間 (Duration of Hypertension) と呼ばれるセクションの定義 (2 つのアイテムを含む) における itemrefs の使用を説明しています。詳細については、「**Section**」 (195ページ) を参照してください。

```
<SECTION REFNAME="HYPERTENSION"
  TITLE="Duration of Hypertension">
  <ITEMREF REFNAME="HTYESNO" ORDER="1"/>
  <ITEMREF REFNAME="HTDURATION" ORDER="2"/>
</SECTION>
```

以下は、PATH エレメントでの Itemref 定義の使用例を示しています。CLIN アイテムはマップされた TESTTEXT コントロールが存在する位置です。

```
<PATH>
  <CHAPTERREF REFNAME="PF_ALL_VISITS"/>
  <PAGEREF REFNAME="LAB"/>
  <SECTIONREF REFNAME="LAB"/>
  <ITEMSETREF REFNAME="LAB"/>
  <ITEMREF REFNAME="CLIN"/>
  <CONTROLREF REFNAME="CCGROUP"/>
  <CONTROLREF REFNAME="TESTTEXT"/>
</PATH>
```

以下は、Itemref エレメントを持つ ItemSet 定義に、3 つのキーアイテムを一意のキーの組み合わせとして指定する例です。

```
<ITEMSET REFNAME="itsDOSE" ITEMREQUIRED="true"
  SDVREQUIRED="true" UNIQUEKEY="true">
  <ITEMREF REFNAME="itmDOSEFromDate" ORDER="1"
```



```

    UNIQUEKEY="true" KEYITEM="true"/>
  <ITEMREF REFNAME="itmDOSETToDate" ORDER="2"
    UNIQUEKEY="true" KEYITEM="true"/>
  <ITEMREF REFNAME="itmDOSEBlisterpack" ORDER="3"
    UNIQUEKEY="true" KEYITEM="true"/>
  <ITEMREF REFNAME="itmDOSETotalCaps" ORDER="4"/>
  <ITEMREF REFNAME="itmDOSENum" ORDER="5"/>
  <ITEMREF REFNAME="itmDOSEMissed" ORDER="6"/>
  <ITEMREF REFNAME="itmDOSEReasChange" ORDER="7"/>
  <ITEMREF REFNAME="Item_InclComments" ORDER="8"/>
</ITEMSET>

```

ItemSet

目的

繰り返し情報のあるフォーム (有害事象や併用薬フォームなど) で使用するアイテムセットを定義します。各アイテムセットがフォームの 1 行に表示されます。

- REPEATING 属性が true に設定されている Section 定義にのみ ItemSet コンポーネントを追加できます。
- 1 つの REPEATING Section 定義に 1 つの ItemSet 定義しか追加できません。
- 通常の繰り返しでない Item 定義は REPEATING Section 定義に追加できません。

構文

```

<ITEMSET
  REFNAME="name"
  [DESIGNNOTE="text"]
  [UUID="id"]
  [ITEMREQUIRED="true|false"]
  [SDVREQUIRED="true|false"]
  [SDVREQUIRED="true|false"]
  [DISPLAYOVERRIDE="READONLY|EDITABLE|HIDDEN"]
  [UNIQUEKEY="true|false"]>
  <ITEMREF+ attributes/>
</ITEMSET>

```

属性

REFNAME="*name*"

他のコントロール定義でアイテムセットを参照するときに使用される名前。この名前は ItemSet 定義間で一意である必要があります。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベース、およびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

ITEMREQUIRED="*true|false*"

CRF でこのアイテムセットが必須かどうかを指定します。オプションは true または false です。デフォルトは、True になっています。False を指定した場合、サブミット後にアイテムセットは CRF 上でグレー表示され、入力が必要でないことが示されます。オプション

SDVREQUIRED="*true|false*"

アイテムセットに SDV が必要かどうかを指定します。オプションは true または false です。デフォルトは、True になっています。False を指定した場合、SDV が必要ではないことを示すために、アイテムセットは CRF の SDV ビュー上でグレー表示されます。オプション

SDVCRITICAL="*true|false*"

アイテムセットが SDV 対象としてマークされているかどうかを示します。オプションは true または false です。デフォルトは、False になっています。

DISPLAYOVERRIDE="READONLY|EDITABLE|HIDDEN"

ユーザ権限グループの権限によって決定され、デフォルトのアイテムセット表示モードよりも優先されます。値は次のとおりです。

- **READONLY** — ユーザは参照は可能ですが、権限グループにかかわらず、アイテムセットへのデータ入力はできません。
- **EDITABLE** — ユーザは権限グループにかかわらずアイテムセットを編集できます。
- **HIDDEN** — ユーザは権限グループにかかわらず、アイテムセットを参照できません。CRF 上で、アイテムセット行に番号が再度割り当てられ、アイテムセットはフォーム上に表示されません。

UNIQUEKEY="*true|false*"

キーアイテムとして定義されているすべてのアイテムによって形成されるキーが一意的かどうかを示します。オプションは true または false です。デフォルトは、False になっています。オプション

- **UNIQUEKEY=true** であるアイテムセット内のキーアイテムの場合、そのアイテムは

すべてのアイテムセットの行で一意でなければなりません。

- アイテムセットキーのアイテムの一意性をチェックする際、削除されたアイテムセット行が考慮されます。
- アイテムセット内のアイテムは、繰り返しフォームのキーアイテムになることができません。

True を指定した場合

- アイテムセットの 2 行が同じキーアイテム値で、同じビジットとフォームでサブミットされると、InForm ソフトウェア は 2 番目のインスタンスの入力を受け付けません。
- キーアイテムのすべての組み合わせが一意でなければなりません。1 つのアイテムセット行のすべてのキーアイテムが、別のアイテムセット行のキーアイテムと同じである場合、エラーが発生します。
- キーアイテムの UNIQUEKEY 属性は設定によって上書きされません。これらは個別に一意でなければなりません。そのため、ITEMSET と ITEMREF の両エレメントに対して True を指定しても意味はありません。

REVISION="number"

アイテムセットのリビジョン番号。

INITIALROWCOUNT="number"

アイテムセット内の行数。

ROWCOUNTFIXED="true|false"

アイテムセットに固定行が含まれるかどうかを示します。

子

ItemSet 定義は 1 つ以上の Itemref 定義を含む必要があります。各 Itemref は事前に定義されたアイテムを参照します。

例

この例は、有害事象を説明する繰り返しアイテムセットを追加する方法を示しています。

- 1 アイテムセット内の各アイテムについてコントロールの定義を作成します。ユーザがデータを入力する CRF 上に表示される質問文のテキストを Item 定義の **QUESTION** 属性に指定します。また、データが表形式の CRF 上に表示される場合にカラムヘッダに表示されるテキストを **LABEL** 属性に指定します。

```
<TEXTCONTROL REFNAME="SYMPTOMS"
  NAME="SYMPTOMS"
  HEIGHT="4"
  LENGTH="40"
  MAXLENGTH="40"
  DATATYPE="STRING"/> <DATETIMECONTROL REFNAME="AESTARTDATE" NAME="AE_STARTDATE"
  UUID="AE-STARTDATE-UUID"
  STARTYEAR="1998"
  ENDYEAR="2000"
  DISPLAYMONTH="true"
  DISPLAYDAY="true"
  DISPLAYYEAR="true"
```

```

    REQUIREMONTH="true"
    REQUIREDAY="true"
    REQUIREYEAR="true"/> <DATETIMECONTROL REFNAME="AEENDDATE" NAME="AE_ENDDATE"
    UUID="AE-ENDDATE-UUID"
    STARTYEAR="1998"
    ENDYEAR="2000"
    DISPLAYMONTH="true"
    DISPLAYDAY="true"
    DISPLAYYEAR="true"
    REQUIREMONTH="true"
    REQUIREDAY="true"
    REQUIREYEAR="true"/> <ITEM REFNAME="AE_SYMPTOM"
    QUESTION="Describe the patient's symptoms: "
    LABEL="Symptoms">
    <CONTROLREF REFNAME="SYMPTOMS"/>
</ITEM>    <ITEM REFNAME="AE_START"
    QUESTION="Date that symptoms first appeared: "
    LABEL="Start Date">
    <CONTROLREF REFNAME="SYMPTOMS"/>
</ITEM>    <ITEM REFNAME="AE_END"
    QUESTION="Date that symptoms stopped: "
    LABEL="End Date">
    <CONTROLREF REFNAME="SYMPTOMS"/>
</ITEM>

```

2 Itemref 定義を使用して、ItemSet 定義で Item 定義をグループ化します。

```

<ITEMSET REFNAME="AE">
    <ITEMREF REFNAME="AE_SYMPTOM"/>
    <ITEMREF REFNAME="AE_START"/>
    <ITEMREF REFNAME="AE_END"/>
</ITEMSET>

```

3 Itemref 定義を使用して、アイテムセットを含む Section 定義を作成します。

Section 定義の **REPEATING** 属性の値は、true である必要があります。

```

<SECTION REFNAME="AESECTION"
    TITLE="Adverse Events"
    REPEATING="true">
    <ITEMREF REFNAME="AE"/>
</SECTION>

```

ItemSetCell

目的

繰り返しデータアイテムセットの特定のセルが、定義済みの値か、空欄か、あるいは定義済みで空欄であることを示します。

構文

```

<ITEMSETCELL
    ITEMREFNAME="name"
    ITEMSETROW="number"
    PREDEFINEDVALUEREFFNAME="name"
    ISBLANK="true|false"
</ITEMSETCELL>

```

属性

ITEMREFNAME="*name*"

空白のセルに関連付けられたアイテムの RefName。必須

ITEMSETROW="*number*"

セルが存在するアイテムセットの行番号。必須

PREDEFINEDVALUEREFFNAME="*name*"

アイテムセット内で定義済みの値を含むセルのコントロールの RefName。オプション

ISBLANK="*true|false*"

アイテムセット内のセルが空白かどうかを示します。オプション

KeyItemref

目的

フォームのインスタンス間を移動するために、繰り返しフォーム内の 1 つのアイテムをキーアイテムとして識別します。InForm アプリケーションのドロップダウンリストにキーアイテムの値が表示され、フォームの特定のインスタンスに移動できます。

KeyItemref は、自身が含まれているフォームの子としてのみ表示され、単独コンポーネントとしてサブミットされることはありません。

次のコントロールタイプのアイテムのみ、キーアイテムとなり、選択リストに含めることができます。

- 日付/時間コントロール。
- ドロップダウンリスト。
- ラジオグループ。

ラジオグループとして定義されたアイテムに、チェックボックスコントロール、グループコントロール、他のラジオグループなどあらゆるタイプのネスト化されたコンパウンドコントロールが含まれている場合は、このアイテムをキーアイテムに指定することはできません。

- テキストボックス。

注： 繰り返しフォームと同様に、アイテムセット用にキーアイテムを指定することができます。アイテムセットキーアイテムを指定するには、ItemSet 定義に Item エlementを含む Itemref エlementの **KEYITEM** 属性を使用します。アイテムセット内のアイテムは、繰り返しフォームのキーアイテムにできません。

構文

<KEYITEMREF

SECTIONREFNAME="*name*"ITEMREFNAME="*name*"[ORDER="*n*"]

```
[UNIQUEKEY="true|false"]/>
```

属性

SECTIONREFNAME="*name*"

KeyItemref が参照するアイテムが存在するセクションの RefName。必須

ITEMREFNAME="*name*"

KeyItemref が繰り返しフォームの定義内に含めるアイテムの RefName。必須

ORDER="*n*"

繰り返しフォームにおいて、ナビゲーション上のドロップダウンリストで各キーアイテムが表示される順番。順序を指定しなかった場合、MedML インストーラユーティリティは Form 定義に現れる順にキーアイテムを並べます。オプション

UNIQUEKEY="*true|false*"

キーアイテムがビジットおよびフォーム内で一意であるべきかどうかを示します。オプションは true または false です。デフォルトは、False になっています。オプション

キーアイテムを一意に定義して、フォームの 2 つのインスタンスが同じビジットで同じキーアイテム値で実行された場合、InForm アプリケーションは 2 番目のインスタンスの入力を受け付けません。

単一のアイテム KEYITEMREF で UNIQUEKEY=true と設定した場合、そのアイテムはすべてのフォームインスタンスを通じて一意でなければなりません。

例

以下のフォーム定義の例は、VS フォームの VS セクションに、キーアイテムとして DATEASSESS アイテムを挿入する方法を示しています。

```
<SECTION REFNAME="VS" TITLE="Vital Signs">
  <ITEMREF REFNAME="DATEASSESS" ORDER="1"/>
  <ITEMREF REFNAME="WEIGHT" ORDER="2"/>
  <ITEMREF REFNAME="TEMPTEXT" ORDER="3"/>
  <ITEMREF REFNAME="BPREADING" ORDER="4"/>
  <ITEMREF REFNAME="PULSERATE" ORDER="5"/>
  <ITEMREF REFNAME="PULSERHYTHM" ORDER="6"/>
  <ITEMREF REFNAME="RESPRATE" ORDER="7"/>
</SECTION>

<FORM REFNAME="VS" TITLE="Vital Signs" MNEMONIC="VS"
  QUESTIONWIDTH="25">
  <SECTIONREF REFNAME="VS"/>
  <KEYITEMREF SECTIONREFNAME="VS" ITEMREFNAME="DATEASSESS"/>
</FORM>
```

MedMLData

目的

スタディ定義コンポーネントを定義する XML ファイル内の最初と最後に必要なエレメントです。他のすべてのエレメントを囲み、囲まれたエレメント群がスタディ定義コンポーネントで構成されていることを MedML インストーラユーティリティに伝えます。

構文

```
<MEDMLDATA xmlns="version_information">
  [xmlns="version_information"]
  <child_components+ attributes/>
</MEDMLDATA>
```

ファイルに含まれる XML のバージョン。MedML インストーラユーティリティで処理する XML には必須です。必要なバージョンテキストは、「PhaseForward-MedML-Inform4」です。

子

MedMLData エレメントの子コンポーネントは、スタディのコンポーネントを定義します。ビジット、フォーム、ルール、イベント、リソースおよびスタディ管理データを定義するためのコンポーネントがあります。

例

次の例は、DOV フォームを定義する MedML エレメントに対するラッパーとして MedMLData エレメントを使用する方法を示しています。

```
<MEDMLDATA xmlns="PhaseForward-MedML-Inform4">

  <DATETIMECONTROL REFNAME="DOV" CAPTIONALIGN="LEFT" ALIGN="LEFT"
    STARTYEAR="2002" ENDYEAR="2008" LANGUAGE="English"
    REQUIREMINUTE="FALSE" REQUIRESECOND="FALSE" REQUIREDAY="TRUE"
    UNKNOWNDAY="FALSE" DISPLAYYEAR="TRUE" REQUIREMONTH="TRUE"
    UNKNOWNYEAR="FALSE" UNKNOWNSECOND="FALSE" DISPLAYDAY="TRUE"
    REQUIREYEAR="TRUE" UNKNOWNMINUTE="FALSE" DISPLAYMINUTE="TRUE"
    DISPLAYSECOND="FALSE" REQUIREHOUR="FALSE"
    UUID="BD991BC0-B0A4-11D2-80E3-00A0C9AF7674"
    DISPLAYHOUR="TRUE"
    UNKNOWNMONTH="FALSE" UNKNOWNHOUR="FALSE"
    CHECKCONSISTENT="TRUE" DISPLAYMONTH="TRUE"/>

  <ITEM REFNAME="DOV" LANGUAGE="English"
    SDVREQUIRED="TRUE"
    QUESTION="Date and Time of Visit"
    UUID="BD991BBF-B0A4-11D2-80E3-00A0C9AF7674"
    ITEMREQUIRED="TRUE"
    CALCULATED="FALSE"
    LABEL="Date and Time of Visit">
    <CONTROLREF REFNAME="DOV" />
  </ITEM>
```

```

<SECTION REFNAME="DOV" REPEATING="FALSE"
  LANGUAGE="English"
  TITLE="Visit Date and Time"
  UUID="BD991BBE-B0A4-11D2-80E3- 00A0C9AF7674">
  <ITEMREF REFNAME="DOV" />
</SECTION>

<FORM REFNAME="frmDOV" LANGUAGE="English"
  TITLE="DATE OF VISIT"
  CONTROLWIDTH="65"
  TYPE="CRF"
  MNEMONIC="DOV"
  QUESTIONWIDTH="35">
  <SECTIONREF REFNAME="DOV" ORDER="1"/>
</FORM>

</MEDMLDATA>

```

PFElement

目的

ドロップダウンリスト内のアイテム、あるいはラジオボタンのリスト内のオプションのようなコンポーネントを定義します。PFElement は定義できる最低レベルの CRF コンポーネントです。PFElement エレメントは、ドロップダウンリストまたはシンプルコントロールなどの別のコントロールの定義に含める必要があります。

注： 他の **CRF** コンポーネントと異なり、**PFElement** 定義はデータベースに複数のバージョンを持つことができません。既存の **PFElement** と同じ **RefName** のままでコンポーネントの改訂バージョンをサブミットした場合、**MedML** インストラユーティリティは警告を発し、コンポーネントの定義を更新しません。

構文

```

<PFELEMENT
  REFNAME="name"
  [DESIGNNOTE="text"]
  LABEL="text"
  [LANGUAGE="name"]
  TYPE="STRING|INTEGER|FLOAT"
  VALUE="n"
  [UUID="id"]>
  <TRANSLATIONS/>
</PFELEMENT>

```


属性

REFNAME="*name*"

他のコントロール定義で PFElement を参照するときに使用される名前。この名前は PFElement 間で一意である必要があります。RefName は、最大 63 文字です。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

LABEL="*text*"

PFElement が表示されるテキスト、すなわちリストに表示する方法。ラベルは最大 255 文字です。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。必須

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エlement がスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

TYPE="STRING|INTEGER|FLOAT"

Element のデータタイプ。STRING (文字列)、INTEGER (整数)、もしくは FLOAT (浮動小数点数)。デフォルトは STRING (文字列) です。必須

注： 1 つの複数選択コントロール (チェックボックスコントロールなど) で使用される PFElements はすべて、同じ **TYPE** 属性を持っている必要があります。

VALUE="*n*"

データベースでコンポーネントが選択されたときに返される値。RefName やラベルと異なる値でも構いません。

UUID="*id*"

Universally Unique Identifier すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列です。オプション

子

スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エlement には 1 つ以上の Translation 定義を含める必要があります。Translation エlement はそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。PFElement 定義内の **LABEL** 属性は翻訳できます。

例

以下のElementは、後でコンポーネントとしてリスト内で使用することができる 2 つの PFElement を定義しています。さらに、この例では **LABEL** 属性が翻訳されています。

```
<PFELEMENT REFNAME="Y" LABEL="Yes" LANGUAGE="en-US"
  TYPE="STRING" VALUE="Y">
  <TRANSLATIONS>
    <TRANSLATION NAME="LABEL" LOCALE="fr-FR" DISPLAYTEXT="Oui"/>
  </TRANSLATIONS>
</PFELEMENT>
<PFELEMENT REFNAME="N" LABEL="No" LANGUAGE="en-US"
  TYPE="STRING" VALUE="N">
  <TRANSLATIONS>
    <TRANSLATION NAME="LABEL" LOCALE="fr-FR" DISPLAYTEXT="Non"/>
  </TRANSLATIONS>
</PFELEMENT>
```

PulldownControl

目的

ユーザがアイテムを 1 つ選択できるドロップダウンリストを定義します。ドロップダウンリストは、Elementref を使用して挿入される定義済みの PFElement から構成されます。さらに、ドロップダウンリストには、Unitref を使用して挿入される定義済みの単位を含めることもできます。

構文

```
<PULLDOWNCONTROL
  REFNAME="name"
  [DESIGNNOTE="text"]
  [NAME="name"]
  [UUID="id"]
  [ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"]
  [CAPTION="text"]
  [LANGUAGE="name"]
  [CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"]
  [UNITDISPLAYTYPE="ELEMENT"]>
  <ELEMENTREF+ attributes>
  <UNITREF attributes>
  <TRANSLATIONS/>
```

</PULLDOWNCONTROL>

属性**REFNAME**="name"

他のコントロール定義でドロップダウンリストを参照するときに使用される名前。この名前は PullDownControl 間で一意である必要があります。必須

DESIGNNOTE="text"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

NAME="name"

他のコントロール定義で PullDownControl を参照するときに使用される名前。オプション

UUID="id"

Universally Unique Identifier; すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"

コントロール内の PFElement の位置合わせ。LEFT（左）、CENTER（中央）、RIGHT（右）、TOP（上）、MIDDLE（中央）、もしくは BOTTOM（下）。デフォルトは LEFT（左）です。オプション

CAPTION="text"

ドロップダウンリストと共に画面上に表示されるテキスト。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

LANGUAGE="name"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合（**TRANSLATIONS** エlementがスタディコンポーネント定義のチャイルドとして含まれている場合）は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"

キャプションの計算値との相対位置。LEFT（左）、RIGHT（右）、TOP（上）、もしくは BOTTOM（下）。デフォルトは LEFT（左）です。オプション

UNITDISPLAYTYPE="ELEMENT"

Unitref 定義を持つ PullDownControl に含まれる単位を表示するために使用されるコントロールのタイプ。「ELEMENT」のみが有効な値です。

子

- 1 つ以上の Elementref 定義。各 Elementref は、ドロップダウンリスト中の 1 つのアイテムを特定する、事前定義済み PFElement を参照します。
- 事前定義済みの単位を参照する 0 または 1 個以上の Unitref 定義。

注: Elementref 定義または Unitref 定義でコンパウンドコントロールを定義する場合は、すべての下位コントロールについて同じ **TYPE** 属性を定義することで、これらが同じデータタイプを返すようにします。さらに、コンパウンドコントロールを定義するとき、InForm ソフトウェア は最大 5 層のネスティングをサポートします。5 層での設計が可能です。パフォーマンスを向上させるためにネストレベルはできるだけ少なくしてください。

- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エlement には 1 つ以上の Translation 定義を含める必要があります。Translation エlement はそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エlement は、スタディコンポーネント定義内の最後のチャイルドElement にする必要があります。PullDownControl 定義の **CAPTION** 属性は翻訳できます。

例

この例の PullDownControl は、薬物治療の理由のリストになっています。さらに、この例では **CAPTION** 属性が翻訳されています。

リストの作成：

- 1 各薬物治療の理由に PFElement を定義してください。

```
<PFELEMENT REFNAME="ADVERSE" LABEL="Adverse Experience" TYPE="INTEGER" VALUE="1"/>
<PFELEMENT REFNAME="CHRONIC" LABEL="Chronic Illness" TYPE="INTEGER" VALUE="2"/>
<PFELEMENT REFNAME="HORMONE" LABEL="Hormone Replacement" TYPE="INTEGER" VALUE="3"/>
<PFELEMENT REFNAME="PROPHYL" LABEL="Prophylaxis" TYPE="INTEGER" VALUE="4"/>
```

- 2 Elementref を使用して、各 PFElement 定義を含むプルダウンコントロールを定義します。

```
<PULLDOWNCONTROL REFNAME="MEDREASON"
  NAME="MEDREASON" LANGUAGE="en-US" CAPTION="Reason">
  <ELEMENTREF REFNAME="ADVERSE" ORDER="1"/>
  <ELEMENTREF REFNAME="CHRONIC" ORDER="2"/>
  <ELEMENTREF REFNAME="HORMONE" ORDER="3"/>
  <ELEMENTREF REFNAME="PROPHYL" ORDER="4"/>
  <TRANSLATIONS>
    <TRANSLATION NAME="CAPTION" LOCALE="sy-SE" DISPLAYTEXT="Anledning"/>
    <TRANSLATION NAME="CAPTION" LOCALE="nn-NO" DISPLAYTEXT="Grunnen"/>
  </TRANSLATIONS>
</PULLDOWNCONTROL>
```

QueryGroup

目的

同じグループ内の他のメンバによって開始されたクエリをクローズすることができるユーザを指定します。

注： ユーザインターフェースではクエリグループを作成できません。クエリグループを定義するには **MedML** を使用する必要があります。

注： グループのユーザは、**InForm** アプリケーションを通じてのみ削除できます。**MedML** インストーラユーティリティの実行によって、ユーザを削除することはできません。

構文

```
<QUERYGROUP
  GROUPNAME="name"
  [GROUPDESCRIPTION="text"]
  [UUID="id"]>
  <USERREF* attributes/>
</QUERYGROUP>
```

属性

GROUPNAME="*name*"

QueryGroup の名前。必須

GROUPDESCRIPTION="*text*"

QueryGroup を説明するテキスト。オプション

UUID="*id*"

Universally Unique Identifier すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列です。オプション

子

クエリグループ定義は 0 以上の Userref 定義を含むことが可能です。各 Userref は、1 人の InForm ユーザを特定する作成済の User 定義を参照します。

例

以下のクエリグループで、Kevin と G e o r g e というユーザは、もう片方のユーザによって開かれたクエリを閉じることができます。

```
<QUERYGROUP GROUPNAME="CRA Query">
  <USERREF USERNAME="Kevin"/>
  <USERREF USERNAME="George"/>
</QUERYGROUP>
```

RadioControl

目的

ユーザが 1 つのオプションを選択する必要があるラジオボタンのリストを定義します。ラジオコントロールは、Controlref を使用して挿入される事前定義済みコンポーネントにより構成されます。RadioControl 定義には以下のタイプのコンポーネントを含めます。

- CalculatedControl
- CheckBoxControl
- GroupControl
- PullDownControl
- RadioControl
- SimpleControl
- TextControl

構文

```
<RADIOCONTROL
  REFNAME="name"
  [DESIGNNOTE="text"]
  [NAME="name"]
  [UUID="id"]
  [ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"]
  [LAYOUT="VERTICAL|HORIZONTAL|NOWRAP"]
  [CAPTION="text"]
  [LANGUAGE="name"]
  [CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"]>
  <CONTROLREF+ attributes/>
  <TRANSLATIONS/>
</RADIOCONTROL>
```

属性

REFNAME="*name*"

他のコントロール定義で **RadioControl** を参照するときに使用される名前。この名前はラジオコントロール間で一意である必要があります。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

NAME="*name*"

他のコントロール定義内でラジオコントロールを参照するときに使用される名前。オプション

UUID="*id*"

Universally Unique Identifier すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列です。オプション

ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"

ラジオコントロール内に **Controlref** を使用して挿入されるコンポーネントの位置合わせ。LEFT (左)、CENTER (中央)、RIGHT (右)、TOP (上)、MIDDLE (中央)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

LAYOUT="VERTICAL|HORIZONTAL|NOWRAP"

ラジオボタンの向き。オプションは、VERTICAL (垂直)、HORIZONTAL (水平)、もしくは NOWRAP (ノーラップ) です。デフォルトは NOWRAP (ノーラップ) です。NOWRAP (ノーラップ) を指定すると、ラジオボタンは水平方向に位置づけられ、ユーザがブラウザウィンドウのサイズを変更しても、ボタンは別の行に改行されません。オプション

CAPTION="*text*"

ラジオボタンリストと共に画面に表示されるテキスト。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能のため、デフォルトプロダクトロケールに依存することは推奨されません。

CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"

キャプションのラジオコントロールとの相対位置。LEFT (左)、RIGHT (右)、TOP (上)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

子

- 1 つ以上の Controlref 定義。各 Controlref は、定義済コンポーネントを照会しラジオボタンのリストの中から 1 つのアイテムを識別します。

注： **Controlref** 定義を用いて複雑なコントロールを定義する際には、すべての下位コントロールに同じ **TYPE** 属性を定義することによって、それらが同じデータタイプを返すようにしてください。さらに、コンパウンドコントロールを定義するとき、**InForm** ソフトウェア は最大 **5** 層のネスティングをサポートしています。**5** 層での設計が可能です。パフォーマンスを向上させるためにネストレベルはできるだけ少なくしてください。

- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エlement には 1 つ以上の Translation 定義を含める必要があります。Translation エlement はそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エlement は、スタディコンポーネント定義内の最後のチャイルドElement にする必要があります。RadioControl 定義の **CAPTION** 属性は翻訳できます。

例 1

この例は引き続き SimpleControl の例です。性別を指定するために PFElements セットを定義し、SimpleControl でこれらを折り返す方法を実証します。この例は、Controlref を使用して、ラジオコントロールに定義済みのシンプルコントロールを含める方法を示します。さらに、この例では **CAPTION** 属性の翻訳の定義も示しています。

- 1 男性/女性の選択を定義する PFElement を作成します。

```
<PFELEMENT REFNAME="MALE" LABEL="Male" TYPE="STRING" VALUE="MElement"/>
<PFELEMENT REFNAME="FEMALE" LABEL="Female" TYPE="STRING" VALUE="FElement"/>
```

- 2 Elementref を使用して各 PFElement の定義を含める SimpleControls を作成します。

```
<SIMPLECONTROL REFNAME="MALE" NAME="MALE">
  <ELEMENTREF REFNAME="MALE"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="FEMALE" NAME="FEMALE">
  <ELEMENTREF REFNAME="FEMALE"/>
</SIMPLECONTROL>
```

- 3 Controlref を使用して各 SimpleControl を挿入し、RadioControl を作成します。

```
<RADIOCONTROL REFNAME="GENDER"
  NAME="GENDERRADIO"
  LANGUAGE="en-US"
  CAPTION="Select one:"
  LAYOUT="HORIZONTAL">
  <TRANSLATIONS>
    <TRANSLATION NAME="CAPTION" LOCALE="pt-PT" DISPLAYTEXT="Selecione um"/>
  </TRANSLATIONS>
  <CONTROLREF REFNAME="MALE" ORDER="1"/>
  <CONTROLREF REFNAME="FEMALE" ORDER="2"/>
</RADIOCONTROL>
```

例 2

この例はラジオボタンのリストを作成するプロセスを示しています。このラジオボタンリストにはいくつかの異なったタイプのコントロール、シンプルコントロール、ネスト

化されたコントロール等が含まれています。以下の例では、スクリーニングされたスタディ症例がスタディを完了しなかった理由を示すラジオボタンリストを定義しています。

リストのコンポーネント：

- このリスト内の次のアイテムは、単純に PFElement として定義した後で、シンプルコントロールの定義に含めます。
 - 無作為割付の基準を満たさなかった
 - 有害事象（臨検的な問題を除く）
 - 臨検的な問題
 - フォローアップ失敗
- Other (specify) (その他（詳細を記述））アイテムはキャプション付きシンプルテキストボックスです。
- 一部のアイテムについては、キャプション付きテキストボックスと追加キャプションを結合します。
 - 症例による判断
 - 医師による判断
 - スポンサーからの要求

これらのアイテムについては、PFElement としてラジオボタンの横に表示されるキャプションを定義し、シンプルコントロールに挿入します。テキストコントロールとしてテキストボックスとそのキャプションを定義します。そして、2 つのコンポーネントを組み合わせるグループコンポーネントを作成します。

- Death (死亡) アイテムは、ラジオボタンの横に表示されるキャプションと、死亡日を指定する 3 つのドロップダウンリストのグループから成ります。このアイテムについて、PFElement としてラジオボタンの横に表示されるキャプションを定義し、シンプルコントロールに挿入します。3 つの日付コンポーネントを日時コントロールとして定義し、グループコントロールに、この日時コントロールを挿入します。最後に、キャプション用のシンプルコントロールと日付用のグループコントロールを含むグループコントロールを定義します。

定義ステップ：

- 1 必要とする PFElements を定義し、SimpleControl 定義でこれらを囲みます。

```
<PFELEMENT REFNAME="CRITERIA" LABEL="Did not meet criteria" TYPE="STRING"
VALUE="CriteriaElement"/>
<PFELEMENT REFNAME="AE" LABEL="Adverse Experience (other than adverse laboratory
experience)" TYPE="STRING" VALUE="AEElement"/>
<PFELEMENT REFNAME="ALE" LABEL="Adverse laboratory experience" TYPE="STRING"
VALUE="ALEElement"/>
<PFELEMENT REFNAME="PD" LABEL="Patient decision" TYPE="STRING"
VALUE="PDElement"/>
<PFELEMENT REFNAME="PHD" LABEL="Physician decision" TYPE="STRING"
VALUE="PHDElement"/>
<PFELEMENT REFNAME="SD" LABEL="Sponsor decision" TYPE="STRING"
VALUE="SDElement"/>
<PFELEMENT REFNAME="DEATHSC" LABEL="Death (Sponsor must be notified and Serious
Adverse Experience Report completed.)" TYPE="STRING" VALUE="DeathElement"/>
<PFELEMENT REFNAME="LOST" LABEL="Lost to follow-up" TYPE="STRING"
VALUE="LostElement"/>
<PFELEMENT REFNAME="OTHER" LABEL="Other (specify):" TYPE="STRING"
```

```

VALUE="OtherElement"/> <SIMPLECONTROL REFNAME="CRITERIA"
NAME="CriteriaElement">
  <ELEMENTREF REFNAME="CRITERIA"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="AE" NAME="AEElement">
  <ELEMENTREF REFNAME="AE"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="ALE" NAME="ALEElement">
  <ELEMENTREF REFNAME="ALE"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="PD" NAME="PDElement">
  <ELEMENTREF REFNAME="PD"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="PHD" NAME="PHDElement">
  <ELEMENTREF REFNAME="PHD"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="SD" NAME="AEElement">
  <ELEMENTREF REFNAME="SD"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="DEATHSC" NAME="DeathElement">
  <ELEMENTREF REFNAME="DEATHSC"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="LOST" NAME="LostElement">
  <ELEMENTREF REFNAME="LOST"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="OTHER" NAME="OtherElement">
  <ELEMENTREF REFNAME="OTHER"/>
</SIMPLECONTROL>

```

- 2 Other (specify) (その他 (詳細を記述)) アイテムと Reason (理由) アイテム用にテキストコントロールを定義します。スペーシングを調整するための HTML 特殊文字 ** ** の使用に注意してください。

[illegible]

- 3 症例による判断、医師による判断、およびスポンサーからの要求用の各テキストボックスと対応するキャプションのグループコントロールを定義します。

```
<GROUPCONTROL REFNAME="CRITERIAGROUP" UUID="PF_SC_CRITERIA"
  NAME="CRITERIAGROUP" LAYOUT="HORIZONTAL">
  <CONTROLREF REFNAME="CRITERIA" ORDER="1"/>
</GROUPCONTROL>
<GROUPCONTROL REFNAME="AEGROUP" UUID="PF_SC_AE"
  NAME="AEGROUP" LAYOUT="HORIZONTAL">
  <CONTROLREF REFNAME="AE" ORDER="1"/>
</GROUPCONTROL>
<GROUPCONTROL REFNAME="ALEGROUP" UUID="PF_SC_ALE"
  NAME="ALEGROUP" LAYOUT="HORIZONTAL">
  <CONTROLREF REFNAME="ALE" ORDER="1"/>
</GROUPCONTROL>
<GROUPCONTROL REFNAME="PDECGRUP" UUID="PF_SC_PATIENTDECISION"
  NAME="PDECGRUP" LAYOUT="HORIZONTAL"
  ALIGN="LEFT">
  <CONTROLREF REFNAME="PD" ORDER="1"/>
  <CONTROLREF REFNAME="REASONTEXT" ORDER="2"/>
</GROUPCONTROL>
<GROUPCONTROL REFNAME="PHDECGRUP" UUID="PF_SC_PHYSICIANDECISION"
  NAME="PHDECGRUP" LAYOUT="HORIZONTAL"
  ALIGN="LEFT">
  <CONTROLREF REFNAME="PHD" ORDER="1"/>
  <CONTROLREF REFNAME="REASONTEXT" ORDER="2"/>
</GROUPCONTROL>
<GROUPCONTROL REFNAME="SDECGRUP" UUID="PF_SC_SPONSORDECISION"
  NAME="SDECGRUP" LAYOUT="HORIZONTAL">
```

```

        ALIGN="LEFT">
        <CONTROLREF REFNAME="SD" ORDER="1"/>
        <CONTROLREF REFNAME="REASONTEXT" ORDER="2"/>
    </GROUPCONTROL>
    <GROUPCONTROL REFNAME="OTHERGROUP" UUID="PF_SC_OTHER"
        NAME="OTHERGROUP" LAYOUT="HORIZONTAL">
        <CONTROLREF REFNAME="OTHER" ORDER="1"/>
        <CONTROLREF REFNAME="REASONTEXT" ORDER="2"/>
    </GROUPCONTROL>
    <GROUPCONTROL REFNAME="LOSTGROUP" UUID="PF_SC_LOST"
        NAME="DEATHGROUP" LAYOUT="HORIZONTAL">
        <CONTROLREF REFNAME="LOST" ORDER="1"/>
    </GROUPCONTROL>

```

- 4 Death (死亡) アイテム用に日時コントロールを定義します。

```

<DATETIMECONTROL REFNAME="COMMONDATE" NAME="COMMONDATE"
    STARTYEAR="1997" ENDYEAR="2002"
    DISPLAYDAY="true" DISPLAYMONTH="true"
    DISPLAYYEAR="true" REQUIREMONTH="true"
    REQUIREDAY="true" REQUIREYEAR="true"/>

```

- 5 DEATHDATE グループコントロール、および Death キャプションを定義するシンプルコントロールを含むグループコントロールを定義します。

```

<GROUPCONTROL REFNAME="DEATHGROUP" UUID="PF_SC_DEATH"
    NAME="DEATHGROUP" LAYOUT="HORIZONTAL">
    <CONTROLREF REFNAME="DEATHSC" ORDER="1"/>
    <CONTROLREF REFNAME="COMMONDATE" ORDER="2"/>
</GROUPCONTROL>

```

- 6 リスト内の各シンプルコントロールまたはコンパウンドコントロールを表す Controlref を含むラジオコントロールを定義します。

```

<RADIOCONTROL REFNAME="REASONRADIO"
    NAME="REASONRADIO" LAYOUT="VERTICAL"
    UUID="PF_SC_REASONCTL">
    <CONTROLREF REFNAME="CRITERIAGROUP" ORDER="1"/>
    <CONTROLREF REFNAME="AEGROUP" ORDER="2"/>
    <CONTROLREF REFNAME="ALEGROUP" ORDER="3"/>
    <CONTROLREF REFNAME="PDECGROUP" ORDER="4"/>
    <CONTROLREF REFNAME="PHDECGROUP" ORDER="5"/>
    <CONTROLREF REFNAME="SDECGROUP" ORDER="6"/>
    <CONTROLREF REFNAME="DEATHGROUP" ORDER="7"/>
    <CONTROLREF REFNAME="LOSTGROUP" ORDER="8"/>
    <CONTROLREF REFNAME="OTHERGROUP" ORDER="9"/>
</RADIOCONTROL>

```

ReportingGroup

目的

注： ユーザインターフェースではレポーティンググループを作成できません。レポーティンググループを定義するには **MedML** を使用する必要があります。

レポート権限を持つユーザが使用できるレポーティング機能とアクセスタイプを定義します。メンバが標準レポートのみにアクセスできるレポーティンググループもあれば、InForm アドホック レポートワークスペースにアクセスできるものもあります。

注： ユーザをグループから削除するには、InForm アプリケーションの **Admin** 機能を通して行う方法しかないことに注意してください。MedML インストーラユーティリティの実行によって、ユーザを削除することはできません。

構文

```
<REPORTINGGROUP
  GROUPNAME="name"
  GROUPDESCRIPTION="name">
  <USERREF USERNAME="name" />
</REPORTINGGROUP>
```

属性

GROUPNAME="*name*"

レポートिंगグループの名前。必須

GROUPDESCRIPTION="*name*"

レポートिंगグループの説明。オプション

子

ReportingGroup 定義は 0 以上の Userref 定義を含むことが可能です。各 Userref は、1 人の InForm ユーザを特定する作成済の User 定義を参照します。

例

以下の例は、アドホックユーザという名前のレポートिंगグループを作成するための構文を示しています。

```
<REPORTINGGROUP
  GROUPNAME="Ad Hoc Users"
  GROUPDESCRIPTION="Ad Hoc Users">
  <USERREF USERNAME="cra" />
</REPORTINGGROUP>
```

Resource

目的

ロードする単一のリソースを識別します。

構文

```
<RESOURCE
  [UUID="id"]
  [FILENAME="file"]
  [DESCRIPTION="text"]
  DATATYPE="TEXT|GIF|JPEG"
  [LANGUAGE="name"]
  [TEXT="text"]
  [DATA="data"]/>
```

属性

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列。**TEXT** 属性または **DATA** 属性に明示的なテキストかデータストリームが含まれない場合、**UUID** 属性あるいは **FILENAME** 属性のいずれかが必要です。

FILENAME="*file*"

リソースが含まれるファイルの名前。**UUID** 属性または **FILENAME** 属性のいずれかが必要です。

DESCRIPTION="*text*"

リソースの説明。オプション

DATATYPE="TEXT|GIF|JPEG"

ファイルに含まれているデータタイプ: TEXT、GIF あるいは JPEG。必須

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注: デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

TEXT="*text*"

DATATYPE が **TEXT** であるリソースの実際のテキスト。実際のテキストを含む場合は、**UUID** 属性または **FILENAME** 属性を指定しないでください。

DATA="*data*"

DATATYPE が GIF または JPEG であるリソースを定義するデータストリーム。

DATATYPE が GIF または JPEG である場合、**UUID**、**FILENAME** あるいは **DATA** 定義が必要です。データストリームを含む場合は、**UUID** 属性または **FILENAME** 属性は指定しないでください。

例 1

以下の例は 2 つのテキストリソースを示しています。

```
<RESOURCE FILENAME="PAGE.HTML"
  UUID="b69ff18f=e466-11d1-9e5c-00a0c9769a33"
  DESCRIPTION="PAGE.HTML"
  DATATYPE="TEXT"/>

<RESOURCE FILENAME="FORM.HTML"
  UUID="b69ff190=e466-11d1-9e5c-00a0c9769a33"
  DESCRIPTION="FORM.HTML"
  DATATYPE="TEXT"/>
```

例 2

以下の例は 3 つの画像リソースを示しています。

```
<RESOURCE FILENAME="SideArrow.GIF"
  UUID="56fc2652-ee9f-11d1-a744-00a0c9af7673"
  DESCRIPTION="SideArrow.GIF"
  DATATYPE="GIF"/>

<RESOURCE FILENAME="SideDocsGray.GIF"
  UUID="573c85d0-ee9f-11d1-a744-00a0c9af7673"
  DESCRIPTION="SideDocsGray.GIF"
  DATATYPE="GIF"/>

<RESOURCE FILENAME="SideDocsYellow.GIF"
  UUID="577a82f4-ee9f-11d1-a744-00a0c9af7673"
  DESCRIPTION="SideDocsYellow.GIF"
  DATATYPE="GIF"/>
```

ReviewStage

目的

データビューアで使用するカスタムレビューステータスに関連付けられるカスタムレビューステージを作成します。ReviewState エlementと共に使用して、カスタムレビューステージをその親カスタムレビューステータスに関連付けます。レビュー

ステータスごとに 3 つのステージを作成する必要があります。

たとえば、依頼者のレビューが必要なデータを追跡するには、「**レビューが必要**」、「**レビュー待ち**」、および「**レビュー完了**」ステージを含む「**レビューステータス 1**」というレビューステータスを作成できます。各ステータスのデータアイテムの数を表示するには、データビューアに移動します。

備考：

カスタムレビューステータスとカスタムレビューステージを定義するには、**MedML** と **MedML** インストラユーティリティを使用する必要があります。**InForm** ユーザインターフェイスではこれらを作成できません。

一度に **5** つのすべてのカスタムレビューステータスを定義する必要はありません。ただし、定義するカスタムレビューステータスごとに、一度に **3** つのすべての関連カスタムレビューステージを定義する必要があります。

構文

```
<REVIEWSTAGE
  REFNAME="name"
  [STAGE="n"]
  [LABEL="name"]
  [MNEMONIC="name"]
  [LANGUAGE="name"]
  <TRANSLATIONS/>
</REVIEWSTAGE>
```

属性

REFNAME="*name*"

カスタムレビューステージの RefName。必須

STAGE="*n*"

1 ～ 3 の数値。InForm アプリケーションで親カスタムレビューステータスのカラム内にカスタムレビューステージを表示する順序を示します。カスタムレビューステータスごとに 3 つのカスタムレビューステージを指定する必要があります。必須

LABEL="*name*"

InForm アプリケーションでホバーヘルプとドロップダウンリストに表示されるカスタムレビューステージの名前。必須

MNEMONIC="*name*"

Data Viewer のカラム見出しに表示されるカスタムレビューステージの略称。必須

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。必須

注：デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

例

```
<REVIEWSTAGE REFNAME="ReviewStage1" STAGE="0" LABEL="Needs Review" MNEMONIC="RS1"
LANGUAGE="en-US">
  <TRANSLATIONS>
    <TRANSLATION NAME="LABEL" DISPLAYTEXT="{f@frf..._[" LOCALE="ja-JP" />
  <TRANSLATION NAME="MNEMONIC" DISPLAYTEXT="{f@frf..._[S1" LOCALE="ja-JP" />
</TRANSLATIONS>
</REVIEWSTAGE>
<REVIEWSTAGE REFNAME="ReviewStage2" STAGE="1" LABEL="Pending" MNEMONIC="RS2"
LANGUAGE="en-US">
  <TRANSLATIONS>
    <TRANSLATION NAME="LABEL" DISPLAYTEXT="•—" LOCALE="ja-JP" />
  <TRANSLATION NAME="MNEMONIC" DISPLAYTEXT="{f@frf..._[S2" LOCALE="ja-JP" />
</TRANSLATIONS>
</REVIEWSTAGE>
<REVIEWSTAGE REFNAME="ReviewStage3" STAGE="2" LABEL="Reviewed" MNEMONIC="RS3"
LANGUAGE="en-US">
  <TRANSLATIONS>
    <TRANSLATION NAME="LABEL" DISPLAYTEXT="{f@frf..._[_i" LOCALE="ja-JP" />
  <TRANSLATION NAME="MNEMONIC" DISPLAYTEXT="{f@frf..._[S3" LOCALE="ja-JP" />
</TRANSLATIONS>
</REVIEWSTAGE>
```

ReviewState

目的

データビューアで使用するレビューステータスを作成します。レビューステータスを

使用すると、データのレビュープロセスを柔軟に追跡できます。最大 5 つのレビューステータスを作成し、必要に応じてカスタマイズできます。各レビューステータスには、カスタマイズ可能な 3 つのステージが含まれます。

たとえば、依頼者のレビューが必要なデータを追跡するには、「**レビューが必要**」、「**レビュー待ち**」、および「**レビュー完了**」ステージを含む「**レビューステータス 1**」というレビューステータスを作成できます。各ステータスのデータアイテムの数を表示するには、データビューアに移動します。

備考：

カスタムレビューステータスとカスタムレビューステージを定義するには、**MedML** と **MedML** インストラユーティリティを使用する必要があります。**InForm** ユーザインターフェイスではこれらを作成できません。

一度に **5** つのすべてのカスタムレビューステータスを定義する必要はありません。ただし、定義するカスタムレビューステータスごとに、一度に **3** つのすべての関連カスタムレビューステージを定義する必要があります。

構文

```
<REVIEWSTATE
  REFNAME="name"
  STATE="n"
  ACTIVATED="true|false"
  LABEL="name"
  MNEMONIC="name"
  LANGUAGE="name">
  <TRANSLATIONS/>
</REVIEWSTATE>
```

属性

REFNAME="*name*"

レビューステータスの RefName。必須

STATE="*n*"

1 ～ 5 の数値。ユーザインターフェイスにレビューステータスを表示する順序を示します。必須

ACTIVATED="*true|false*"

true または false。レビューステータスを InForm アプリケーションで使えるようにするかどうかを示します。必須

LABEL="*name*"

InForm アプリケーションでホバーヘルプとドロップダウンリストに表示されるカスタムレビューステージの名前。必須

MNEMONIC="*name*"

Data Viewer のカラム見出しに表示されるカスタムレビューステージの略称。必須

MNEMONIC="*name*"

レビューステータスの略称。必須

LANGUAGE="*name*"テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エlement がスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。必須

注: デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

子

- レビューステータスのステージを定義する 3 つの ReviewStage Element。たとえば、各レビューステータスには次のレビューステージを含めることができます。
 - レビューが必要
 - レビュー待ち
 - レビュー完了

例

```
<REVIEWSTATE REFNAME="DataReview1" STATE="1" ACTIVATED="true" LABEL="Data Review"
MNEMONIC="R1" LANGUAGE="en-US">
  <REVIEWSTAGE REFNAME="ReviewStage1" STAGE="0" LABEL="Needs Review" MNEMONIC="RS1"
  LANGUAGE="en-US">
    <TRANSLATIONS>
      <TRANSLATION NAME="LABEL" DISPLAYTEXT="-vf@frf..._[" LOCALE="ja-JP" />
      <TRANSLATION NAME="MNEMONIC" DISPLAYTEXT="f@frf..._[" LOCALE="ja-JP" />
    </TRANSLATIONS>
  </REVIEWSTAGE>
</REVIEWSTATE>
```

```

</TRANSLATIONS>
  </REVIEWSTAGE>
  <REVIEWSTAGE REFNAME="ReviewStage2" STAGE="1" LABEL="Pending" MNEMONIC="RS2"
    LANGUAGE="en-US">
    <TRANSLATIONS>
      <TRANSLATION NAME="LABEL" DISPLAYTEXT="•0—" LOCALE="ja-JP" />
    <TRANSLATION NAME="MNEMONIC" DISPLAYTEXT="f@frf..._S2" LOCALE="ja-JP" />
  </TRANSLATIONS>
  </REVIEWSTAGE>
  <REVIEWSTAGE REFNAME="ReviewStage3" STAGE="2" LABEL="Reviewed" MNEMONIC="RS3"
    LANGUAGE="en-US">
    <TRANSLATIONS>
      <TRANSLATION NAME="LABEL" DISPLAYTEXT="f@frf..._i" LOCALE="ja-JP" />
      <TRANSLATION NAME="MNEMONIC" DISPLAYTEXT="f@frf..._S3" LOCALE="ja-JP" />
    </TRANSLATIONS>
  </REVIEWSTAGE>
</TRANSLATIONS>
<TRANSLATION NAME="LABEL" DISPLAYTEXT="ff [f^f@frf... [" LOCALE="ja-JP" />
<TRANSLATION NAME="MNEMONIC" DISPLAYTEXT="f@frf..._l" LOCALE="ja-JP" />
</TRANSLATIONS>
</REVIEWSTATE>

```

Rightref

目的

権限グループの定義に、事前に定義された権限を含めることができます。Rightref は、自身が含まれている権限グループの子としてのみ表示され、単独コンポーネントとしてサブミットされることはありません。

構文

```

<RIGHTREF
  RIGHT="name"/>

```

属性

RIGHT="name"/>

Rightref によって参照される権限の名前。これは、Right 定義の Right 属性で特定された名前と同様です。

例

PF Admin 権限グループの以下の定義 (スタディ管理者が実行できるアクティビティを識別する) は、Rightref を使用して権限グループに権限定義を含める方法を示しています。

```

<RIGHTSGROUP GROUPNAME="PF Admin Rights Group">
  <RIGHTREF RIGHT="Activate Site User"/>
  <RIGHTREF RIGHT="Deactivate Site User"/>
  <RIGHTREF RIGHT="Activate Sponsor User"/>
  <RIGHTREF RIGHT="DeActivate Sponsor User"/>
  <RIGHTREF RIGHT="Make user active"/>
  <RIGHTREF RIGHT="Create Sites"/>
  <USERREF USERNAME="Kevin"/>
</RIGHTSGROUP>

```

RightsGroup

目的

権限のセットを作成し、ユーザに権限セットを割り当てることができます。InForm ソフトウェア には、事前に定義された権限と権限グループのセットが含まれています。

構文

```
<RIGHTSGROUP
  GROUPNAME="name"
  [GROUPDESCRIPTION="text"]
  [UUID="id"]>
  <RIGHTREF* attributes/>
  <USERREF* attributes/>
  <ITEMGROUPREF* attributes/>
</RIGHTSGROUP/>
```

属性

GROUPNAME="*name*"

権限グループの名前。必須

GROUPDESCRIPTION="*text*"

権限グループの説明。オプション

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

子

RightsGroup 定義には 0 または 1 個以上の次の定義を含めることができます。

- Rightref 定義。各 Rightref は、ユーザが実行することができる InForm 機能を識別する事前作成済みの権利定義を参照します。
- Userref 定義。各 Userref は、1 人の InForm ユーザを特定する事前作成済みのユーザ定義を参照します。
- ItemGroupref 定義。各 ItemGroupref は、ユーザが権限グループ固有の優先表示を設定できるアイテムのグループを識別する、事前作成済みアイテムグループの定義を参照します。

例

以下の例では、スタディ管理者が実行できるアクティビティを識別する PF Admin 権限グループの定義を示しています。また、この定義により、管理者権限セットが admin ユーザに割り当てられます。

```
<RIGHTSGROUP GROUPNAME="PF Admin Rights Group">
```

```

<RIGHTREF RIGHT="Activate Site User"/>
<RIGHTREF RIGHT="Deactivate Site User"/>
<RIGHTREF RIGHT="Activate Sponser User"/>
<RIGHTREF RIGHT="DeActivate Sponser User"/>
<RIGHTREF RIGHT="Make user active"/>
<RIGHTREF RIGHT="Create Sites"/>
<USERREF USERNAME="admin"/>
</RIGHTSGROUP>

```

以下の例では、CRC RG 権限グループは、CRC ユーザと CRC_Hidden ItemGroupref の定義を含んでいます。

```

<RIGHTSGROUP GROUPNAME="CRC RG">
  <RIGHTREF RIGHT="Print" />
  <RIGHTREF RIGHT="Reports" />
  <RIGHTREF RIGHT="Enroll Patients" />
  <RIGHTREF RIGHT="View CRF" />
  <RIGHTREF RIGHT="Enter Data into a CRF" />
  <RIGHTREF RIGHT="Edit Data on a CRF" />
  <RIGHTREF RIGHT="Enter Comments into a CRF" />
  <RIGHTREF RIGHT="Mark a CRF as Ready for SV" />
  <RIGHTREF RIGHT="Mark and unmark a CRB as Ready for SV" />
  <RIGHTREF RIGHT="Answer Query" />
  <RIGHTREF RIGHT="View Connections" />
  <RIGHTREF RIGHT="View Default Connection" />
  <RIGHTREF RIGHT="Synchronize New Data" />
  <USERREF USERNAME="crc" />
  <ITEMGROUPREF REFNAME="CRC_Hidden" DISPLAYOVERRIDE="HIDDEN"/>
</RIGHTSGROUP>

```

Right

目的

ユーザが実行できる特定の InForm アクティビティを識別する権限の定義をロードします。InForm ソフトウェア には事前に定義された権限が含まれています。

構文

```
<RIGHT
  RIGHT="name" />
```

属性

RIGHT="name"/>

権限の名前。

例

以下の例には、InForm アプリケーションにあらかじめ定義されている権限のうちのいくつかの定義が含まれています。

```
<RIGHT RIGHT="Create User Right" />
<RIGHT RIGHT="Modify User Rights" />
<RIGHT RIGHT="Activate Site User" />
<RIGHT RIGHT="Deactivate Site User" />
<RIGHT RIGHT="Activate Sponsor User" />
<RIGHT RIGHT="DeActivate Sponsor User" />
<RIGHT RIGHT="Make user active" />
<RIGHT RIGHT="Modify System Configuration" />
```

Rule

目的

ルールと実行スクリプトを定義し、イベントとルールとの関連付けに使用します。

構文

```
<RULE
  REFNAME="name"
  [DESIGNNOTE="text"]
  [UUID="id"]
  [DESCRIPTION="text"]
  [ENABLED="true|false"]
  [HELPTXT="text"]
  SCRIPTTYPE="SERVERRULE|BROWSERRULE|SERVERCALCULATION|SERVERCONVERSION|
  SERVERRANDOMIZATION|CLINTRIALDERIVATION|CLINTRIALRULE"
  [SCRIPTTEXT="text"]
  [SCRIPTFILE="file"]
  [MSGISDERIVED="true|false"]
  [EMPTYISTRUE="true|false"]
  [MSGTEXT="text"]
  [RULEACTION="REPORT|REJECT"]>
<RULEARG* attributes/>
```

```
<EVENTREF attributes/>
<TESTCASE attributes/>
</RULE>
```

属性

REFNAME="name"

AttachRuleSet エlement内でアイテムにルールを関連付けるときに使用される名前。
スペースを含み 63 文字以内。必須

DESIGNNOTE="text"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

UUID="id"

Universally Unique Identifier すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列です。オプション

DESCRIPTION="text"

ルールの説明オプション

ENABLED="true|false"

ルールがアイテムに付加された場合、適切な状況下で有効になり、実行されるかどうかを示します。オプションは true または false です。デフォルトは、True になっています。オプション

HELPTEXT="text"

ルールによって実行されるデータチェックの説明。ルールが付加されているデータアイテムの CRF ヘルプに、この説明が表示されます (CRF ヘルプが適切にマッピング定義されている場合)。ルールが付加されたアイテムまたはアイテムセットによって変わる、コンテキスト固有のヘルプを作成できます。InForm ソフトウェア は、各アイテムまたはアイテムセットのルール引数として定義された値をヘルプテキストに代入します。これにより複数のルールの適用に同じヘルプテキストを使用することができます。オプション

SCRIPTTYPE="SERVERRULE|BROWSERRULE|SERVERCALCULATION|SERVERCONVERSION|SERVERRANDOMIZATION|CLINTRIALDERIVATION|CLINTRIALRULE"

ルールのスクリプトの目的。オプションは以下の通りです。

- **Serverrule** — サーバ上で実行されるルール。
- **Browserrule** — ブラウザ上で実行されるルール。数字フィールドでのテキスト入力、アルファベットフィールドまたはスマートコントロールでの数値入力のチェック。2 つのブラウザサイドルールが InForm アプリケーションでサポートされています。スマートコントロールを使用すると、関連付けられているテキストコントロールまたはリストでユーザが値を入力または選択したときに、自動的にラジオボタンおよびチェックボックスが選択されて提供されます。
- **Servercalculation** — このルールは、他の 1 つ以上のアイテム値に基づき計算された

データアイテムの結果を提供します。

- **Serverconversion** — このルールは単位をある標準から別の標準へ変換します。
- **Serverrandomization** — このルールは計算コントロールに割り当てる無作為割付された薬剤キット番号を決定します。
- **Clintrialderivation** — ルールは Clintrial スタディで使用され、Clintrial データベースの他の 1 つ以上のアイテムの値に基づいている値が持っている派生データアイテムの結果を提供します。
- **Clintrialrule** — このルールは Clintrial スタディで使用されます。

必須

SCRIPTTEXT="text"

スクリプトのテキスト。Visual Basic スクリプトのみがサポートされます。**SCRIPTTEXT** 属性または **SCRIPTFILE** 属性のいずれかが必要です。

SCRIPTFILE="file">

スクリプトのテキストを含んでいるファイルの名前。Visual Basic スクリプトのみがサポートされます。**SCRIPTTEXT** 属性または **SCRIPTFILE** 属性のいずれかが必要です。

MSGISDERIVED="true|false"

Clintrial ルールで、実行時に Clintrial 計算文によってルールテキストが作成されるかどうかを示します。オプションは true または false です。ルールが Clintrial ルールである場合は必須です。true の場合は、MSGTEXT 値を特定しないでください。

EMPTYISTRUE="true|false"

Clintrial ルールでは、null 戻り値が true または false に変更されるかどうかを示します。ルールが Clintrial ルールである場合、必須です。

MSGTEXT="text"

Clintrial ルールが失敗されるときに表示するテキスト。240 文字以下です。ルールが Clintrial ルールの場合は特定しないでください。オプション

RULEACTION="REPORT|REJECT"

Clintrial ルールで、失敗したルールの検証ステータスが上書き可能かどうかを示します。ルールが Clintrial ルールである場合、必須です。

- **Report** — ルールが失敗しても、検証ステータスをバリデーション済み (Passed Validation) (0) に設定できます。
- **Reject** — レコードはバリデーション失敗 (Failed Validation) ステータス (-1) を持つ必要があります。

子

ルール定義は以下のコンポーネントを含むことができます。

- 0 以上の Rulearg 定義が許可されます。Rulearg 定義はルールスクリプトに渡される

パラメータを含んでいます。

- ある Event 定義でルールが失敗する場合に生じるアクションを定義するために要求されます。ルールにイベントを関連付けるには、Eventref エlementを使用します。

例

以下のルール定義は、Eventref Elementによる evtGeneric イベント定義を含んでいます。さらに、Rulearg Elementが min、max、qtext、および addpath の引数を定義するために使用されます。

```
<RULE REFNAME="rulRangeCheck"
  DESCRIPTION="Ensure data are within upper and lower range"
  ENABLED="true"
  SCRIPTTYPE="SERVERRULE">
<![CDATA[Name : rulRangeCheck
'Desc :Compares minimum and maximum ranges entered to ensure that
'data are within the expected range. This is a generic rule that can
'be attached to any item that has an upper and lower range.
'Args:
'min - minimum value
'max - maximum value
'qtext - querytext
'addpath - additional path to the control (beyond the item)
'
```

Option Explicit

Dim Min, Max, qtext, addpath, item, v_val

```
Min = Patient.GetArgument("min")
Max = Patient.GetArgument("max")
qtext = Patient.GetArgument("qtext")
addpath = Patient.GetArgument("addpath")
item = Patient.GetCurPath() & addpath
```

```
Patient.AddNamedValue "QUERYTEXT", qtext
Result.Rulepassed = 1'true
v_val = Patient.GetValue(item, "", 0,0,0)
  If cDbl(v_val) < cDbl(Min) or cDbl(v_val) > cDbl(Max) then
    Result.Rulepassed = 0
  End If
```

```
]]>
<EVENTREF REFNAME="evtGeneric"/>
<RULEARG
  NAME="addpath"
  TYPE="STRING"/>
<RULEARG
  NAME="qtext"
  TYPE="STRING"/>
<RULEARG
  NAME="min"
  TYPE="STRING"/>
<RULEARG
```

```

        NAME="max"
        TYPE="STRING"/>
    <TESTCASE
        RESULT="Pass"
        INPUT_1="15"
        INPUT_2="22"
        INPUT_3="93"
        INPUT_4="2"/>
</RULE>

```

Rulearg

目的

ルールスクリプトの `GetArgument` メソッドによって参照されるパラメータに引数を渡します。Rulearg 定義セットを作成して複数の AttachRuleSet 定義内のさまざまなアイテムやアイテムセットに関連付けることによって、同じルールスクリプトを何度も使用して、複数のアイテムまたはアイテムセットでさまざまな値範囲をテストすることができます。さらに、Rulearg エlementを使用して、ルールによって実行されるデータチェックの CRF ヘルプ説明をカスタマイズすることができます。

構文

```

<RULEARG
    NAME="name"
    VALUE="value"
    TYPE="STRING|NUMERIC|FLOAT|DATE"/>

```

属性

NAME="*name*"

値を指定するルールの名前。必須

VALUE="*value*"

ルールに挿入する値。必須

TYPE="STRING|NUMERIC|FLOAT|DATE"

ルールのデータタイプ。オプションは、STRING (文字列)、NUMERIC (数値)、FLOAT (浮動小数点数)、もしくは (日付) です。デフォルトは STRING (文字列) です。必須

例

以下の例では、Height Range Checking (身長範囲確認) ルールをビジット 1 の症例背景フォームの Height (身長) アイテムに関連付けています。ルールは、このアイテム内の値を参照し、AttachRuleSet の RuleArg のパラメータと、このパラメータに対して指定されている値とで比較を行います。InForm ソフトウェア が AttachRuleSet で RuleArg 値を見つけない場合、ルールのデフォルト値を探します。以下の例では、身長の最小値は、AttachRuleSet で指定されている 34 です。最大値は AttachRuleSet で定義されていないため、InForm ソフトウェア はルールで定義されている最大値 90 を使用し

ます。

Rulearg 値がない場合、Systolic Range Checking (最高血圧の範囲確認) ルールのデフォルト値が検出されます。

```
<RULE REFNAME="Height Range Checking Rule"
  DESCRIPTION="Height Range Checking Rule"
  ENABLED="true"
  SCRIPTTYPE="SERVERRULE"
  SCRIPTFILE="HeightRangeCheck.vbs">
<RULEARG NAME="Min" TYPE="NUMERIC"/>
<RULEARG NAME="Max" VALUE="90" TYPE="NUMERIC"/>
<EVENTREF REFNAME="Height Range"/>
</RULE>

<ATTACHRULESET REFNAME="Height Range Checking Rule" RULENAME="Height Range
Checking Rule"
  FORMSETNAME="Visit1"
  FORMNAME="DEM"
  SECTIONNAME="DEM"
  ITEMNAME="HEIGHT"
  HELPTEXT="Height is expected to be between 34- 76 IN or 88-191 CM."
  ATTACHTYPE="true"
  ACTIVE="true">
<RULEARG NAME="Min" VALUE="34" TYPE="NUMERIC"/>
</ATTACHRULESET>
```

Section

目的

フォーム上の関連するアイテムまたはアイテムセットのグループをセクション見出しの下で定義します。

構文

```
<SECTION
  REFNAME="name"
  [DESIGNNOTE="text"]
  [TITLE="text"]
  [UUID="id"]
  [NOTE="text"]
  [LANGUAGE="name"]
  [REPEATING="true|false"]>
  <ITEMREF+ attributes/>
  <TRANSLATIONS/>
</SECTION>
```

属性

REFNAME="*name*"

フォーム定義でセクションを参照するときに使用される名前。この名前はセクション間で一意であり、かつ 31 文字以下である必要があります。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

TITLE="*text*"

画面上のセクションの一番上に表示されるセクションタイトルのテキスト。

TRANSLATIONS 属性を使用して、属性値を翻訳することができます。オプション

UUID="*id*"

Universally Unique Identifier すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列です。オプション

注： 特記事項：繰り返しフォームセット（「**FormSet**」（139ページ）を作成するとき（主に予定外のビジットを識別する場合）、フォームセットの最初のフォームは以下の **UUID** を含むセクション定義を含んでいる必要があります。

BD991BBE-B0A4-11D2-80E3- 00A0C9AF7674

NOTE="*text*"

セクションタイトルの直下に表示するノートのテキスト。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合（**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合）は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

REPEATING="true|false"

セクションがアイテムセット定義を含むかどうかを示します。オプションは true または false です。デフォルトは、False になっています。オプション

アイテムセットに関する以下の規則に留意してください。

- REPEATING 属性が true に設定されているセクション定義にのみアイテムセットコンポーネントを加えることができます。
- 1 つの REPEATING セクション定義に 1 つのアイテムセット定義しか追加できません。
- 通常の繰り返しでないアイテム定義は、REPEATING セクション定義内に挿入することができません。

子

- 1 つ以上の Itemref 定義。各 Itemref は、事前定義済みのアイテムまたはアイテムセットを参照します。

注： 繰り返しフォームセット (139ページ) が予定外のビジットを特定するために作成された場合、フォームセットの最初のフォームは **Date of Visit** セクション定義を含む必要があります。この **Date of Visit** セクションは 1 つのアイテムから成ります。

DateTimeControl 定義はビジットの月、日、年、時、および分を含んでいます。

DateTimeControl 定義はビジットの月、日、年、時、および分を含んでいます。**Section、Item、および DateTimeControl** コントロール定義は例セクションで特に示したように特定の **UUID** を含む必要があります。

注： InForm アプリケーションと MedML インストーラユーティリティは、**UUID** の英字を大文字に変換します。

- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エlementには 1 つ以上の Translation 定義を含める必要があります。Translation エlementはそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エlementは、スタディコンポーネント定義内の最後のチャイルドElementにする必要があります。Section 定義では、**NOTE** 属性と **TITLE** 属性を翻訳できます。属性を翻訳する場合は、スタディコンポーネント定義の一部であるすべての翻訳可能な属性の翻訳定義を含めます。

例 1

この例では、Duration of Hypertension (高血圧症の期間) というセクションを生成する方法が示されています。セクションは 2 つの Itemref から構成されています。

- 最初の Itemref は、質問とラジオコントロールから成るアイテムを参照します。この例では、「はい/いいえ」ラジオコントロールが事前に定義されているものとします。
- 2 つ目の Itemref は、質問とグループコントロールから成るアイテムを参照します。グループコントロールは 2 つのテキストコントロールで構成されています。それぞれについて、Unitref Elementを使用して Unit 定義が含まれています。

1 年と月の Unit の定義を作成します。

```
<UNIT REFNAME="YEARS" SYMBOL="Year(s)" CLASSIFICATION="Time"
```

```

    BASEREFNAME="MONTHS" CONVERSIONTOBASE="12" CONVERSIONFROMBASE=".0833
    UUID="498100f6-e9df-11d1-9e60- 00a0c9769a33"/>
<UNIT REFNAME="MONTHS" SYMBOL="Month(s)" CLASSIFICATION="Time"
    BASEREFNAME="MONTHS" CONVERSIONTOBASE="1" CONVERSIONFROMBASE="1"
    UUID="498100f7-e9df-11d1-9e60- 00a0c9769a33"/>

```

- 2 Unitref エレメントを使用して、テキストコントロール定義に年と月の単位を挿入します。

```

<TEXTCONTROL REFNAME="HTYEARSTXT"
    NAME="HTYEARSTXT"
    HEIGHT="1"
    LENGTH="3"
    MAXLENGTH="3"
    DATATYPE="INTEGER">
    <UNITREF* attributes/>
</TEXTCONTROL> <TEXTCONTROL REFNAME="HTMONTHSTXT"
    NAME="HTMONTHSTXT"
    HEIGHT="1"
    LENGTH="3"
    MAXLENGTH="3"
    DATATYPE="INTEGER">
    <UNITREF* attributes/>
</TEXTCONTROL>

```

- 3 テキストコントロールを使用して GroupControl 定義を作成します。

```

<GROUPCONTROL REFNAME="HTDURATIONGRP" NAME="HTDURATIONGROUP"
    LAYOUT="HORIZONTAL">
    <CONTROLREF REFNAME="HTYEARSTXT" ORDER="1"/>
    <CONTROLREF REFNAME="HTMONTHSTXT" ORDER="2"/>
</GROUPCONTROL>

```

- 4 セクション内の 2 つのラインアイテム用の Item 定義を作成します。

```

<ITEM REFNAME="HTYESNO" QUESTION="Was hypertension previously diagnosed? ">
    <CONTROLREF REFNAME="YESNORADIO"/>
</ITEM>
<ITEM REFNAME="HTDURATION" QUESTION="If hypertension was previously diagnosed,
    enter duration of hypertension: ">
    <CONTROLREF REFNAME="HTDURATIONGRP"/>
</ITEM>

```

- 5 高血圧症の期間のセクション定義にアイテム定義を同封するために Itemrefs を使用します。

```

<SECTION REFNAME="HYPERTENSION"
    TITLE="Duration of Hypertension">
    <ITEMREF REFNAME="HTYESNO" ORDER="1"/>
    <ITEMREF REFNAME="HTDURATION" ORDER="2"/>
</SECTION>

```

例 2

この例はあらゆるビジットの最初のフォームと繰り返しフォームセット (139ページ) の最初のフォームに必要とされる Date of Visit セクション定義を示しています。このセクションには、ユーザがビジットの日時を入力できるアイテム (DOV アイテム) を 1 つだけ含める必要があります。このセクションは、別のフォームとして作成することを推奨します。さらに、この例では **TITLE** 属性と **NOTE** 属性を翻訳しています。

```

<DATETIMECONTROL REFNAME="DOV" NAME="DOV"
    UUID="BD991BC0-B0A4-11D2-80E3- 00A0C9AF7674"
    STARTYEAR="1997" ENDYEAR="2004"
    DISPLAYDAY="true" DISPLAYMONTH="true"
    DISPLAYYEAR="true" DISPLAYHOUR="true"
    DISPLAYMINUTE="true"
    REQUIREMONTH="true" REQUIREYEAR="true"/>
<ITEM REFNAME="DOV" QUESTION="Date and time of visit:"
    UUID="BD991BBF-B0A4-11D2-80E3-00A0C9AF7674">

```

```

    <CONTROLREF REFNAME="DOV"/>
</ITEM>

<SECTION REFNAME="DOV" TITLE="Date Of Visit"
  NOTE="Complete this form before any others in the visit."
  LANGUAGE="en-US"
  UUID="BD991BBE-B0A4-11D2-80E3-00A0C9AF7674">
  <ITEMREF REFNAME="DOV" ORDER="1"/>
  <TRANSLATIONS>
    <TRANSLATION NAME="TITLE" LOCALE="fr-FR" DISPLAYTEXT="Date de la visite"/>
    <TRANSLATION NAME="NOTE" LOCALE="fr-FR"
      DISPLAYTEXT="Remplissez ce formulaire avant tout autre à la visite."/>
  </TRANSLATIONS>
</SECTION>
<FORM REFNAME="DOV" TITLE="Date of Visit" MNEMONIC="DOV" TYPE="CRF">
  <SECTIONREF REFNAME="DOV"/>
</FORM>

```

Sectionref

目的

- 定義済みセクションをフォームの定義に含めます。
- フォームのセクションごとに 1 つの Sectionref エlementを含めます。
- マッピング定義の Path Elementでのマップされたコントロールの位置としてセクションの RefName を指定します。
- Path Elementによって定義された RefName パスに 1 つの Sectionref Elementを含めます。

Sectionref は Form または Path Elementのチャイルドとしてのみ定義されます。単独のコンポーネントとしてサブミットされません。

構文

```

<SECTIONREF
  REFNAME="name"
  [ORDER="n"]/>

```

属性

REFNAME="*name*"

フォームまたはパスの定義に含まれるセクションの RefName。必須

ORDER="*n*"

フォーム定義における各 Sectionref の順序。順序を指定しない場合、MedML インストラユーティリティは、Sectionref が参照するアイテムを入力された順番で順序付けします。オプション

注： Path Elementが Sectionref 定義を含む場合、**REFNAME** 属性のみが有効です。

例

この例は症例背景フォームの定義を示しています。2 つのセクション、症例背景（DEM; Demographics）および喫煙履歴（SH; Smoking History）から構成されます。

```
<FORM REFNAME="DEM" TITLE="Demographics" MNEMONIC="DEM"
FORMTYPE="CRF">
  <SECTIONREF REFNAME="DEM"/>
  <SECTIONREF REFNAME="SH"/>
</FORM>
```

以下の例は、マッピングされた COMMONDATE コントロールが出現する位置として LEADEC G セクションを示します。

```
<PATH>
  <CHAPTERREF REFNAME="PF_ALL_VISITS"/>
  <PAGEREF REFNAME="ECG"/>
  <SECTIONREF REFNAME="LEADEC G"/>
  <ITEMSETREF REFNAME="1"/>
  <ITEMREF REFNAME="DATEASSESS"/>
  <CONTROLREF REFNAME="COMMONDATE"/>
</PATH>
```

Sequence

目的

番号付けするシーケンスを指定します。Sequence により、スポンサーは、スクリーニングされた症例、登録済み症例、クエリおよび他のタイプの番号付きデータに番号を割り振る方法を示すことができます。各シーケンスは SequenceType エレメントで定義されたシーケンスタイプと関連付けられています。

構文

```
<SEQUENCE
  UUID="id"
  SEQUENCENAME="name"
  SEQUENCETYPENAME="name" />
```

属性

UUID="*id*"

Universally Unique Identifier すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列です。必須

注：InForm アプリケーションと MedML インストーラユーティリティは、UUID の英字を大文字に変換します。

SEQUENCENAME="*name*"

シーケンス定義を作成している番号タイプの記述的名前。必須

SEQUENCETYPENAME="*name*"

シーケンス番号が属するシーケンスタイプの名前。必須

例

以下の例は、XML エlement を利用した Query (クエリ)、Enrollment (症例登録)、Screening (スクリーニング) および Randomization (無作為割付) 番号割り付けシーケンスを示しています。

```
<SEQUENCE SEQUENCENAME="Query Number Sequence"
  SEQUENCETYPENAME="Query"
  UUID="5b7d4eb4-0465-11d2-a414-00a0c963e0ac" />
```

```
<SEQUENCE SEQUENCENAME="Enrollment Number Sequence"
  SEQUENCETYPENAME="Enrollment"
  UUID="eb75b898-078b-11d2-a417-00a0c963e0ac" />
```

```
<SEQUENCE SEQUENCENAME="Screening Number Sequence"
  SEQUENCETYPENAME="Screening"
  UUID="f7f1b3b8-0b5c-11d2-a418-00a0c963e0ac" />
```

```
<SEQUENCE SEQUENCENAME="Simple SimpleCentral"
  SEQUENCETYPENAME="Randomization"
  UUID="4F4A0246-5009-11d2-931C- 00A0C9769A13" />
```

SequenceType

目的

InForm アプリケーションが自動的に連番を作り出すエンティティタイプを定義します。Sequence エlementを使用して番号の順番を定義する場合、SequenceType 定義を含めることにより、シーケンスが属するタイプを指定します。InForm ソフトウェア は以下のタイプのシーケンス番号を定義しています。

- スクリーニング
- 症例登録
- クエリ
- 無作為割付

構文

```
<SEQUENCETYPE
  [UUID="id"]
  SEQUENCETYPENAME="name" />
```

属性

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

SEQUENCETYPENAME="*name*"

シーケンスタイプの名前。必須

例

以下の SequenceType 定義が、InForm ソフトウェア で使用される基本シーケンスタイプを作成します。

```
<SEQUENCETYPE SEQUENCETYPENAME="Enrollment"/>
<SEQUENCETYPE SEQUENCETYPENAME="Screening"/>
<SEQUENCETYPE SEQUENCETYPENAME="Query"/>
<SEQUENCETYPE SEQUENCETYPENAME="Randomization"/>
```

SignatureGroup

目的

ケースブックまたは CRF に署名することができるユーザグループを指定します。

注: ユーザインターフェースでは署名グループを作成できません。署名グループを定義するには **MedML** を使用する必要があります。

グループのユーザは、**InForm** アプリケーションでのみ削除できます。**MedML** インストーラユーティリティの実行によって、ユーザを削除することはできません。

構文

```
<SIGNATUREGROUP
  GROUPNAME="name"
  [GROUPDESCRIPTION="text"]
  [UUID="id"]
  [LANGUAGE="name"]
  [CRFTEXT="text"]
  [CRFFILE="file"]
  [CRFMEANING="text"]
  [CRBTEXT="text"]
  [CRBFILE="file"]
  [CRBMEANING="text"]>
  <USERREF* attributes/>
  <TRANSLATIONS/>
</SIGNATUREGROUP>
```

属性

GROUPNAME="name"

署名グループの名前必須

GROUPDESCRIPTION="text"

署名グループの説明オプション

UUID="id"

Universally Unique Identifier すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列です。オプション

LANGUAGE="name"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

CRFTEXT="text"

署名グループに関連付けられた CRF に対して、そのメンバが署名する際に表示される電子署名宣誓書のテキスト。**CRFTEXT** 属性または **CRFFILE** 属性のいずれかを指定します。どちらの属性も指定しない場合、MedML インストーラユーティリティはデフォルトの CRF 署名テキストリソースで提供されるテキストを使用します。テキストは次のものを含む必要があります。

- 署名の意味。たとえば、CRF で提供されるデータの正確さを証明するため、などになります。
- 電子署名が法的に手書き署名と同等の拘束力があるという署名者の意思。

任意で、テキストにユーザの名および姓を表すための代替文字 (%s) を含むことが可能で

す。例セクションを参照してください。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

CRFFILE="file"

署名グループに関連付けられた CRF に対して、そのメンバが署名する際に表示される電子署名宣誓書のテキストを含む HTML、またはテキストファイルのパス名。パス名は、MedML インストーラユーティリティを実行するディレクトリに対して相対パスである必要があります。**CRFTEXT** 属性または **CRFFILE** 属性のいずれかを指定します。どちらの属性も指定しない場合、MedML インストーラユーティリティはデフォルトの CRF 署名テキストリソースで提供されるテキストを使用します。オプション

CRFMEANING="text"

CRF に対する署名の意味を要約したテキスト。このテキストは、[署名詳細] ページに表示されます。また、このテキストは、CRF 上の署名済および必要な署名のリストにも表示されます。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。

注: 署名の意味を示すテキストは、**CRF** が署名された施設の言語で保存されます。この言語は、異なる施設スタディロケールを持つ別の施設に症例が移動された場合も変更されません。

CRBTEXT="text"

電子署名宣誓書のテキストは、署名グループに関連するケースブックに署名する際に、署名グループのメンバに表示されます。**CRBTEXT** 属性または **CRBFILE** 属性のいずれかを指定します。どちらの属性も指定しない場合、MedML インストーラユーティリティはデフォルトのケースブック署名テキストリソースで提供されるテキストを使用します。テキストは次のものを含む必要があります。

- 署名の意味。たとえば、フォームで提供されるデータの正確さを証明するため、などになります。
- 電子署名が法的に手書き署名と同等の拘束力があるという署名者の意思。

任意で、テキストにユーザの名および姓を表すための代替文字 (%s) を含むことが可能です。例セクションを参照してください。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

CRBFILE="file"

署名グループに関連付けられているケースブックに署名する際、その署名グループのメンバに対して表示される電子署名宣誓書のテキストを含む HTML またはテキストファイルのパス名。パス名は、MedML インストーラユーティリティを実行するディレクトリに対して相対パスである必要があります。**CRBTEXT** 属性または **CRBFILE** 属性のいずれかを指定します。どちらの属性も指定しない場合、MedML インストーラユーティリティはデフォルトのケースブック署名テキストリソースで提供されるテキストを使用します。オプション

CRBMEANING="text"

ケースブックの署名の意味を要約したテキスト。このテキストは、[署名詳細] ページに表示されます。また、このテキストは、ケースブックへの署名用に使用される CRF 上の署名済および必要な署名のリストにも表示されます。**TRANSLATIONS** 属性を使用し

て、属性値を翻訳することができます。

UPDATE="true|false"

指定したプロパティのみを変更して、指定した署名グループの増分更新を実行します。

子

- 0 以上の Userref 定義。各 Userref は、1 人の InForm ユーザを特定する作成済の User 定義を参照します。
- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エlement には 1 つ以上の Translation 定義を含める必要があります。Translation エlement はそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エlement は、スタディコンポーネント定義内の最後のチャイルドElement にする必要があります。SignatureGroup 定義では、**CRFTEXT** 属性、**CRBTEXT** 属性、**CRFMEANING** 属性、および **CRBMEANING** 属性を翻訳できます。属性を翻訳する場合は、スタディコンポーネント定義の一部であるすべての翻訳可能な属性の翻訳定義を含めます。

注： **CRFFILE** 属性と **CRBFILE** 属性は翻訳できません。

例

次の署名グループの、ユーザ (dobrien と lhill) は署名することができます。

```
<SIGNATUREGROUP GROUPNAME="CRA Signature">
  <USERREF USERNAME="dobrien"/>
  <USERREF USERNAME="lhill"/>
</SIGNATUREGROUP>
```

以下の SignatureGroup 定義は、CRF へ署名テキストを挿入する方法を示しています。2 つの %s 文字は、表示されている電子署名宣誓書にユーザの氏名を含める必要があることを示しています。>Font< タグ属性はシングルクォーテーションで囲う必要があることに注意してください。これは、タグ属性がダブルクォーテーションで囲まれた **CRFTEXT** 属性内で扱われるためです。さらに、SignatureGroup 定義には、**CRFTEXT** 属性の翻訳文字列の定義も含まれています。

```
<SIGNATUREGROUP GROUPNAME="PI Signature" LANGUAGE="en-US"
  CRFTEXT="&lt;font name='Arial' size='2'&gt;By my dated signature below,
  I, %s %s, verify that this case report form accurately displays
  the results of the examinations, tests, evaluations and
  treatments noted within.&lt;br&gt;&lt;br&gt;
  Pursuant to Section 11.100 of Title 21 of the Code of Federal
  Regulations, this is to certify that I intend that this
  electronic signature is to be the legally binding equivalent
  of my handwritten signature.&lt;br&gt;&lt;br&gt;
  To this I do attest by supplying my user name and password and clicking
  the button marked &t b &gt;Submit&lt;/b&gt; below.&lt;/font&gt;"
  CRFMEANING="Approval">
  <USERREF USERNAME="John"/>
  <USERREF USERNAME="George"/>
  <TRANSLATIONS>
    <TRANSLATION NAME="CRFTEXT" LOCALE="fr-FR"
      DISPLAYTEXT="&lt;font name='Arial' size='2'&gt;Par ma signature dat_e
      ci-dessous,
      Moi, %s %s, je confirme que ce cas formulaire de rapport affiche avec pr_cision
      les r_sultats des examens, des tests, des_valuations et des
      constat_es dans les traitements. &lt;br&gt; &lt;br&gt;"
```

```

Conform ment _ la Section de 11,100 Titre 21 du Code of Federal
R_glement, il_s'agit de certifier que j'ai l'intention que ce
signature _lectronique doit _tre juridiquement contraignant _quivalent
de ma signature manuscrite. &lt;br&gt; &lt;br&gt;
Pour cela, je ne t moignent par la fourniture de mon nom d'utilisateur et
mot de passe et cliquez sur
le bouton &lt; b &gt;Submit&lt;/b&gt; ci-dessous.&lt;/font&gt;
<TRANSLATION NAME="CRFMEANING" LOCALE="fr-FR"
DISPLAYTEXT="Approbation"/>
</TRANSLATIONS>
</SIGNATUREGROUP>

```

SignCRF

目的

署名を要求するフォームを識別し、ユーザが CRF に署名するために所属すべき署名グループを指定します。

構文

```

<SIGNCRF
  SIGNATUREGROUPNAME="name"
  FORMREFNAME="name"
  [RESETFORMSTATE="true|false"]
  [INVALIDATIONLEVEL="USER|GROUP"]
  [FINALCRF="true|false"]/>

```

属性

SIGNATUREGROUPNAME="*name*"

FORMREFNAME 属性で指定されているフォームへの署名が承認されるユーザを含む署名グループの RefName。必須

FORMREFNAME="*name*"

SIGNATUREGROUPNAME 属性で指定される SignatureGroup のメンバであるユーザによって署名される必要がある CRF の RefName。必須

注： スクリーニングと登録 CRF に署名する必要はありません。SignCRF 定義にこれらの RefName を含めないでください。

RESETFORMSTATE="*true|false*"

新しい署名グループを、署名済み CRF に関連付けることにより、CRF の状態を、署名されていない状態にリセットするかどうかを示します。CRF がリセットされても、元の署名は有効のままです。ただし、CRF は、新しく関連付けられた署名グループのメンバからも署名を受けることが必要になります。デフォルトは、False になっています。
オプション

INVALIDATIONLEVEL="USER|GROUP"

データアイテムが CRF の署名後にインポートされるときに署名を無効にするかどうかを指定します (コード化された値が Central Coding アプリケーションでインポートされ

る場合など)。オプションは以下の通りです。

- **USER** — 署名したユーザが、インポートされたアイテムを表示できる場合は、CRF またはケースブックの署名は無効になります。
- **GROUP** — 署名グループ内で少なくとも 1 人のユーザがインポートされたアイテムを表示できる場合は、CRF またはケースブックの署名は無効になります。

INVALIDATIONLEVEL 属性を指定しない場合、InForm アプリケーションではフォームが直接またはインポートによって編集されるたびに署名が無効になります。オプション

FINALCRF="true|false"

FORMREFNAME 属性で指定されたフォームに署名することで、ケースブック全体が署名されるかどうかを示します。デフォルトは、False になっています。オプション

例

以下の例では、SignCRF エlementを使用して、DEM、VS および SC の各フォームが「CRA 署名」署名グループのメンバによって署名される必要があることを指定する方法を示しています。FINALCRF 属性は、SC フォームに署名することでケースブックに署名することを示します。

```
<SIGNCRF SIGNATUREGROUPNAME="CRA Signature" FORMREFNAME="DEM"/>
<SIGNCRF SIGNATUREGROUPNAME="CRA Signature" FORMREFNAME="VS"
  INVALIDATIONLEVEL="USER"/>
<SIGNCRF SIGNATUREGROUPNAME="CRA Signature" FORMREFNAME="SC"
  FINALCRF="true"/>
```

SimpleControl

目的

ラジオボタンあるいはチェックボックスのリストなどのコンパウンドコントロールのオプションを定義します。シンプルコントロールを使用することにより、個々のオプションを異なるタイプのコントロールとして作成するコンパウンドコントロールを定義できます。例えば、ラジオボタンリストについて、あるオプションではユーザが明示的に選択するオプションであり、あるオプションではユーザがテキストを入力するオプション、といった具合です。以下のタイプのコンパウンドコントロールは、1 つ以上のシンプルコントロールを含むことができます。

- CheckBoxControl
- GroupControl
- Item
- RadioControl

構文

```
<SIMPLECONTROL
  REFNAME="name"
  [DESIGNNOTE="text"]
```

```

[NAME="name"]
[UUID="id"]
[CAPTION="text"]
[LANGUAGE="name"]
[CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"]
[ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM">]
[UNITDISPLAYTYPE="ELEMENT">
  <ELEMENTREF+ attributes/>
  <UNITREF attributes/>
  <TRANSLATIONS/>
</SIMPLECONTROL>

```

属性

REFNAME="*name*"

他のコントロールの定義内でシンプルコントロールを参照するときに使用される名前。この名前はシンプルコントロール間で一意である必要があります。必須

DESIGNNOTE="*text*"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

NAME="*name*"

他のコントロールの定義内でシンプルコントロールを参照するときに使用される名前。オプション

UUID="*id*"

Universally Unique Identifier; すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

CAPTION="*text*"

シンプルコントロールと共に画面上に表示されるテキスト。このキャプションは、PFElement に対して定義されたキャプションに添える追加テキストを定義する機会を提供します。**TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。オプション

LANGUAGE="*name*"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注： デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

CAPTIONALIGN="LEFT|RIGHT|TOP|BOTTOM"

キャプションの計算値との相対位置。LEFT (左)、RIGHT (右)、TOP (上)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

ALIGN="LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM"

コントロール内の PFElement の位置合わせ。LEFT (左)、CENTER (中央)、RIGHT

(右)、TOP (上)、MIDDLE (中央)、もしくは BOTTOM (下)。デフォルトは LEFT (左) です。オプション

UNITDISPLAYTYPE="ELEMENT"

Unitref 定義を持つシンプルコントロールに含まれる、単位を表示するために使用されるコントロールのタイプ。「ELEMENT」のみが有効な値です。

子

- 事前定義済みの PFElement を参照する 1 つの Elementref 定義。
- 事前定義済みの単位を参照する 0 または 1 個以上の Unitref 定義。
- スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エlementには 1 つ以上の Translation 定義を含める必要があります。Translation エlementはそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Translations エlementは、スタディコンポーネント定義内の最後のチャイルドElementにする必要があります。SimpleControl 定義の **CAPTION** 属性は翻訳できます。

例

この例は、性別を指定する PFElement のセットを定義し、それをシンプルコントロールに含める方法を示しています。シンプルコントロールは、ラジオコントロールなどのコンパウンドコントロールに含めることができます。

1 "男性"、"女性"の選択肢を定義する PFElements を作成します。

```
<PFELEMENT REFNAME="MALE" LABEL="男性" TYPE="STRING" VALUE="MElement"/>
<PFELEMENT REFNAME="FEMALE" LABEL="女性" TYPE="STRING" VALUE="FElement"/>
```

2 Elementref を使用して各 PFElement の定義を含む SimpleControl を作成します。

```
<SIMPLECONTROL REFNAME="MALE"
  <ELEMENTREF REFNAME="MALE"/>
</SIMPLECONTROL>
<SIMPLECONTROL REFNAME="FEMALE">
  <ELEMENTREF REFNAME="FEMALE"/>
</SIMPLECONTROL>
```

ラジオボタンリストでの SimpleControl の使用例は、「**RadioControl**」(174ページ)を参照してください。

Site

目的

スタディが実施される施設を定義します。

注: ユーザーインターフェースでは施設を定義できません。施設グループを定義するには **MedML** を使用する必要があります。

構文

```
<SITE
  [NAME="name"]
```

```
[MNEMONIC="name"]
[ADDRESS="addr1"]
[ADDRESS2="addr2"]
[CITY="name"]
[STATE="name"]
[PROVINCE="name"]
[ZIPCODE="code"]
[POSTCODE="code"]
[COUNTRY="name"]
[PHONE="num"]
[ALTPHONE="num"]
[FAX="num"]
[EMAIL="addr"]
[TIMEZONE="name"]
[STARTDATE="date"]
[ENDDATE="date"]>
[SVAUTOSELECTRATE="num"]
[SVFIRSTNSUBJECTS="num"]
[SITESERVER="server name"]
[SITEDATEFORMAT="MONTH_DAY_YEAR|DAY_MONTH_YEAR|YEAR_MONTH_DAY"]
[STUDYLOCALE="text"]
[USERNAMEORDER="F,L|L,F"]>
</SITE>
```

属性

NAME="*name*"

施設の名前。必須

MNEMONIC="*name*"

施設の略称。必須

ADDRESS="*addr1*"

施設アドレスの第 1 のライン。オプション

ADDRESS2="*addr2*"

施設アドレスの第 2 ライン。オプション

CITY="*name*"

施設の市区町村。オプション

STATE="*name*"

施設の都道府県。オプション

PROVINCE="*name*"

施設の都道府県。オプション

ZIPCODE="*code*"

施設のジップコード。オプション

POSTCODE="*code*"

施設郵便番号。オプション

COUNTRY="*name*"

施設の国。オプション

PHONE="*num*"

施設の電話番号。オプション

ALTPHONE="*num*"

施設の予備電話番号。オプション

FAX="*num*"

施設のファクス番号オプション

EMAIL="*addr*"

施設への問い合わせ用電子メールアドレス。オプション

TIMEZONE="*name*"

施設が位置するタイムゾーン。内部のユニバーサルシステム時間から現地時間に変更す

る場合に使用されます。この属性値は、次のいずれか 1 つである必要があります。

- Windows レジストリキー
HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Time Zones
内のサブキー名の 1 つ。

このオプションにより、SITE MedML がオペレーティングシステムのロケールで処理されます。

- InForm サーバのオペレーティングシステムロケールの表示名。
このオプションにより、SITE MedML が、表示名と一致するオペレーティングシステムのロケールでのみ処理されます。

必須

STARTDATE="date"

施設がオンラインになった日付。ユーザは、指定された日以前には施設用データを追加することができません。日付情報を指定する際には、次のことに注意してください。

- 年値は、100 と 9999 の間である必要があります。また、省略データ形式の場合でも、年は 4 桁すべて入力してください。
- 多くの日時形式があります。次のテーブルの例をご覧ください。

フォーマット	例
"dd month yyyy"	"25 January 1996"
"hh:mm:ss" (12 時間表記)	"8:30:00"
"hh:mm:ss" (24 時間表記)	"20:30:00"
"month dd, yyyy hh:mm:ss:"	"January 25, 1996 8:30:00"
"hh:mm:ss mon dd, yyyy"	"8:30:00 Jan. 25, 1996"
"mm/dd/yyyy hh:mm:ss"	"1/25/1996 8:30:00"

必須

ENDDATE="date"

施設がオフラインになった日付。たとえば、最後の症例が完了し、ロックされた日。
オプション

SITESERVER="server name"

施設サーバとして特定されたサーバの名前。施設サーバは、無作為割付、スクリーニング、症例登録、および症例番号の作成などのアクティビティのために使用されます。

SITEDATEFORMAT="MONTH_DAY_YEAR"

日付形式がユーザレベルで指定されていない場合に、その施設について表示される日付形式。オプション

STUDYLOCALE="text"

施設で優先するスタディロケールのコード。スタディロケールは、スタディメタデータを定義するロケールであり、ビジット名、CRF 名、セクションラベル、質問、およびコントロールラベルを含みます。値は、そのスタディのスタディバージョン用スタディロケールと一致する必要があります。必須

注：症例を施設に追加した後は、ユーザインターフェースからも **MedML** からも施設のスタディロケールを変更できません。

SVAUTOSELECTRATE="n"

SDV を実施する必要がある、SDV 必須とマークされたフォームの割合に対応する数値。

SVFIRSTNSUBJECTS="n"

症例に登録済ステータスが割り当てられた時間に基づいて、順番に SDV を実施する症例の数。たとえば、スタディに登録されている最初の 7 つの症例に対して SDV を実施する必要がある場合は、SVFIRSTNSUBJECTS=7 と指定します。

USERNAMEORDER="F,L|L,F"

電子署名宣誓書に表示されるユーザの姓と名の順番。デフォルトでは、[システム構成] ページでそのスタディに対して指定されているユーザ名の順序になります。施設用に指定された順番はスタディ用に指定された順番より優先されます。

- **F,L** — 名を先に、姓を後ろにします。
- **L,F** — 姓を先に、名を後ろにします。

オプション

UPDATE="true|false"

指定したプロパティのみを変更して、指定した施設の増分更新を実行します。

例

次の例は、Username が Dr Cortina というユーザを含む、スペイン、バレンシアにある

Clinica Ortopedica という名前の施設を定義する方法を示しています。

```
<SITE NAME="Clínica Ortopédica"
  MNEMONIC="CORTO"
  ADDRESS="2150 Avenida Universidad"
  ADDRESS2="Oficina 14B"
  CITY="Valencia"
  STATE="ES"
  PHONE="+34 96-555-55-55"
  FAX="+34 96-555-55-99"
  EMAIL="drcortina@clinicaortopedica.com"
  TIMEZONE="CET"
  STARTDATE="10/23/2008"
  STUDYLOCALE="es-ES"
  USERNAMEORDER="L,F">
</SITE>
```

SiteGroup

目的

指定された施設にアクセスできるユーザのグループを指定します。

注：ユーザをグループから削除するには、InForm アプリケーションの **Admin** 機能を通して行う方法しかないことに注意してください。MedML インストーラユーティリティの実行によって、ユーザを削除することはできません。

構文

```
<SITEGROUP
  SITENAME="name">
  <USERREF* attributes/>
</SITEGROUP>
```

属性

SITENAME="name"

SiteGroup にアクセス権を付与する施設の名前。この名前は、施設の Site 定義内の **NAME** 属性に対応します。必須

子

SiteGroup 定義には 0 または 1 個以上の Userref 定義を含めることができます。各 Userref は、1 人の InForm ユーザを特定する作成済みの User 定義を参照します。

例

以下の例では、ユーザ Marge と Jonah が Beth Israel という名前の施設の SiteGroup に割り当てられています。

```
<SITEGROUP SITENAME="Beth Israel">
  <USERREF USERNAME="Marge"/>
  <USERREF USERNAME="Jonah"/>
</SITEGROUP>
```

Sponsor

目的

スタディスポンサーを定義します。Sponsor 定義はドキュメンテーション用です。

注：スポンサーを定義するには **MedML** を使用する必要があります。ユーザインターフェイスではこのタスクを実行することはできません。

構文

```
<SPONSOR
  [NAME="name"]
  [PROGRAM="text"]
  [THERAPEUTICAREA="text"]
  [NOTE="text"]
  [ADDRESS="addr1"]
  [ADDRESS2="addr2"]
  [CITY="name"]
  [STATE="name"]
  [PROVINCE="name"]
  [ZIPCODE="code"]
  [POSTCODE="code"]
  [COUNTRY="name"]
  [PHONE="num"]
  [ALTPHONE="num"]
  [FAX="num"]
  [EMAIL="addr"]
  [CONTACTUSERREF="username"]
  [LANGUAGE="name"]
  [LOGOFILE="file"]
  [LOGOTYPE="GIF|JPEG|TEXT"]>
</SPONSOR>
```

属性

NAME="*name*"

スポンサーの名前。オプション

注：**CRF Submit** ソフトウェアを使用して **PDF** ファイルを生成する予定である場合は、名前属性が必要です。

PROGRAM="*text*"

スタディの名前。オプション

THERAPEUTICAREA="*text*"

スタディの治療領域。オプション

NOTE="*text*"

スタディの説明。オプション

ADDRESS="*addr1*"

スポンサーアドレスの第 1 ライン。オプション

ADDRESS2="*addr2*"

スポンサーアドレスの第 2 ライン。オプション

CITY="*name*"

スポンサーアドレスの市区町村。オプション

STATE="*name*"

スポンサーアドレスの都道府県。オプション

PROVINCE="*name*"

スポンサーアドレスの都道府県。オプション

ZIPCODE="*code*"

スポンサーのジップコード。オプション

POSTCODE="*code*"

スポンサーの郵便番号。オプション

COUNTRY="*name*"

スポンサーアドレスの国。オプション

PHONE="*num*"

スポンサーの電話番号。オプション

ALTPHONE="*num*"

スポンサーの予備電話番号。オプション

FAX="*num*"

スポンサーのファックス番号。オプション

EMAIL="*addr*"

スポンサーの電子メールアドレス。オプション

CONTACTUSERREF="*username*"

スポンサーの第一連絡先であるユーザの名前。オプション

LANGUAGE="*name*"

スポンサーとの連絡に使用される言語。デフォルトは英語。オプション

LOGOFILE="*file*"

スポンサーのロゴファイルのファイル名。オプション

LOGOTYPE="GIF|JPEG|TEXT"

スポンサーのロゴファイルのファイルタイプ。オプションは、GIF、JPEG、または TEXT です。Logofile 名を指定する場合には必須です。

例

この例は、Acme Pharma と呼ばれる依頼者のための定義を示しています。

```
<SPONSOR NAME="Acme Pharma"
  PROGRAM="AP603 Study"
  NOTE="New Drug for Hypertension"
  ADDRESS="123 Acme Court"
  ADDRESS2="Building 3-A"
  CITY="Western"
  STATE="Massachusetts"
  ZIPCODE="01240"
  PHONE="413-555-6666"
  FAX="413-555-7777"
  EMAIL="jsmith@acmepharmaceutical.com"
  CONTACTUSERREF="jsmith"
  LOGOFILE="acmelogo.gif"
  LOGOTYPE="GIF"/>
```

StudyVersionDoc

目的

Documentation エレメントで定義する場合と同様に、スタディバージョンにドキュメントを割り当てます。

構文

```
<STUDYVERSIONDOC
  VERSIONDESCRIPTION="n"
  DOCREFNAME="name"
  [ORDER="n"]/>
```

属性

VERSIONDESCRIPTION="*n*"

ドキュメントを関連付けるスタディバージョンの番号。必須

DOCREFNAME="*name*"

Documentation エレメントで指定されたドキュメントの RefName。必須

[ORDER="*n*"]

ドキュメントもしくは、ヘルプウィンドウに、ドキュメントを表すタブが表示される順番。デフォルトは、StudyVersionDoc 定義を持つ XML ファイル内でそのドキュメントが参照されている順番です。オプション

例

MedMLData 定義ファイルからの以下の引用は、StudyVersionDoc エレメントを使用して、ドキュメントセットの StudyVersion を 1 に設定する方法を示しています。

```
<MEDMLDATA>

<!-- Documents -->
<STUDYVERSIONDOC VERSIONDESCRIPTION="1" DOCREFNAME="Protocol"
ORDER="1"/>
<STUDYVERSIONDOC VERSIONDESCRIPTION="1" DOCREFNAME="Study"
ORDER="2"/>
<STUDYVERSIONDOC VERSIONDESCRIPTION="1" DOCREFNAME="Visit"
ORDER="3"/>
<STUDYVERSIONDOC VERSIONDESCRIPTION="1" DOCREFNAME="CRB"
ORDER="4"/>
```

StudyVersionSite

目的

施設が現在使用しているスタディバージョンを記録します。このコンポーネントは、

スタディバージョンが施設内スタディ審査委員会（IRB; Institutional Review Board）の承認を必要としない方法で変更される都度、および、承認が必要なスタディバージョン変更に対して IRB 承認が得られた場合に更新する必要があります。

注 : StudyVersionSite コンポーネントを使用して、施設が使用しているスタディバージョンを特定するには、施設定義をロードする **XML** ファイルを実行する必要があります。

構文

```
<STUDYVERSIONSITE
  VERSIONDESCRIPTION="n"
  [SITENAME="name"]
  [SITEMNEMONIC="name"]
  [ACCEPTDATE="date"]/>
```

属性

VERSIONDESCRIPTION="*n*"

現在施設で使用されている スタディバージョンの番号。この番号は、スタディバージョンが定義された **VERSIONDESCRIPTION** 属性と一致している必要があります。必須

SITENAME="*name*"

施設の名前。**SITENAME** 属性あるいは **SITEMNEMONIC** 属性のいずれかが必要です。この名前は、Site が定義された **NAME** 属性と一致する必要があります。

SITEMNEMONIC="*name*"

施設の略名。この名前は、施設が定義された **MNEMONIC** 属性と一致する必要があります。**SITENAME** 属性あるいは **SITEMNEMONIC** 属性のいずれかが必要です。

ACCEPTDATE="*date*"

スタディバージョン変更の IRB 承認の日付。

注 : ACCEPTDATE 属性はドキュメンテーション用として、繰り返しフォームに表示されるスタディバージョンを区別するために使用されます。施設が新しいスタディバージョンに移行する実際の日付は、その施設の **StudyVersionSite** コンポーネントがデータベースで改訂される日付になります。

例

この例は StudyVersionSite コンポーネントセットを示しています（それぞれが 5 つの施設の各施設に対応します）。

```
<STUDYVERSIONSITE VERSIONDESCRIPTION="2" SITEMNEMONIC="PF"
ACCEPTDATE="6/30/1998" />
<STUDYVERSIONSITE VERSIONDESCRIPTION="1" SITEMNEMONIC="BID"
ACCEPTDATE="6/30/1998" />
<STUDYVERSIONSITE VERSIONDESCRIPTION="2" SITEMNEMONIC="BCH"
ACCEPTDATE="6/30/1998" />
<STUDYVERSIONSITE VERSIONDESCRIPTION="2" SITEMNEMONIC="MGH"
ACCEPTDATE="6/30/1998" />
```

```
<STUDYVERSIONSITE VERSIONDESCRIPTION="2" SITEMNEMONIC="BWH"  
ACCEPTDATE="6/30/1998" />
```

StudyVersion

目的

あるスタディのスタディバージョンとそのプロトコルの下にあるフォームセットをグループ化します。改訂されたフォームまたはスタディドキュメントを実装する必要がある場合は、都度新しい StudyVersion 定義を作成し、変更されたフォームバージョンを適用する各施設の StudyVersionSite 定義をアップデートします。そして、使用する各スタディドキュメントの StudyVersionDoc 定義をアップデートします。

構文

```
<STUDYVERSION
  UUID="id"
  STUDYNAME="name"
  VERSIONDESCRIPTION="text"
  [PROTOCOL="name"]
  [SPONSORDATE="date"]
  [TRADEDRUGNAME="name"]
  [GENERICDRUGNAME="name"]
  [SPONSORDRUGNAME="name"]>
  <FORMSET+ attributes/>
</STUDYVERSION>
```

属性

UUID="id"

Universally Unique Identifier; すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

STUDYNAME="name"

スタディの名前。必須

VERSIONDESCRIPTION="text"

スタディバージョンの説明。例えば、このバージョンでの変更内容の説明。必須

PROTOCOL="name"

IRB によって承認されたスタディプロトコルの名前とバージョン番号。オプション

SPONSORDATE="date"

スタディバージョンのスポンサー承認日付（必要な場合）。オプション

TRADEDRUGNAME="name"

スタディが行われる薬剤の商号。オプション

GENERICDRUGNAME="name"

スタディが行われる薬剤の一般名。オプション

SPONSORDRUGNAME="name"

その薬剤のスタディを依頼しているスポンサーの名前。オプション

子

StudyVersion 定義は 1 つ以上の FormSet 定義を含む必要があります。各フォームセットは、1 つのビジットまたは他のフォームグループを定義し、そのセットに含まれるフォームを指定します。フォームセットは再使用できないため、これらはスタディバージョンの定義内で定義されます。

特記事項：StudyVersion 定義に 1 つ以上の共通 CRF を含む場合は、以下のことに注意してください。**COMMONCRF** フォームセットに共通 CRF を含めることで共通 CRF を定義し、いずれかの症例のデータが入力された場合、StudyVersion 定義の **COMMONCRF** フォームセットから共通 CRF を削除しても、フォームを通常の CRF に戻すことはできません。同様に、症例のデータが入力された後に、通常の **VISIT** フォームセットにある CRF を **COMMONCRF** フォームセットに追加して、共通 CRF に変更することはできません。これらの方法でスタディバージョンを変更しようとすると、症例データを失うことになります。

既存の共通 CRF と同じデータを取り込む通常の CRF を作成する必要がある場合、共通 CRF とは異なる **RefName** を持つ別の **Form** 定義として CRF を作成し、スタディバージョンの中の該当する **VISIT** フォームセットに追加してください。

例

以下の例は、次のような 4 つのフォームセットを含んでいる StudyVersion 定義の第 2 バージョンを示しています。

- 第 4 週、第 2 週および第 0 週のビジット
- スクリーニングログを含んでいるスクリーニングセット

```
<STUDYVERSION
  VERSIONDESCRIPTION="2"
  STUDYNAME="Hypertension Study"
  PROTOCOL="Protocol XYZZY">
  <FORMSET REFNAME="Visit1" TITLE="Week -4"
    LANGUAGE="English"TYPE="Visit"
    SCHEDULED="true"ORDER="1">
    <FORMREF REFNAME="DEM"/>
    <FORMREF REFNAME="FH"/>
  </FORMSET>
  <FORMSET REFNAME="Visit2" TITLE="Week -2"
    TYPE="Visit"SCHEDULED="true">
    <FORMREF REFNAME="VS"/>
  </FORMSET>
  <FORMSET REFNAME="Visit3" TITLE="Week 0"
    TYPE="Visit"SCHEDULED="true">
    <FORMREF REFNAME="VS"/>
  </FORMSET>
  <FORMSET REFNAME="screen" TITLE="Screening Log"
    UUID="d882ce38-0f42-11d2-a419-00a0c963e0ac"
    TYPE="SCREENING"SCHEDULED="false">
```

```
<FORMREF REFNAME="screen"/>
</FORMSET>
```

SVCriticalForm

目的

SDV 対象フォームリストにフォームを追加します。

構文

```
<SVCRITICALFORM
  FORMREFNAME="name"
  SITEMNEMONIC="name"
  CRITICAL="true|false"
</SVCRITICALFORM>
```

属性

FORMREFNAME="name"

SDV 対象フォームリストに追加するフォームの RefName。必須

SITEMNEMONIC="name"

フォームが関連付けられている施設の略称。必須

CRITICAL="true|false"

フォームが SDV 対象としてマークされているかどうかを示します。オプションは true または false です。必須

SVCriticalItem

目的

ITEM の SVCRITICAL 設定を上書きします。

構文

```
<SVCRITICALITEM
  ITEMREFNAME="name"
  CRITICAL="study default|true|false"
  FORMREFNAME="name"
  SITEMNEMONIC="name"
</SVCRITICALITEM>
```

属性

ITEMREFNAME="name"

SDV 対象としてマークするアイテムの RefName。必須

CRITICAL="study default|true|false"

アイテムを SDV 対象としてマークするかどうかを示します。オプションはスタディのデフォルト、または true か false です。必須

スタディのデフォルトオプションでは、アイテムの事前定義済の上書きをリセットでき

ます。CRITICAL 属性にスタディのデフォルトを指定した場合、ITEM エLEMENTの SDVCRITICAL 属性がそのアイテムの SDV 対象設定を決定します。

FORMREFNAME="name"

SDV 対象としてマークするアイテムが存在するフォームの RefName。オプション

SITEMNEMONIC="name"

アイテムが関連付けられている施設の略称。オプション

- すべての施設を通じてアイテムを SDV 対象としてマークするには、SVCriticalITEM エLEMENTを使用し、SITEMNEMONIC は指定しないでください。
- 1 つの施設にのみアイテムを SDV 対象としてマークするには、次のいずれかを実行します。
 - SVCritical エLEMENTを使用し、SITEMNEMONIC を指定する。
 - ITEM の SVCritical 設定を使用する。
 - InForm ユーザインターフェイス を使用する。

SysConfig

目的

InForm 構成変数の設定を指定します。

構文

```
<SYSCONFIG
  CONFIGNAME="name"
  TYPE="0"
  VALUE="text" />
```

属性

CONFIGNAME="name"

構成変数の名前。必須

InForm アプリケーションは次の構成変数を持っています。

- **AllowPasswordReuse** — 1 (はい) もしくは 0 (いいえ)。ユーザがパスワード更新を行う場合に、以前使用していたパスワードに変更できるかどうかを示します。デフォルトは 1 です。
- **AutoAnswerManualQueries** — 1 (オン) もしくは 0 (オフ)。データアイテム変更がデータアイテムルールを満たすと、InForm アプリケーションが自動的にマニュアルクエリに回答するかどうかを示します。デフォルトは 1 です。
- **CookServer** — MedML メタデータ定義のインストールに使用されるサーバの名前。
- **DaysPasswordExpiration** — InForm ソフトウェアがユーザにパスワードの変更を要求するまでの日数。デフォルトは 30 です。
- **DISPLAYTRAFFICLIGHTS** — 1 (はい) もしくは 0 (いいえ)。信号アイコンを

ステータスインジケータとして使用するかどうかを示します。読み取り専用

- **EnableForgotPassword** — 1 (はい) もしくは 0 (いいえ)。ユーザが自分のパスワードを忘れてしまった場合に、ユーザからのパスワードリセット要求機能を有効にするかどうかを示します。デフォルトは 1 です。
- **EmailForForgotPasswordNotification** — ユーザからのパスワードリセット要求があったときに通知を受ける管理者の電子メールアドレス。
- **EmailForNewSiteAndUserNotification** — 新しい施設または新しいユーザが追加される場合に、通知を受ける管理者の電子メールアドレス。
- **EnforceVisitDate** — 1 (はい) もしくは 0 (いいえ)。すべてのビジットの最初のフォームに「ビジットの日付」を使用する必要があるかどうかを示します。デフォルトは 0 です。読み取り専用
- **EnrollWithIncompleteForms** — 1 (はい) もしくは 0 (いいえ)。InForm アプリケーションが、上書き許可後にスクリーニングまたは登録情報が不完全な症例の登録を許可するかどうかを指定します。デフォルトは 0 です。
- **ExePlanServer** — 実行計画が実行されるサーバとして定義されるサーバの名前。
- **FORMSAVEMODE** — 「フォームが正常にサブミットされました」というメッセージを表示する場所と形式。
 - インライン：メッセージはフォームのヘッダーに表示されます。
 - ポップアップ：メッセージはポップアップとして表示されます。続行するには、[OK] をクリックする必要があります。デフォルト。

- **InactivateRetryCount** — ユーザアカウントが無効になるまでのログイン失敗件数。デフォルトは 3 です。
- **INLINEDURATION** — 「フォームが正常にサブミットされました」というメッセージがフォームのヘッダーに表示されてから消えるまでの秒数 (1 ~ 9 秒) を指定します。FORMSAVEMODE がインラインに設定されている場合に適用されます。
- **LDAPBDN** — InForm およびレポーティング認証に使用される LDAP 階層のエントリ。
- **LDAPSERVER** — Sun ONE ディレクトリサーバがインストールされるサーバの名前。
- **MaxNumOfResubmissions** — 失敗した実行計画のサブミットを再試行できる最大回数。この回数を上回ると、エラーとしてイベントログに記録され、実行計画のキューから削除されます。デフォルトは 2 です。
- **MinPasswordLength** — パスワードに必要とする最小文字数。デフォルトは 6 です。
- **MinutesReauthenticate** — InForm ソフトウェアで、ユーザに再度ログインを要求するまでの認証不要な期間 (分単位)。デフォルトは 5 です。
- **MinutesReIdentification** — InForm ソフトウェアによって、ユーザが再ログインを要求されるまでのセッション有効期間 (分単位)。デフォルトは 120 です。
- **NumOfExePlanListenThreads** — 残りの実行計画を処理するために、バックグラウンドで実行されるスレッド数。デフォルトは 4 ですが、実行計画を完了するには最低でも 1 は必要です。
- **OneNonAlphaNumericCharacter** — 1 (はい) もしくは 0 (いいえ)。パスワードに少なくとも 1 つの特殊文字を含む必要があるかどうかを示します。デフォルトは 0 です。
- **OneNumericalCharacter** — 1 (はい) もしくは 0 (いいえ)。パスワードに少なくとも 1 つの数字を含む必要があるかどうかを示します。デフォルトは 0 です。
- **OneUppercaseCharacter** — 1 (はい) もしくは 0 (いいえ)。パスワードに少なくとも 1 つの大文字を含む必要があるかどうかを示します。デフォルトは 0 です。
- **PatientSequence** — 症例番号を割り当てるためのフォーマット。読み取り専用
- **PostQueryForConflictResolution** — 1 (はい) もしくは 0 (いいえ)。同期中に、2 つの異なるサーバから 1 つのデータアイテムに入力されるデータが見つかった場合、クエリを作成するかどうかを示します。デフォルトは 1 です。
- **QueryMaxLength** — CRF 上でアイテムの下に表示されるクエリテキストの最大文字数。デフォルトは 350 です。
- **QUERYSELECTION** — クエリを未解決または仮登録の状態で作成するかどうかを指定します。
- **RandCentralStratified** — セントラル階層化無作為割付計画のためのシーケンス番号フォーマット (複数の薬剤キットリスト。症例には階層化基準に基づいて薬剤キットが割り当てられます)。
- **RandomizationSrc** — デフォルト無作為割付ソースデータベースにアクセスする無作為割付ソースマネージャ (COM オブジェクト) の名前。デフォルト名は

「Inform.PFRandomization.1」です。読み取り専用

- **RandSimpleCentral** — シンプルセントラル無作為割付計画用シーケンス番号フォーマット (連続した番号が割り当てられたセントラル薬剤キットリスト。読み取り専用)
- **RandSimpleCentralSRC** — シンプルセントラル無作為割付計画のために無作為割付ソースデータベースにアクセスする無作為割付ソースマネージャ (COM オブジェクト) の名前。読み取り専用
- **RandSimpleSite** — シンプル施設無作為割付計画のためのシーケンス番号フォーマット (各施設用の薬剤キットリスト。症例には、これらの施設のリストにある薬剤キットが順に割り当てられます)。読み取り専用
- **RandSimpleSiteSRC** — シンプル施設無作為割付ソースデータベースにアクセスする無作為割付ソースマネージャ (COM オブジェクト) の名前。読み取り専用
- **RandStratifiedBySite** — 施設階層化無作為割付計画のためのシーケンス番号フォーマット (各施設に対する複数の薬剤キットリスト。症例には、施設と階層化基準に基づいて薬剤キットが割り当てられます)。読み取り専用
- **RandStratifiedBySiteSRC** — 施設階層化無作為割付ソースデータベースにアクセスする無作為割付ソースマネージャ (COM オブジェクト) の名前。読み取り専用
- **REPORTINGINTERNALURI** — CRNConfig インストーラの実行時に設定される Cognos パラメータです。これは InForm サーバが Cognos サーバとの通信に使用する内部 URI です。情報は、cogstartup.xml に含まれています。たとえば、次のようになります: `http://appsru02.north.pf.com:9300/p2pd/servlet/dispatch`
- **REPORTINGSERVER** — Cognos 8・BI Web サービスの URL。
- **REPORTINGAUTHENTICATIONNAMESPACE** — レポートサーバーで、InForm ユーザの認証に使用される LDAP 名前空間。
- **REPORTINGUSERROOT** — 企業の最上位レポートフォルダ。1 つのレポートサーバーに複数の企業をホスト中で、各企業のレポートフォルダをセットアップしてある場合は、このフィールドのみを使用します。異なる企業のスタディをホストしていない場合、または各企業に対し個別のレポートフォルダをセットアップしていない場合、このフィールドを空白のままにします。
- **RequireCommentForNA** — 1 (はい) もしくは 0 (いいえ)。フォーム上で質問への応答に「該当なし」、「不明」、または「未実施」が入力されたときに、InForm システムからユーザにコメントの入力を求めるかどうかを示します。デフォルトは 0 です。
- **ScreeningSequence** — スクリーニング番号を割り当てるためのシーケンス番号フォーマット。読み取り専用
- **SponsorEditFrozen** — 1 (はい) もしくは 0 (いいえ)。フォームがフリーズ済とマークされた後にスポンサーが編集できるかどうかを示します。デフォルトは 0 です。
- **SSLFlag** — 1 (はい) もしくは 0 (いいえ)。データの暗号化のため SSL を有効にするかどうかを示します。デフォルトは 1 です。

注: このオプションを有効にするには、スタディを停止して再起動する必要があります。

- **TrialDateFormat** — 「Month_Day_Year」、「Day_Month_Year」、もしくは

「Year_Month_Day」。スタディ内に日付を表示するときの形式を示します。
デフォルトは「Month_Day_Year」です。

- **UNC_DownloadDirectory** — [リスティング] ボタンでデータリスティングをダウンロードするときの物理 InForm サーバディレクトリの完全パス名。読み取り専用
- **UniqueIntIDOBSwitch** — スタディ、施設、または、なし。スタディまたは施設について、InForm アプリケーションが、症例のイニシャルと生年月日の一意の組み合わせを要求するか、またはまったく要求しないかを示します。
 - 0 (デフォルト) — イニシャルと生年月日の組み合わせは、一意である必要はありません。
 - 1 — 施設内でイニシャルと生年月日の組み合わせが一意である必要があります。
 - 2 — 臨床試験内でイニシャルと生年月日の組み合わせが一意である必要があります。

注：一意のイニシャルと生年月日を不要と指定しているときに、重複したイニシャルと生年月日を持つ症例が入力され、その後、一意の **ID** を必須と指定した場合、以前入力された重複情報は無視されます。別の施設への症例の移動を許可しようと予定している場合には、次のことに注意する必要があります。同じイニシャルと生年月日を持つ別の症例が存在する施設に症例を移動する場合で、そのスタディが一意のイニシャルと生年月日を必要としない、または施設の一意性のみを必要とするときには、症例の移動は失敗します。症例のイニシャルを変更して一意の組み合わせにする必要があります。

この状況を回避するには、**UniqueIntIDOBSwitch** 属性を使用して、スタディ内で一意のイニシャルと生年月日が必要であると設定します。

- **UniquePatIDSwitch** — スタディ、施設、または、なし。スタディまたは施設について、InForm アプリケーションが、一意の症例 **ID** を要求するか、またはまったく要求しないかを示します。
 - 0 — 症例 **ID** は、一意である必要はありません。
 - 1 — 施設内で症例 **ID** が一意である必要があります。
 - 2 (デフォルト) — スタディ内で症例 **ID** が一意である必要があります。

備考：

InForm アプリケーションでデータビューアを使用する場合は、各施設または各スタディ内で一意の症例 **ID** を要求するようにシステムを設定する必要があります。

一意のイニシャルと生年月日を不要と指定しているときに、重複したイニシャルと生年月日を持つ症例が入力され、その後、一意の **ID** を必須と指定した場合、以前入力された重複情報は無視されます。別の施設への症例の移動を許可しようと予定している場合には、次のことに注意する必要があります。同じイニシャルと生年月日を持つ別の症例が存在する施設に症例を移動する場合で、そのスタディが一意のイニシャルと生年月日を必要としない、または施設の一意性のみを必要とするときには、症例の移動は失敗します。症例のイニシャルを変更して一意の組み合わせにする必要があります。

この状況を回避するには、**UniqueIntIDOBSwitch** 属性を使用して、スタディ内で一意のイニシャルと生年月日が必要であると設定します。

- **ViewCRFSignList**—1 (はい) もしくは 0 (いいえ)。署名が必要な各 CRF に、必須署名

リストを表示するかどうかを示します。デフォルトは 1 です。読み取り専用

- **Virtual_DownloadDirectory** — [リスティング] ボタンでデータリスティングをダウンロードするとき使用される InForm サーバ仮想ディレクトリの完全パス名。読み取り専用

注 : Server Friendly Name パラメータを設定できます。これは、[クエリ詳細] 画面や [署名詳細] 画面に表示されるユーザフレンドリサーバ名となり、InForm アプリケーションの Admin ユーザインターフェースからのみ設定できます。これに相当する MedML の変数はありません。

TYPE="0"

0 が現在入力可能な唯一の値です。必須

VALUE="n"

コンフィギュレーション変数に割り当てる値。必須

例

```
<SYSCONFIG CONFIGNAME="MinutesReauthenticate" TYPE="0" VALUE="5"/>
<SYSCONFIG CONFIGNAME="DaysPasswordExpiration" TYPE="0" VALUE="30"/>
<SYSCONFIG CONFIGNAME="MinutesReIdentification" TYPE="0" VALUE="120"/>
<SYSCONFIG CONFIGNAME="MinPasswordLength" TYPE="0" VALUE="6"/>
<SYSCONFIG CONFIGNAME="InactivateRetryCount" TYPE="0" VALUE="3"/>
<SYSCONFIG CONFIGNAME="OneNumericalCharacter" TYPE="0" VALUE="0"/>
<SYSCONFIG CONFIGNAME="OneUppercaseCharacter" TYPE="0" VALUE="0"/>
<SYSCONFIG CONFIGNAME="OneNonAlphaNumericCharacter" TYPE="0" VALUE="0"/>
<SYSCONFIG CONFIGNAME="AllowPasswordReuse" TYPE="0" VALUE="1"/>
<SYSCONFIG CONFIGNAME="EnableForgotPassword" TYPE="0" VALUE="1"/>
<SYSCONFIG CONFIGNAME="EmailForForgotPasswordNotification" TYPE="0" VALUE=""/>
<SYSCONFIG CONFIGNAME="EmailForNewSiteAndUserNotification" TYPE="0" VALUE=""/>
<SYSCONFIG CONFIGNAME="EnrollWithIncompleteForms" TYPE="0" VALUE="0"/>
<SYSCONFIG CONFIGNAME="NumOfExePlanListenThreads" TYPE="0" VALUE="4"/>
<SYSCONFIG CONFIGNAME="MaxNumOfResubmissions" TYPE="0" VALUE="2"/>
<SYSCONFIG CONFIGNAME="RequireCommentForNA" TYPE="0" VALUE="0"/>
<SYSCONFIG CONFIGNAME="SSLflag" TYPE="0" VALUE="1"/>
<SYSCONFIG CONFIGNAME="RandomizationSrc" TYPE="0"
VALUE="Inform.PFRandomization.1" />
<SYSCONFIG CONFIGNAME="RandSimpleCentral" TYPE="0" VALUE="SC:RND-%q"/>
<SYSCONFIG CONFIGNAME="AutoAnswerManualQueries" TYPE="0" VALUE="1"/>
<SYSCONFIG CONFIGNAME="QueryMaxLength" TYPE="0" VALUE="80"/>
<SYSCONFIG CONFIGNAME="EnforceVisitDate" TYPE="0" VALUE="0"/>
<SYSCONFIG CONFIGNAME="DEFAULTTCPMAXIMIZED" VALUE="1"/>
<SYSCONFIG CONFIGNAME="SPONSOREDITFROZEN" VALUE="0"/>
<SYSCONFIG CONFIGNAME="UNIQUEPATIDSWTCH" VALUE="2"/>
<SYSCONFIG CONFIGNAME="UNIQUEINTLDOBSWTCH" VALUE="0"/>
<SYSCONFIG CONFIGNAME="PostQueryForConflictResolution" VALUE="1"/>
<SYSCONFIG CONFIGNAME="TrialDateFormat" VALUE="MONTH_DAY_YEAR"/>
<SYSCONFIG CONFIGNAME="NavigationMode" VALUE="0"/>
<SYSCONFIG CONFIGNAME="FORMSAVEMODE" VALUE="1"/>
<SYSCONFIG CONFIGNAME="INLINEDURATION" VALUE="3"/>
<SYSCONFIG CONFIGNAME="ViewCRFSignList" VALUE="1"/>
<SYSCONFIG CONFIGNAME="QUERYSELECTION" VALUE="QUERYACTION_CREATEOPEN"/>
```

翻訳

目的

スタディコンポーネントの属性に関連付けられた翻訳済みテキスト文字列を、翻訳テキストが使用されるロケールと一緒に指定します。スタディバージョンで必要とされる属

性とロケールの各組み合わせに対して Translation エlement を 1 つ使用してください。属性を翻訳する場合は、スタディコンポーネント定義の一部であるすべての翻訳可能な属性の翻訳定義を含めます。

Translation Element は、Translations Element の子です。

構文

```
<TRANSLATION>
  NAME="attributename"
  LOCALE="text"
  DISPLAYTEXT="text"
/>
</TRANSLATION>
```

属性

NAME="attributename"

翻訳済みテキストが用意されたスタディコンポーネントの属性の名前。必須

LOCALE="text"

翻訳済みテキストの言語とカルチャのコード。必須

DISPLAYTEXT="text"

翻訳済みテキスト文字列。必須

例

以下の例は、スペイン語とドイツ語に翻訳されたアイテム質問とラベルを示しています。

```
<ITEM refname="itmCardio"
  LANGUAGE="en-US"
  QUESTION="Cardiovascular"
  LABEL="Cardio"
  CALCULATED="false"
  ITEMREQUIRED="true"
  SDVREQUIRED="true"
  <CONTROLREF REFNAME="PEFstatus"/>
  <TRANSLATIONS>
    <TRANSLATION NAME="QUESTION" LOCALE="es-ES" DISPLAYTEXT="Sistema
cardiovascular"/>
    <TRANSLATION NAME="QUESTION" LOCALE="de-DE"
  DISPLAYTEXT="Herz-Kreislauf-System"/>
    <TRANSLATION NAME="LABEL" LOCALE="es-ES" DISPLAYTEXT="Cardiovascular"/>
    <TRANSLATION NAME="LABEL" LOCALE="de-DE" DISPLAYTEXT="Herz-Kreislauf"/>
  </TRANSLATIONS>
</ITEM/>
```

Translations

目的

スタディコンポーネントの翻訳可能な属性の翻訳済み文字列のセットを定義します。Translations 定義は、翻訳済み文字列を含んだスタディコンポーネントの子です。Translations エlementは、スタディコンポーネント定義内の最後のチャイルドElementにする必要があります。属性を翻訳する場合は、スタディコンポーネント定義の一部であるすべての翻訳可能な属性の翻訳定義を含めます。

注：Translations Elementを使用する場合は、関連付けられているスタディコンポーネントの **LANGUAGE** 属性が必要です。

構文

```
<TRANSLATIONS>
  <TRANSLATION* attributes/>
</TRANSLATIONS>
```

子

Translations Elementは 1 つ以上の Translation 定義を含む必要があります。Translation Elementはそれぞれ、スタディコンポーネントの 1 つの属性を、あるロケールの言語に変換することを意味します。

例

以下の例は、スペイン語とドイツ語に翻訳されたアイテム質問とラベルを示しています。

```
<ITEM refname="itmCardio"
  LANGUAGE="en-US"
  QUESTION="Cardiovascular"
  LABEL="Cardio"
  CALCULATED="false"
  ITEMREQUIRED="true"
  SDVREQUIRED="true"
  <CONTROLREF REFNAME="PEFstatus"/>
  <TRANSLATIONS>
    <TRANSLATION NAME="QUESTION" LOCALE="es-ES" DISPLAYTEXT="Sistema
cardiovascular"/>
    <TRANSLATION NAME="QUESTION" LOCALE="de-DE"
DISPLAYTEXT="Herz-Kreislauf-System"/>
    <TRANSLATION NAME="LABEL" LOCALE="es-ES" DISPLAYTEXT="Cardiovascular"/>
    <TRANSLATION NAME="LABEL" LOCALE="de-DE" DISPLAYTEXT="Herz-Kreislauf"/>
  </TRANSLATIONS>
</ITEM/>
```

Unit

目的

データ入力単位のタイプを指定します。Unit 定義には、InForm アプリケーションがレポートまたは統計解析のために標準単位へ変換する際に使用する変換ルールへの参照が含まれます。

構文

```
< UNIT
```

```

REFNAME="name"
[DESIGNNOTE="text"]
[UUID="id"]
SYMBOL="text"
CLASSIFICATION="text"
[LANGUAGE="name"]
BASEREFNAME="name"
[CONVERSIONTOBASE="name"]
[CONVERSIONFROMBASE="name"]>
  <TRANSLATIONS/>
</UNIT>

```

属性

REFNAME="name"

他のコントロールの定義内で単位を参照するときに使用される名前。この名前は単位間で一意である必要があります。必須

注： **TYPE="STRING"** と定義されたテキストコントロールには、単位を設定できません。
TYPE="INTEGER" または **"FLOAT"** と定義されたテキストコントロールに対して単位を設定できます。

DESIGNNOTE="text"

コンポーネントの設計に関して取り込む情報。最大 255 文字のテキスト文字を入力できます。設計情報のみとして使用され、表示されません。オプション

UUID="id"

Universally Unique Identifier; すべてのスタディ、スタディデータベースおよびコンピュータにおいてコンポーネントを一意に識別する文字列。オプション

以下の単位定義には、必須ではない事前定義された UUID が含まれています。

単位	UUID
BPDIAS	PF_BP_DIAS
BPSYS	PF_BP_SYS
セルシウス	498100ff-e9df-11d1-9e60- 00a0c9769a33
センチメートル	498100f5-e9df-11d1-9e60- 00a0c9769a33
ファーレンハイト	498100fe-e9df-11d1-9e60- 00a0c9769a33
インチ	498100f2-e9df-11d1-9e60- 00a0c9769a33
キログラム	498100f8-e9df-11d1-9e60- 00a0c9769a33
ポンド	498100fc-e9df-11d1-9e60- 00a0c9769a33

SYMBOL="text"

画面に単位を表示する際に使用するラベル **TRANSLATIONS** 属性を使用して、属性値を翻訳することができます。必須

CLASSIFICATION="text"

単位表示される計測タイプ以下の分類が事前定義されています:

- 長さ
- 体重
- 体温
- 圧力
- 音量
- 面積
- 時間
- 頻度

必須

LANGUAGE="name"

テキスト値を表示するためのカルチャと言語のコード。**LANGUAGE** 属性を含めない場合、インストールされているプロダクトロケールがデフォルトになります。テキスト文字列が翻訳されている場合 (**TRANSLATIONS** エレメントがスタディコンポーネント定義のチャイルドとして含まれている場合) は必須。

注: デフォルトプロダクトロケールは変更が可能なため、デフォルトプロダクトロケールに依存することは推奨されません。

BASEREFNAME="name"

変換のベースとして使用されるコンポーネントの RefName。必須

CONVERSIONTOBASE="name"

入力値を基準値へ変換するために使用される VBScript 変換ルールの RefName。
オプション

CONVERSIONFROMBASE="name"

BASEREFNAME 属性で指定された単位からこの単位に切り替えるために利用する VBScript 変換ルールの RefName。現在使われていません。

子

スタディコンポーネントの翻訳可能な属性の翻訳文字列のセットを指定する 0 または 1 つの Translations 定義。Translation エレメントには 1 つ以上の Translation 定義を含める必要があります。Translation エレメントはそれぞれ、スタディコンポーネントの 1 つの属性のあるロケール言語への翻訳を表します。Unit 定義では、**SYMBOL** 属性を翻訳

できます。

例 1

この例では、「Inches（インチ）」の単位を定義しています。この例では、定義される単位と、変換のベースとして指定された単位の両方ともがインチになっています。したがって、**CONVERSIONTOBASE** 属性で指定される変換ルールでは、「1」が入力値に掛けられるだけです。

```
<UNIT REFNAME="INCHES" SYMBOL="IN" CLASSIFICATION="Length"
  BASEREFNAME="Inches" CONVERSIONTOBASE="UnitIsBase"
  UUID="498100f2-e9df-11d1-9e60-00a0c9769a33"/>
```

UnitIsBase 変換ルール：

```
Data.Result=Data.BaseValue*1
```

例 2

この例では、「Centimeters（センチメートル）」の単位を定義しています。この例では、変換のベースとして指定されている単位は「インチ」です。したがって、**CONVERSIONTOBASE** 属性で示される変換ルールにより、センチメートルからインチに変換するために使用される係数が提示されます。

```
<UNIT REFNAME="CENTIMETERS" SYMBOL="CM" CLASSIFICATION="Length"
  BASEREFNAME="Inches" CONVERSIONTOBASE="CmToInches"
  UUID="498100f5-e9df-11d1-9e60-00a0c9769a33"/>
```

CmToInches 変換ルール：

```
Data.Result=Data.Basevalue*.3937
```

Unitref

目的

別のコントロールの定義内に `unit` 定義を含めます。Unitref は自身が含まれているコントロールの子としてのみ定義され、単独のコンポーネントとしてサブMITトされません。Unitref は、以下のタイプのどのコントロールにも含めることができます。

- CalculatedControl
- PullDownControl
- SimpleControl
- TextControl

構文

```
<UNITREF
  REFNAME="name"/>
```

属性

REFNAME="name"

Unitref がコントロール定義に含める単位の RefName。必須

注 : **Controlref** 定義を用いてコンパウンドコントロールを定義する場合には、すべての下位コントロールが同じ **TYPE** 属性で定義され、同じデータタイプを返すようにしてください。

さらに、コンパウンドコントロールを定義する際には、**InForm** ソフトウェア が最大 **5** 層までのネスティングをサポートすることに注意してください。**5** 層での設計が可能です、パフォーマンスを向上させるためにネストレベルはできるだけ少なくしてください。

例

この例では、ポンドとキログラムの Unit 定義が、TextControl 定義に Unitref エレメントを使用することで含まれています。

- 1 ポンドとキログラムの Unit 定義を作成します。以下の例において、キログラムは変換のベース単位です。

```
<UNIT REFNAME="KILOGRAMS" SYMBOL="KG" CLASSIFICATION="Weight"
  BASEREFNAME="KILOGRAMS" CONVERSIONTOBASE="UnitIsBase"
  UUID="498107f2-e9df-11d1-9e60- 00a0c9769a33"/>
<UNIT REFNAME="POUNDS" SYMBOL="LB" CLASSIFICATION="Weight"
  BASEREFNAME="KILOGRAMS" CONVERSIONTOBASE="LbToKg"
  UUID="498109f5-e9df-11d1-9e60- 00a0c9769a33"/>
```

- 2 ポンドとキログラムの Unit 定義を参照する Unitref エレメントを使用する TextControl 定義を作成します。この例では、単位オプションはドロップダウンリストに表示されます。

```
<TEXTCONTROL REFNAME="txtVSWeight"
  LANGUAGE="English" NAME="txtVSWeight" ALIGN="LEFT" CAPTIONALIGN="LEFT"
  UNITDISPLAYTYPE="RADIO" HEIGHT="1" LENGTH="6" MAXLENGTH="5" DATATYPE="FLOAT"
  PRECISION="1" MINVALUE="0" MAXVALUE="0" MINPROPERTY="NOTSPECIFIED"
  MAXPROPERTY="NOTSPECIFIED">
  <UNITREF REFNAME="POUNDS" />
  <UNITREF REFNAME="KILOGRAMS" />
```

</TEXTCONTROL>

Update_Form_Section

目的

スタディバージョンを変更することなく、フォーム内のセクションを更新します。この XML は、フォーム上の既存セクションを変更するために使用されます。新しいセクションの追加用には使用できません。症例データが既にこのセクションに存在している場合、そのデータはそのセクションの古いバージョンに関連付けられます。更新後に入力されたデータは、そのセクションの新しいバージョンに関連付けられます。

注： 更新する CRF とセクションの RefName を決定する際は、Oracle では Central Designer ソフトウェア で InForm RefName レポートを実行することを推奨しています。詳細については、『Central Designer InForm デザイナガイド』を参照してください。

このエレメントによりアイテムを削除した場合、データはデータベースから削除されるのではなく、フォーム上に表示されなくなるだけです。この方法でフォームアイテムを削除するよりも、スタディのスタディバージョンを更新することによって、フォームアイテムを削除の方が適切です。

構文

```
<METADATA>
<UPDATE_FORM_SECTION
  FORM_REFNAME="name"
  FORM_REVISION="number"
  SECTION_REFNAME="name"
  SECTION_REVISION="number"/>
</METADATA>
```

属性

FORM_REFNAME="name"

変更するフォームの RefName。必須

FORM_REVISION="number"

変更するフォームの改訂番号。必須

SECTION_REFNAME="name"

変更するセクションの RefName。必須

SECTION_REVISION="number"

変更するセクションの改訂番号。必須

Update_Form_Section エレメントを使用したセクション定義の更新

- 1 スタディを停止して、スタディへのアクセスやスタディデータの変更ができなくなっていることを確認します。

- 2 変更するセクションの XML に対して、必要な変更を行います。
- 3 MedML インストーラユーティリティを使用して、セクションの新しいバージョンをインストールします。
- 4 UPDATE_FORM_SECTION 用 XML を作成します。SQL を使用して、新しいセクション改訂番号または古いセクション改訂番号を検索できます。また、データベース内を直接検索することもできます。
- 5 FORM_REVISION 番号については、PF_PAGE テーブルを開き、対応する PAGEREFFNAME の最新 (最も大きい数) の PAGEREVISIONNUMBER を検索します。SECTION_REVISION 番号については、PF_SECTION テーブルを開き、対応する SECTIONREFNAME の最新 (最も大きい数) の SECTIONREVISIONNUMBER を検索します。
- 6 MedML インストーラユーティリティを実行して、「UPDATE__FORM__SECTION」XML を適用します。MedML インストーラユーティリティは、フォームの RefName および SQL を使用して、リビジョンが正しいこと、およびそのフォームに、更新したセクションが含まれていることを検証します。
- 7 スタディを再開します。
- 8 この更新適用後、更新されたフォームのバージョンをテストします。MedML インストーラユーティリティは、既存の症例データに対しては、アイテムメタデータの検証を行いません。

Update_Section_Item

目的

スタディバージョンを変更することなく、セクション内のアイテムを更新します。この XML は、既存のアイテムを変更するのに使用されます。新しいアイテムの追加には使用できません。症例データがこのアイテムに既存する場合、このデータはそのアイテムの古いバージョンに関連付けられます。更新後入力したデータは、アイテムの新しいバージョンに関連付けられます。

注： 更新するセクションとアイテムの **RefName** を決定する際は、**Oracle** では **Central Designer** ソフトウェア で **InForm RefName** レポートを実行することを推奨しています。詳細については、『**Central Designer InForm デザイナガイド**』を参照してください。

警告:新しいスタディバージョンを作成せずにスタディ定義の変更を行うと、**Good Clinical Practice (GCP)** 違反となり、スタディデータに問題が生じる原因になります。

構文

```
<METADATA>
<UPDATE_SECTION_ITEM
  SECTION_REFNAME="name"
  SECTION_REVISION="number"
  ITEM_REFNAME="name"
  ITEM_REVISION="number"/>
</METADATA>
```

属性

SECTION_REFNAME="name"

変更するセクションの RefName。必須

SECTION_REVISION="number"

変更するセクションの改訂番号。必須

ITEM_REFNAME="name"

変更するアイテムの RefName。変更されるアイテムがアイテムセットに含まれる場合、この RefName はアイテムセットの RefName になります。必須

ITEM_REVISION="number"

変更するアイテムの改訂番号。変更されるアイテムがアイテムセットに含まれる場合、この改訂番号はアイテムセットの改訂番号になります。必須

Update_Section_Item エレメントを使用したセクション定義の更新

- 1 スタディを停止して、スタディへのアクセスやスタディデータの変更ができなくなっていることを確認します。
- 2 変更するアイテムの XML に対して必要な変更を行います。
- 3 MedML インストーラユーティリティを使用して、アイテムの新しいバージョンをインストールします。
- 4 UPDATE_SECTION_ITEM 用 XML を作成します。SQL を使用して、新しいアイテム改訂番号または古いアイテム改訂番号を検索できます。また、データベース内を直接検索することもできます。
- 5 SECTION_REVISION 番号については、PF_SECTION テーブルを開き、対応する SECTIONREFNAME について最新 (最も大きい数) の SECTIONREVISIONNUMBER を探します。ITEM_REVISION 番号については、PF_ITEM テーブルを開き、対応する ITEMREFNAME について最新 (最も大きい数) の ITEMREVISIONNUMBER を探します。

注： 変更しているアイテムが **itemset** に含まれる場合ここに入力する **RefName** と改訂番号は **itemset** のためのものです。**Itemset** の **RefName** と改訂番号も **PF_ITEM** テーブルに保存されています。

- 6 MedML インストーラユーティリティを実行して、「UPDATE_SECTION_ITEM」XML

を適用します。MedML インストーラユーティリティは、CRF の RefName および改訂が正しいこと、およびその CRF に、更新したアイテムが含まれていることを検証します。

- 7 スタディを再開します。
- 8 この更新適用後、更新されたフォームのバージョンをテストします。MedML インストーラユーティリティは、既存の症例データに対しては、アイテムメタデータの検証を行いません。

ユーザ

目的

InForm スタディデータベースにアクセスできるユーザを定義します。スタディの特定部分へのユーザアクセスは、そのユーザの次の属性によって決定されます。

- 施設グループ — ユーザに特定のスタディ施設へのアクセス権を付与します。
- 権限グループ — ユーザに、特定のアクティビティを実行する権限セットへのアクセス権を付与します。
- クエリグループ — クエリをクローズする権限を持つユーザに、同じクエリグループの別のメンバーが開始したクエリをクローズする追加の権限を付与します。
- 署名グループ — 署名が必要なドキュメントに署名する権限をユーザに付与します。施設のドキュメントに署名するには、ユーザは署名グループおよび適切な施設グループに属している必要があります。

Sponsor もしくは Site の定義にユーザを含めることも可能です。どちらかの定義にユーザが含まれているということは、そのユーザが依頼者または施設の連絡窓口であるということを意味します。

ユーザのパスワードを含む、既存のアクティブなユーザに関する情報をインストールしようとする、InForm アプリケーションは安全対策としてそのユーザを非アクティブにします。

注： **IMAGEFILE** 属性を使用する場合には、ユーザ画像を定義するリソース **XML** ファイルをロードした後で、データベースにスタディ定義ファイルをロードする必要があります。

構文

```
<USER
  USERNAME="name"
  USERTYPE="SYSTEM|SITE|SPONSOR"
  [FIRSTNAME="name"]
  [LASTNAME="name"]
  [DISPLAYNAME="name"]
  [DESCRIPTION="text"]
  [TITLE="name"]
  [ADDRESS="addr1"]
  [ADDRESS2="addr2"]
  [CITY="name"]
  [STATE="name"]
  [PROVINCE="name"]
  [ZIPCODE="code"]
  [POSTCODE="code"]
  [COUNTRY="name"]
```

```
[PHONE="num"]
[ALTPHONE="num"]
[FAX="num"]
[EMAIL="addr"]
[BEEPER="num"]
[HOMESCREENURL="url"]
[IMAGEFILE="file"]
[IMAGETYPE="GIF|JPEG|TEXT"]
[LANGUAGE="name"]
[ACTIVESTATE="true|false"]
[DELETESTATE="true|false"]
[PASSWORD="name"]
[USERDATEFORMAT="MONTH_DAY_YEAR|DAY_MONTH_YEAR|YEAR_MONTH_DAY"]
PRODUCTLOCALE="en-US|ja-JP"
STUDYLOCALE="text"/>
```

属性

USERNAME="name"

データベース内でユーザを識別する名前。必須

USERTYPE="SYSTEM|SITE|SPONSOR"

ユーザタイプ。

- **SYSTEM** — 特殊システム機能を有するユーザ。たとえば、InForm アプリケーションが自動的にクエリを作成するとき、クエリ発生元に割り当てられるユーザ名は「autoquery」です。autoquery ユーザはシステムユーザです。
- **SITE** — 施設に関連付けられているユーザ。
- **SPONSOR** — 依頼者に関連付けられているユーザ。

必須

FIRSTNAME="name"

ユーザの名。オプション

LASTNAME="name"

ユーザの姓。オプション

DISPLAYNAME="name"

InForm アプリケーションでナビゲーションペインに表示されるユーザ名。最大の長さは 63 文字。文字列はより短いものを推奨します。オプション

DESCRIPTION="text"

ユーザの説明。たとえば、スタディでのユーザの役割。オプション

TITLE="name"

ユーザの肩書き。オプション

ADDRESS="addr1"

ユーザアドレスの第 1 ライン。オプション

ADDRESS2="addr2"

ユーザアドレスの第 2 ライン。オプション

CITY="name"

ユーザアドレスの市区町村。オプション

STATE="name"

ユーザアドレスの都道府県。（Province とは併用しません）オプション

PROVINCE="name"

ユーザアドレスの都道府県。（State とは併用しません）オプション

ZIPCODE="code"

ユーザのジップコード。（Postcode とは併用しません）オプション

POSTCODE="code"

ユーザの郵便番号。（Zipcode とは併用しません）オプション

COUNTRY="name"

ユーザアドレスの国。オプション

PHONE="num"

ユーザの電話番号。オプション

ALTPHONE="num"

ユーザの予備電話番号。オプション

FAX="num"

ユーザのファクス番号オプション

EMAIL="addr"

ユーザの電子メールアドレスオプション

HOMESCREENURL="url"

ユーザが InForm アプリケーションにログインする際表示される最初の画面を特定する内部あるいは外部の URL です。アドレスは http:// から始まり、ファイルが格納されているサーバを名前あるいは IP アドレスで指定する必要があります。オプション

IMAGEFILE="file"

InForm アプリケーションのナビゲーションツールバー上に表示されるイメージファイルの名前。オプション

IMAGETYPE="GIF|JPEG|TEXT"

イメージファイルのタイプ：GIF、JPEG、または TEXT。イメージファイルのファイル名を指定した場合は必須です。

LANGUAGE="name"

ユーザが指定した言語。デフォルトは英語。オプション

ACTIVESTATE="true|false"

ユーザがアクティブであるかどうかを示します。オプションは true または false です。デフォルトは、True になっています。オプション

DELETESTATE="true|false"

ユーザが削除されたかどうかを示します。オプションは true または false です。デフォルトは、False になっています。ユーザは削除されても、データベースにはそのまま残ります。オプション

PASSWORD="name"

ユーザのパスワード。オプション

USERDATEFORMAT="MONTH_DAY_YEAR|DAY_MONTH_YEAR|YEAR_MONTH_DAY"

この特定のユーザに対して表示する InForm アプリケーションページでの希望の日付フォーマット。

PRODUCTLOCALE="en-US|ja-JP"

ユーザのプロダクトロケールのコード。プロダクトロケールは、InForm ユーザインターフェイスにシステムデータが表示されときのロケールになります。値は「en-US」（英語）または「ja-JP」（日本語）になります。必須

STUDYLOCALE="text"

ユーザのスタディロケールのコード。スタディロケールは、スタディメタデータを定義するロケールであり、ビジット名、CRF 名、セクションラベル、質問、およびコントロールラベルを含みます。値は、そのスタディのスタディバージョン用スタディロケールと一致する必要があります。必須

UPDATE="true|false"

指定したプロパティのみを変更して、指定したユーザの増分更新を実行します。

例 1

以下の例では、データベースに CRC というユーザ名を追加します。

```
<USER USERNAME="crc"
  USERTYPE="SITE"
  FIRSTNAME="Clinical"
  LASTNAME="Research Coordinator"
  DISPLAYNAME="CRC"
  HOMESCREENURL="/inform1/custom/HomeDefault.html"
  IMAGEFILE="..\Resources\UserPics\StudyCoordinator75.gif"
  IMAGETYPE="GIF"
  ACTIVESTATE="false"
  DELETESTATE="false"
  PASSWORD="gcp"
  USERDATEFORMAT="MMM/DD/YYYY"/>
```

例 2

次の例は、データベースにスイスにいるユーザを加えます。

```
<USER USERNAME="jcrecy"
  USERTYPE="SITE"
```

```

FIRSTNAME="Jeanne"
LASTNAME="Crecy"
DISPLAYNAME="J Crecy"
HOMESCREENURL="/inform1/custom/Default.html"
IMAGEFILE="..\Resources\UtilsPhotos\coordinateur_détudes75.gif"
IMAGETYPE="GIF"
ACTIVESTATE="false"
DELETESTATE="false"
PASSWORD="gcp"/>
USERDATEFORMAT="DD/MMM/YYYY"
PRODUCTLOCALE="en-US"
STUDYLOCALE="fr-CH"/>

```

UserImage

目的

既存ユーザの画像を更新します。UserImage エlementを使用する場合、他のすべての既存のユーザ情報は変わりません。

注：他の個々のユーザ属性を更新する場合は、**InForm Admin** ユーザインターフェイスまたは **MedML** インストーラユーティリティをご利用ください。**MedML** インストーラユーティリティを使って **User** 定義を更新する場合、ユーザ情報セット全体を繰り返す必要があります。ユーザ画像以外については属性単位で更新を行うことはできません。

構文

```

<USERIMAGE
  USERNAME="name"
  [IMAGEFILE="file"]
  [IMAGETYPE="GIF|JPEG|TEXT"]
  [LANGUAGE="name"]/>

```

属性

USERNAME="name"

データベース内でユーザを識別する名前。必須

IMAGEFILE="file"

InForm アプリケーションのナビゲーションツールバー上に表示されるイメージファイルの名前。ファイル名を指定しない場合、そのユーザ用に定義されているイメージファイルがあれば、MedML インストーラユーティリティは現在のイメージファイルを削除します。オプション

IMAGETYPE="GIF|JPEG|TEXT"

イメージファイルのタイプ：GIF、JPEG、または TEXT。イメージファイルのファイル名を指定した場合は必須です。

LANGUAGE="name"

ユーザのイメージファイルの言語。デフォルトは英語。オプション

例

以下の例では、「sm」というユーザ名のユーザに、xena.jpg ファイル内の画像を追加す

るために、UserImage エlementが使用されています。

```
<USERIMAGE USERNAME="sm"
  IMAGEFILE="..\Resources\UserPics\xena.jpg"
  IMAGETYPE="GIF"/>
```

Userref

目的

以下のいずれかのコンポーネント定義に、事前定義されたユーザを含むことができます。

- QueryGroup
- RightsGroup
- SignatureGroup
- Site
- SiteGroup
- Sponsor

Userref は、自身が含まれるコンポーネントの子としてのみ表示されます。つまり、単独コンポーネントとしてはサブミットされません。

注：ユーザをグループから削除するには、InForm アプリケーションの **Admin** 機能を通して行う方法しかないことに注意してください。MedML インストーラユーティリティの実行によって、ユーザを削除することはできません。

構文

```
<USERREF
  USERNAME="name"/>
```

属性

USERNAME="name"

User 定義の **USERNAME** 属性で指定されている、コンポーネントに含めるユーザの名前。

例

以下の例は、SiteXQueries というクエリグループの定義へユーザ（Homer と Marge）を挿入する方法を示しています。

```
<QUERYGROUP GROUPNAME="SiteXQueries">
  <USERREF USERNAME="Homer"/>
  <USERREF USERNAME="Marge"/>
</QUERYGROUP>
```

VerbatimType

目的

コード化ディクショナリで定義されている特定の Verbatim のタイプを識別します。

構文

```
<VERBATIMTYPE  
  NAME="name"/>
```

属性

NAME="name"

Verbatim アイテムのタイプ。次の Verbatim タイプは、Central Coding アプリケーションでサポートするディクショナリ用に定義されています。

- **AE** — 有害事象
- **DISEASE** — 疾患
- **LABDATA** — ラボデータ
- **MEDPROD** — 医薬品

AE、DISEASE、および LABDATA は MedDRA ディクショナリの有効な verbatim タイプです。MEDPROD は WHO-DD ディクショナリの有効な verbatim タイプです。
オプション

例

```
<DICTIONARY TYPE="WHODD" VERSION="05Q4" CULTURE="en-US">  
  <CODETARGET NAME="ATC 1.CODE"/>  
  <CODETARGET NAME="ATC 1.TERM"/>  
  <CODETARGET NAME="ATC 2.CODE"/>  
  <CODETARGET NAME="ATC 2.TERM"/>  
  <CODETARGET NAME="ATC 3.CODE"/>  
  <CODETARGET NAME="ATC 3.TERM"/>  
  <CODETARGET NAME="ATC 4.CODE"/>  
  <CODETARGET NAME="ATC 4.TERM"/>  
  <CODETARGET NAME="Preferred Name.CODE"/>  
  <CODETARGET NAME="Preferred Name.TERM"/>  
  <CODETARGET NAME="Ingredients.AddInfo"/>  
  <CODETARGET NAME="Trade Name.CODE"/>  
  <CODETARGET NAME="Trade Name.TERM"/>  
  <CODETARGET NAME="Medicinal Product.CODE"/>  
  <CODETARGET NAME="Medicinal Product.TERM"/>  
  <CODETARGET NAME="Name Specifier.AddInfo"/>  
  <CODETARGET NAME="Country of Sale.AddInfo"/>  
  <CODETARGET NAME="MA Holder.AddInfo"/>  
  <CODETARGET NAME="MA Holder Country.AddInfo"/>  
  <CODETARGET NAME="Company.AddInfo"/>  
  <CODETARGET NAME="Company Country.AddInfo"/>
```

```
<CODETARGET NAME="ICH Med Prod ID.AddInfo"/>
<CODETARGET NAME="Sequence Number 3.AddInfo"/>
<CODETARGET NAME="Sequence Number 4.AddInfo"/>
<CODETARGET NAME="MA Number.AddInfo"/>
<CODETARGET NAME="MA Date.AddInfo"/>
<CODETARGET NAME="MA Withdrawal Date.AddInfo"/>
<CODETARGET NAME="Product Type.AddInfo"/>
<CODETARGET NAME="Product Group.AddInfo"/>
<CODETARGET NAME="Pharmaceutical Product.AddInfo"/>
<CONTEXTITEM NAME="Route Of Administration"/>
<CONTEXTITEM NAME="Indication"/>
<VERBATIMTYPE NAME="MEDPROD" />
</DICTIONARY>
```

スクリプトオブジェクトリファレンス

Conversionオブジェクト

InForm アプリケーションでは、単位として指定する CRF コンポーネントを定義できます。データをオンライン上で 1 つの単位として表示/データベースへ保存するために、他の単位を参照して単位 CRF コンポーネントを定義することができます。例えば、CRC がオンス単位かポンド単位のどちらかで体重を入力できるように CRF を設計し、データベース上ではオンスで保存されるように指定することも可能です。変換後にデータベースに保存したデータのユニットをベース単位と呼び、ベース単位に変換されたデータ値を正規化された値と呼びます。

入力された単位値とベース単位値間の変換を行うには、Conversion（変換）ルールタイプを用いてルールを作成します。これらのルール用のスクリプトでは、Data オブジェクトを使用できます。

Data オブジェクト

Data オブジェクトを使用することにより、アイテムに入力された値と、変換後にデータベースに格納される正規化された値について、変換方法を指定できます。変換ルールは、XML ファイルとして単位定義に添付されます（ルールの RefName を含みます）。

Data オブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	目的
Result	FLOAT	変換の結果を保持します。
BaseValue	NUMERIC	変換される値を含んでいます。

例

以下の変換ルールスクリプトは、センチメートルをミリメートルに変換します。

```
Data.Result=Data.BaseValue*10
```

実行計画のオブジェクトとメソッド

実行計画はイベントに関連付けられ、イベントが実行される時に動作します。InForm アプリケーションは、以下のいずれかを行う実行計画をサポートします。

- 電子メールを送る。
- Windows ログにメッセージを登録する。

実行計画の定義のスクリプトでは、このトピックで説明されているオブジェクトとメソッドを使用できます。

注: 実行計画は、MedML と MedML インストラユーティリティによってのみ定義できます。

Global オブジェクト

現在の施設、実行計画を実行したユーザ、または現在の施設で主要連絡先として指定されたユーザについての情報を取得できます。さらに、Global オブジェクトは、ルールにより生成されたメッセージを、電子メールメッセージまたは Windows ログに渡すこともできます。Global オブジェクトは以下のプロパティ（Get のみです）を持っています。

プロパティ	目的
Site	施設データを持つ Site オブジェクトの現在のプロパティを保持する。
SiteUser	現在の施設の主担当として指定されたユーザのデータを持つ User オブジェクトの、現在のプロパティを保持する。
User	実行計画を実行させたユーザのデータを持つ User オブジェクトの、現在のプロパティを保持する。
ContextString	Patient オブジェクトあるいは Result オブジェクトの Message プロパティから受け取ったメッセージを保持する。ContextString は string プロパティです。

Global メソッド

メソッド	目的
EMailToGroup	スタディデータベースに定義されているクエリグループ、署名グループ、あるいはユーザマネージャグループのメンバーに電子メールを送ります。
EMailToInFormUser	指定された InForm ユーザに電子メールを送ります。
EMailToInternetUser	電子メールをインターネットユーザに送ります。
EMailToUser	InForm 内部でのみ使用されるメソッドです。
GetNamedValue	Patient あるいは Result オブジェクトの AddNamedValue メソッドで指定された名前に関連するデータ値を返します。
LogMessage	指定されたメッセージを Windows ログに記録します。
OutputDebug	サポートされていません。

Siteオブジェクト

現在の施設に関する情報を得ることができます。Site オブジェクトは以下のプロパティ（Get のみです）を持っています。

プロパティ	タイプ	プロパティに保持される内容
ContactID	DWORD	現在の施設の主担当として指定された InForm ユーザのデータベース ID
SiteName	string	施設名
Address	string	施設の住所
Phone	string	施設の電話番号
AltPhone	string	施設の予備電話番号
Beeper	string	施設のポケットベル番号
Email	string	施設の電子メールアドレス
Fax	string	施設のファクス番号
TimeZone	string	施設のタイムゾーン

Site メソッド

メソッド	目的
SetSite	施設 ID で指定された施設のデータと共に Site オブジェクトをキャッシュからロードします。Oracle 専用。

Userオブジェクト

現在のユーザに関する情報を得ることができます。User オブジェクトは以下のプロパティ（Get のみです）を持っています。

プロパティ	タイプ	プロパティに保持される内容
FirstName	string	ユーザの名前
LastName	string	ユーザの姓
Address	string	ユーザの住所
Phone	string	ユーザの電話番号
AltPhone	string	ユーザの予備電話番号
Beeper	string	ユーザのポケットベル番号
Email	string	ユーザの電子メールアドレス

プロパティ	タイプ	プロパティに保持される内容
Fax	string	ユーザのファクス番号
UserName	string	InForm アプリケーションへのログインで使用するユーザアカウント名

User メソッド

メソッド	目的
SetUser	ユーザ ID で指定されたユーザのデータと共に User オブジェクトをキャッシュからロードします。Oracle 専用。

EMailToGroup(GroupID,Subject,Msg)

EMailToGroup は Oracle 内部でのみ使用可能です。

スタディデータベースに定義されているクエリグループ、署名グループ、あるいはユーザマネージャグループのメンバに電子メールを送ります。

引数

GroupID

電子メールを受け取るために内部的に割り当てられた、クエリグループ、署名グループ、またはユーザマネージャグループのデータベース ID。

症例

電子メールの件名を含む文字列。

Msg

電子メールメッセージのテキストを含む文字列。

注：GroupID 引数は、電子メールの受信者を特定します。以下のレジストリエントリが、送信者(From アドレス)を指定します。HKLM/SOFTWARE/OracleHS/InForm/PFMngrExecutionPlan。エントリは、「FromAddress」という名前の string 値です。デフォルトは、「nobody@Oracle.com」です。

EMailToInFormUser "InFormUser,szSubject,szMsg"

指定された InForm ユーザに電子メールを送ります。

引数

InForm ユーザ

電子メールを受け取るために内部的に割り当てられた InForm ユーザのデータベース ID。

症例

電子メールの件名を含む文字列。

Msg

電子メールメッセージのテキストを含む文字列。

注 : InFormUser 引数は、電子メールの受信者を特定します。以下のレジストリエントリが、送信者(From アドレス)を指定します。HKLM/SOFTWARE/OracleHS/InForm/PFMngrExecutionPlan。エントリは、「FromAddress」という名前の **string** 値です。デフォルトは、「nobody@Oracle.com」です。

例

Global.EMailToInFormUser "sm", "CRFUnLock", "Sending InForm User email when Unlock-CRF"

EMailToInternetUser "EmailAddress,szSubject,szMsg"

電子メールをユーザに送ります。

引数

EmailAddress

電子メールを受け取るための InForm ユーザの電子メールアドレス。

Subject

電子メールの件名を含む文字列。

Msg

電子メールメッセージのテキストを含む文字列。

注 : EmailAddress 引数は、電子メールの受信者を特定します。以下のレジストリエントリが、送信者(From アドレス)を指定します。HKLM/SOFTWARE/OracleHS/InForm/PFMngrExecutionPlan。エントリは、「FromAddress」という名前の **string** 値です。デフォルトは、「nobody@Oracle.com」です。

例

Global.EMailToInternetUser "user@Oracle.com", "CRFUnLock", "Sending External User email when Unlock-CRF"

GetNamedValue(Name)

Patient あるいは Result オブジェクトの AddNamedValue メソッドで指定された名前に関連するデータ値を返します。

引数

名前

AddNamedValue メソッドから渡された名前を含む文字列。

例

```
aeserval_m = Global.GetNamedValue("aeserval")
if aeserval_m = "Y" then
  ptinit_m = Global.GetNamedValue("ptinit")
  aedate_m = Global.GetNamedValue("aedate")
  aeevent_m = Global.GetNamedValue("aeevent")
  site_m = Global.GetNamedValue("sitenm")
  emailtxt="An SAE has occurred at Site:"&site_m&" for Patient:"&ptinit_m &" on AE
  Date:"&aedate_m&".The event is :"&aeevent_m&".
  call Global.EmailtoInternetUser("test@phaseforward.com","SAE Occurrence",emailtxt)
end if
```

LogMessage "Message"

指定されたメッセージを Windows ログに記録します。

引数

Message

Windows ログに記録するためのメッセージテキストを含むストリング。

例

```
Global.LogMessage "Query generated"
```

SetSite(SiteID)

SetSite(SiteID) は内部専用です。

施設 ID で指定された施設のデータと共に Site オブジェクトをキャッシュからロードします。

引数

SiteID

Site オブジェクトを使用してアクセスする施設のデータベース ID。

SetUser(UserID)

SetUser(UserID) は内部専用です。

ユーザ ID で指定されたユーザのデータと共に User オブジェクトをキャッシュからロードします。

引数

UserID

User オブジェクトを使用してアクセスするユーザのデータベース ID。

Randomization オブジェクトとメソッド

スタディで無作為割付を実行する時、いくつかの構成ステップが必要です。その一つは、症例 (症例) の登録時に、各症例 (症例) に割り当てる薬剤キット番号を発行するスクリプトを作成することです。この無作為割付スクリプトは、ルールや計算に使用できるオブジェクトやメソッドを使用することができるうえ、このトピックで説明する Randomization オブジェクトとメソッドを使用することもできます。

MedML と MedML インストラユーティリティを使用し、無作為割付シーケンス番号を発行して薬剤キットを割り当てる CRF 上の計算コントロールに無作為割付スクリプトを付加します。DRUGKITSECTION という定義済みのセクション定義が InForm インストールツリーの XMLBase フォルダ内の core_randomization.xml ファイルにあります。

Randomization オブジェクト

次の薬剤キット番号を割り当てるために GetNextKit メソッドで使用する症例 (症例)、施設、無作為割付、およびスタディ改訂の情報を割り当てることができます。

Randomization オブジェクトは以下のプロパティを持っています。

プロパティ	目的
PatientID	現在の症例 ID を保持します。
SiteID	現在の施設の ID を持っています。施設 ID は、データベース内でその施設を表す、内部的に与えられた値です。

プロパティ 目的	
タイプ	無作為割付のタイプを保持します。 <ul style="list-style-type: none">1 (シンプルセントラル) — 臨床試験で薬剤キットのリストを 1 つ使用します。それぞれの新しい症例には、そのリストで次の薬剤キット番号が割り当てられます。2 (セントラル階層化) — 臨床試験に薬剤キットのリストが複数あります。それぞれの新しい症例は、入力された症例データに基づいて薬剤キットリストに割り当てられます。次に、そのリストで次の薬剤キット番号が症例に割り当てられます。3 (シンプル施設) — 施設ごとに異なる薬剤キットリストがあります。それぞれの新しい症例には、症例の施設用リストで次の薬剤キット番号が割り当てられます。4 (施設による階層化) — 各施設に複数の薬剤キットリストがあります。それぞれの新しい症例は、症例の施設用のリストのセットに割り当てられます。次に、入力された症例データに基づいて、症例は施設の薬剤キットリストのなかの 1 つに割り当てられます。最後に、そのリストで次の薬剤キット番号が症例に割り当てられます。
Source	無作為割付ソースリスト名かあるいは階層化コード。
Revision	スタディバージョンの改訂番号。

GetNextKitメソッド (KitInfo)

Randomization.Source プロパティで指定された無作為割付ソースから次のシーケンス番号を得ます。シーケンスの次の番号が追加キット情報と関連付けられている場合、GetNextKit は対応するキット情報も返します。

引数

キット情報がある場合には、返されたシーケンス番号に関連付けられているキット情報。

例

以下の例はシンプルセントラル (Type 1)無作為割付計画向けの無作為割付ルールを説明しています。

```
Randomization.SiteID = Patient.GetSiteID
Randomization.Type = 1
Randomization.Source = "SimpleCentral"
Randomization.PatientID = Patient.GetID
Randomization.Revision = 0
SeqNum = Randomization.GetNextKit(KitInfo)
Result.Result= SeqNum + " / " + KitInfo
```

以下の例はセントラル階層化(Type 2)無作為割付計画向けの無作為割付ルールを説明しています。この例では、階層化は症例のビジット 1 の症例背景フォームに記入された体重

に基づいています。症例の体重に基づき、InForm ソフトウェア が CS_WT150 もしくは CS_WT275 無作為割付ソースリストから次のシーケンス番号を得ます。

```
Function GetRndSourceList()
wt = Patient.GetValue("0.Visit1.DEM.DEM.0.WEIGHT", "", 0,0,0)
IF (wt > 90 AND wt < 150) THEN
GetRndSourceList = "CS_WT150"
ELSEIF (wt > 150 AND wt < 275) THEN
GetRndSourceList = "CS_WT275"
ELSE
GetRndSourceList = ""
END IF
End Function
```

```
'set properties
Randomization.SiteID = Patient.GetSiteID
Randomization.Type = 2
Randomization.Source = GetRndSourceList
Randomization.PatientID = Patient.GetID
Randomization.Revision = 0
'randomize the patient and return result
SeqNum = Randomization.GetNextKit(KitInfo)
Result.Result= SeqNum + " / " + KitInfo
```

以下の例はシンプル施設 (Type 3)無作為割付計画向けの無作為割付ルールを説明します。この例では、InForm アプリケーションが次のシーケンス番号を得る無作為割付ソースリストが症例の施設によって決まります。

```
Function GetRndSourceList()
szSite = Patient.GetSiteName();
IF (szSite = "Mass General") THEN
GetRndSourceList = "SS_PF"
ELSEIF (szSite = "Beth Israel") THEN
GetRndSourceList = "SS_BID"
ELSE
GetRndSourceList = ""
END IF
End Function
```

```
'set properties
Randomization.SiteID = Patient.GetSiteID
Randomization.Type = 3
Randomization.Source = GetRndSourceList
Randomization.PatientID = Patient.GetID
Randomization.Revision = 0
'run the randomization and return result
SeqNum = Randomization.GetNextKit(KitInfo)
Result.Result= SeqNum + " / " + KitInfo
```

以下の例では施設による階層化 (Type 4) 無作為割付計画向けの無作為割付ルールを説明しています。この例では、InForm アプリケーションが次のシーケンス番号を得る無作為割付ソースリストが症例の施設および身長によって決まります。この例は Oracle 無作為割付ソースリスト SR_PF_HT45 と SR_PF_HT75、および Beth Israel 無作為割付ソース

リスト SR_BID_HT45 と SR_BID_HT75 を使用しています。

```
Function GetRndSourceList()
szSite = Patient.GetSiteName();
ht = Patient.GetValue("0.Visit1.DEM.DEM.0.HEIGHT", "", 0,0,0)
```

```
SELECT CASE szSite
CASE "Mass General"
IF (ht > 30 AND ht < 45) THEN
GetRndSourceList = "SR_PF_HT45"
ELSEIF (ht > 45 AND < 75) THEN
GetRndSourceList = "SR_PF_HT75"
ELSE
GetRndSourceList = ""
END IF
```

```
CASE "Beth Israel"
IF (ht > 30 AND ht < 45) THEN
GetRndSourceList = "SR_BID_HT45"
ELSEIF (ht > 45 AND < 75) THEN
GetRndSourceList = "SR_BID_HT75"
ELSE
GetRndSourceList = ""
END IF
```

```
CASE ELSE
GetRndSourceList = ""
END SELECT
End Function
```

```
'setup randomization object properties
Randomization.SiteID = Patient.GetSiteID
Randomization.Type = 4
Randomization.Source = GetRndSourceList
Randomization.PatientID = Patient.GetID
Randomization.Revision = 0
'randomize the patient and return result
SeqNum = Randomization.GetNextKit(KitInfo)
Result.Result= SeqNum + " / " + KitInfo
```

GetNextNumberメソッド (KitInfo)

Randomization.Source プロパティで指定された無作為割付ソースから次のシーケンス番号を得ます。このメソッドは GetNextKitNumber メソッドとは異なります。(シーケンス番号に関連付けられた追加のキット情報は返されません。

引数

キット情報がある場合には、返されたシーケンス番号に関連付けられているキット情報。

ルールおよび計算のオブジェクトとメソッド

CRF ルールのスクリプト（CRF のアイテムに入力されたデータにエディットチェックを行うスクリプト）、および計算ルールのスクリプト（1 つ以上の関連したアイテムの値を基に、CRF のデータアイテムの値を計算するスクリプト）は、以下のオブジェクトを使用することができます。

<注> CRF ルールおよび計算定義は、**Central Designer** アプリケーションの ルールウィザードを使用するか、または **MedML** と **MedML** インストーラユーティリティを使用して作成することができます。

Patientオブジェクト

症例に関するフォーム値を得ることができます。Patient オブジェクトには、以下のプロパティがあります（全てが Get と Put プロパティ）。

プロパティ	タイプ	目的
Result	VARIANT	計算結果を保持します。
Message	BSTR	実行計画のスクリプトで使用される Global オブジェクトの ContextString プロパティに文字列を渡します。このプロパティは CRF ルールのみに適用されます。
RulePassed	BOOL	エディットチェックを通ったか、失敗したかを指定します。もし RulePassed が true (1 に等しい)の場合、エディットチェックは成功であり;false (0 に等しい)の場合、エディットチェックは失敗です。
BaseValue	NUMERIC	変換される値を含んでいます。

Patient メソッド

CRF ルールおよび計算のスクリプトで、多くの同じメソッドを使用することができます。特記されている場合を除き、Patient メソッドは CRF ルールと計算の両方で利用可能です。

Method	目的
AddNamedValue	GetNamedValue メソッドでアクセスできる引数のリストに 名/値のペアを追加します。
ClearValues	指定されたオブジェクトのプロパティをクリアします。
FormHasState	CRF が指定した状態であるかどうかを表します。 FormHasState はブール値を返します。
FormHasStateEx	CRF が指定された状態であるかどうかを表すブール値を返します。

Method	目的
FormHasStateRF	繰り返し CRF のインスタンスが指定された状態であるかどうかを表すブール値を返します。
GetArgument	指定されたルール引数の値を返します。
GetArgumentCount	ルールコンテキストで使用された引数の数を返します。
GetAssociationCount	ビジットにある特定の CRF インスタンスに関連付けられている繰り返しフォームインスタンス数を返します。
GetAssociationValue	GetAssociationCount メソッドの最新の呼び出しで返されたカウントに到達するまで、CRF の関連を繰り返します。GetAssociationCount の値に相当する関連に到達すると、GetAssociationValue は指定された関連フォームインスタンスに対してのインデックスと RefName を含む配列を返します。
GetCurPath	ルールが実行されているコンテキストのパスを返します。
GetCurrentFormIndex	現在の繰り返しフォームのインスタンスのインデックス番号を返します。
GetCurrentISRowNum	現在の行のインデックス (新しい行の場合は 0) 番号を返します。
GetCurrentVisitIndex	現在のビジットのインデックス番号を返します。
GetDefaultValue	GetValue と同じ。デフォルトパラメータを持ちます。
GetEnteredValue	指定された CRF コンポーネントに入力されたデータの値を返します。
GetEnteredValueRF	繰り返し CRF の特定インスタンスにおいて、指定された CRF コンポーネントに入力されたデータの値を返します。
GetFormCount	指定されたビジットに対して作成された繰り返し CRF インスタンス数を返します。
GetID	現在の症例の内部データベース ID を返します。
GetItemsetCount	アイテムセットに現在入力されている行の数を返します。 注： これは、順に採番したり、行を識別する手段としては使用しないでください。
GetItemsetRowID	アイテムセット行を一意に識別する行 ID を返します。
GetItemsetCountRF	繰り返し CRF の特定インスタンスにおいて、アイテムセットに現在入力されている行数を返します。
GetPatientNumber	現在の症例に割り当てられている番号を返します。
GetRefName	ルールが実行されるコンテキストの指定されたオブジェクトタイプ の RefName を文字列で返します。

Method	目的
GetRefPath	ルールが実行されるコンテキストの指定されたオブジェクトタイプの RefName パスを文字列で返します。
GetScreeningNumber	現在の症例のスクリーニング番号を文字列で返します。
GetServerTime	現在のサーバの日時値を GMT として文字列で返します。
GetSiteID	現在の症例の施設に関する内部 ID を返します。
GetSiteMnemonic	現在の症例の施設の施設略名を文字列で返します。
GetSiteName	現在の症例の施設名を返します。
GetSiteTime	現在の施設のタイムゾーンでサーバの日時値を返します。
GetTrialName	現在の症例が参加しているスタディの名前を返します。
GetValue	指定された CRF コンポーネントに入力されたデータの正規化された値を返します。
GetValueRF	繰り返し CRF の特定インスタンスにおいて、指定された CRF コンポーネントに入力されたデータの正規化された値を返します。
GetVisitCount	VisitRefName で指定されたビジットのビジット数を返します。
IsCBSelected	チェックボックスコントロールのシンプルコントロールが選択されているかどうかを確認して、チェックボックスの状態を示すブール値を返します。
IsCBSelectedRF	繰り返し CRF の特定のインスタンスにおいて、チェックボックスコントロールのシンプルコントロールが選択されているかどうかを確認して、チェックボックスの状態を示すブール値を返します。
IsItemSetDeleted	エントリ追加アイテムセットのアイテムが削除されているかどうかを示します。
IsItemSetDeletedRF	繰り返し CRF の特定インスタンスにおいて、エントリ追加アイテムセットのアイテムが削除されているかどうかを示します。
IsValidItemPath	指定されたパスの存在をデータベースでチェックし、RefName パスが有効であるかどうかを確認します。
IsValidItemPathRF	繰り返し CRF の特定インスタンスにおいて、指定されたパスの存在をデータベースでチェックし、RefName パスが有効であるかどうかを確認します。
ItemHasBeenNormalizedEx	データが正規化されたかどうか確認することによって、参照されるデータアイテムに関連する単位が選択されているかを示します。

Method	目的
ItemHasBeenNormalizedRF	繰り返し CRF の特定インスタンスにおいて、データが正規化されたかどうか確認することによって、参照されるデータアイテムに関連する単位が選択されているかを示します。
LoadCRB	症例 CRB をメモリにロードします。
OutputDebug	テストスクリプトの実行時に、指定された変数の値を表示します。
RemoveEmptyDynamicVisit	ビジットの CRF のいずれにデータが入力されていない場合、症例にスケジューリングされているダイナミックビジットを削除します。
RemoveNotStartedDynamicForm	CRF が開始されていない場合、症例の CRB からダイナミックフォームを削除します。
SetDynamicVisitStart	ダイナミックビジットに症例をスケジューリングします。
SetEnteredValue	指定された CRF コンポーネントに入力されたデータの値を使用します。
SetEnteredValueRF	繰り返し CRF の特定インスタンスにおいて、指定された CRF コンポーネントに入力されたデータの値を使用します。
SetNewValue	Oracle 内部でのみ使用されます。
SetNewValueRF	Oracle 内部でのみ使用されます。
SetValue	指定された CRF コンポーネントに入力されたデータの正規化された値を使用します。
SetValueRF	繰り返し CRF の特定インスタンスにおいて、指定された CRF コンポーネントに入力されたデータの正規化された値を使用します。
ShowDynamicForm	他の CRF での計算結果を基にして、ビジットにダイナミック CRF を挿入します。

Resultオブジェクト

CRF ルールで生成されたエディットチェックの結果や、計算によって生成されるデータアイテム値を、取得・設定できるようにします。Result オブジェクトのプロパティは Patient オブジェクトのプロパティと同一です。

プロパティ	タイプ	目的
Result	VARIANT	計算結果を保持します。
Message	BSTR	実行計画のスクリプトで使用する Global オブジェクトの ContextString プロパティに文字列を渡します。このプロパティは CRF ルールのみに適用されます。

プロパティ	タイプ	目的
RulePassed	BOOL	エディットチェックを通ったか、失敗したかを指定します。もし RulePassed が true (1 に等しい) の場合、エディットチェックは成功であり;false (0 に等しい) の場合、エディットチェックは失敗です。
BaseValue	NUMERIC	変換される値を含んでいます。

Result メソッド

Result オブジェクトで利用可能なメソッドは、Patient オブジェクトで利用できるメソッドと同じです。

メソッド	目的
AddNamedValue	GetNamedValue メソッドでアクセスできる引数のリストに名/値のペアを追加します。
ClearValues	指定されたオブジェクトのプロパティをクリアします。
FormHasState	CRF が指定した状態であるかどうかを表します。 FormHasState はブール値を返します。
FormHasStateEx	CRF が指定された状態であるかどうかを表すブール値を返します。
FormHasStateRF	繰り返し CRF のインスタンスが指定された状態であるかどうかを表すブール値を返します。
GetArgument	指定されたルール引数の値を返します。
GetArgumentCount	ルールコンテキストで使用された引数の数を返します。
GetAssociationCount	ビジットにある特定の CRF インスタンスに関連付けられている繰り返しフォームインスタンス数を返します。
GetAssociationValue	GetAssociationCount メソッドの最新の呼び出しで返されたカウントに到達するまで、CRF の関連を繰り返します。 GetAssociationCount の値に相当する関連に到達すると、GetAssociationValue は指定された関連フォームインスタンスに対してのインデックスと RefName を含む配列を返します。
GetCurPath	ルールが実行されているコンテキストのパスを返します。
GetCurrentFormIndex	現在の繰り返しフォームのインスタンスのインデックス番号を返します。
GetCurrentISRowNum	現在の行のインデックス (新しい行の場合は 0) 番号を返します。
GetCurrentVisitIndex	現在のビジットのインデックス番号を返します。
GetDefaultValue	GetValue と同じ。デフォルトパラメータを持ちます。

メソッド	目的
GetEnteredValue	指定された CRF コンポーネントに入力されたデータの値を返します。
GetEnteredValueRF	繰り返し CRF の特定インスタンスにおいて、指定された CRF コンポーネントに入力されたデータの値を返します。
GetFormCount	指定されたビジットに対して作成された繰り返し CRF インスタンス数を返します。
GetID	現在の症例の内部データベース ID を返します。
GetItemsetCount	アイテムセットに現在入力されている行の数を返します。 注： これは、順に採番したり、行を識別する手段としては使用しないでください。
GetItemsetRowID(REFN AMEPath, VisitIndex)	アイテムセット行を一意に識別する行 ID を返します。
GetItemsetCountRF	繰り返し CRF の特定インスタンスにおいて、アイテムセットに現在入力されている行数を返します。
GetPatientNumber	現在の症例に割り当てられている番号を返します。
GetRefName	ルールが実行されるコンテキストの指定されたオブジェクトタイプ の RefName を文字列で返します。
GetRefPath	ルールが実行されるコンテキストの指定されたオブジェクトタイプの RefName パスを文字列で返します。
GetScreeningNumber	現在の症例のスクリーニング番号を文字列で返します。
GetServerTime	現在のサーバの日時値を GMT として文字列で返します。
GetSiteID	現在の症例の施設に関する内部 ID を返します。
GetSiteMnemonic	現在の症例の施設の施設略名を文字列で返します。
GetSiteName	現在の症例の施設名を返します。
GetSiteTime	現在の施設のタイムゾーンでサーバの日時値を返します。
GetTrialName	現在の症例が参加しているスタディの名前を返します。
GetValue	指定された CRF コンポーネントに入力されたデータの正規化された値を返します。
GetValueRF	繰り返し CRF の特定インスタンスにおいて、指定された CRF コンポーネントに入力されたデータの正規化された値を返します。
GetVisitCount	VisitRefName で指定されたビジットのビジット数を返します。
IsCBSelected	チェックボックスコントロールのシンプルコントロールが選択されているかどうかを確認して、チェックボックスの状態を示すブール値を返します。

メソッド	目的
IsCBSelectedRF	繰り返し CRF の特定のインスタンスにおいて、チェックボックスコントロールのシンプルコントロールが選択されているかどうかを確認して、チェックボックスの状態を示すブール値を返します。
IsItemSetDeleted	エントリ追加アイテムセットのアイテムが削除されているかどうかを示します。
IsItemSetDeletedRF	繰り返し CRF の特定インスタンスにおいて、エントリ追加アイテムセットのアイテムが削除されているかどうかを示します。
IsValidItemPath	指定されたパスの存在をデータベースでチェックし、RefName パスが有効であるかどうかを確認します。
IsValidItemPathRF	繰り返し CRF の特定インスタンスにおいて、指定されたパスの存在をデータベースでチェックし、RefName パスが有効であるかどうかを確認します。
ItemHasBeenNormalizedEx	データが正規化されたかどうか確認することによって、参照されるデータアイテムに関連する単位が選択されているかを示します。
ItemHasBeenNormalizedRF	繰り返し CRF の特定インスタンスにおいて、データが正規化されたかどうか確認することによって、参照されるデータアイテムに関連する単位が選択されているかを示します。
LoadCRB	症例 CRB をメモリにロードします。
OutputDebug	テストスクリプトの実行時に、指定された変数の値を表示します。
RemoveEmptyDynamicVisit	ビジットの CRF のいずれにデータが入力されていない場合、症例にスケジューリングされているダイナミックビジットを削除します。
RemoveNotStartedDynamicForm	CRF が開始されていない場合、症例の CRB からダイナミックフォームを削除します。
SetDynamicVisitStart	ダイナミックビジットに症例をスケジューリングします。
SetEnteredValue	指定された CRF コンポーネントに入力されたデータの値を使用します。
SetEnteredValueRF	繰り返し CRF の特定インスタンスにおいて、指定された CRF コンポーネントに入力されたデータの値を使用します。
SetNewValue	Oracle 内部でのみ使用されます。
SetNewValueRF	Oracle 内部でのみ使用されます。
SetValue	指定された CRF コンポーネントに入力されたデータの正規化された値を使用します。

メソッド	目的
SetValueRF	繰り返し CRF の特定インスタンスにおいて、指定された CRF コンポーネントに入力されたデータの正規化された値を使用します。
ShowDynamicForm	他の CRF での計算結果を基にして、ビジットにダイナミック CRF を挿入します。

AddNamedValue"Name","Value"

GetNamedValue メソッドでアクセスできる引数のリストに名/値のペアを追加します。GetNamedValue は実行計画の定義で使用される Global オブジェクトのメソッドです。また、このメソッドを使用して、ダイナミックなクエリテキストを生成し、ルールに関連付けられている Event 定義で指定されるデフォルトテキストよりも優先させることが可能です。

以下に示す、事前に定義された名/値のペアはダイナミックなクエリテキスト生成に使用できます。

- **QUERYTEXT** — 値にはクエリのテキストを指定します。
- **QUERYSTATE** — 値には“仮登録”または“オープン”を使用して、ダイナミックに生成されたクエリの状態を指定します。これらの値では、大文字と小文字が区別されません。

引数

名前

アイテム名を表す文字列。これは単なる文字列で、アイテムの RefName ではありません。

値

追加する値。

例 1

```
'Check if ae is serious
'-----
AESer = "0.CommonCRF.AVR.AE.AE.SAE"
AESerVal_g=Patient.GetValue(AESer,"",0,0,0)
if AESerVal_g = "Y" then
sitenm_g=Patient.GetSiteName()
ptinit_g=cstr(Patient.GetValue("0.screen.screen.screen.0.patientinitials","",0,0,0))
aedate_g=Patient.GetValue("0.CommonCRF.AVR.AE.AE.STARTDT","",0,0,0)
aceevent_g=Patient.GetValue("0.CommonCRF.AVR.AE.AE.AEDESC","",0,0,0)
patient.AddNamedValue "sitenm", sitenm_g
patient.AddNamedValue "ptinit", ptinit_g
patient.AddNamedValue "aedate", aedate_g
patient.AddNamedValue "aceevent", aceevent_g
patient.AddNamedValue "aeserval", aeserval_g
```


End If

例 2

```
Result.RulePassed = 0
```

Patient.AddNamedValue "QUERYTEXT", "手首囲はありますが、単位はありません；確かめてください。

```
if (Patient.IsItemSetDeleted(Patient.GetCurPath(),0,0)) then
```

```
    Patient.AddNamedValue "QUERYSTATE", "仮登録"
```

```
else
```

```
    Patient.AddNamedValue "QUERYSTATE", "オープン"
```

```
End if
```

例 3

以下のスクリプトは、血圧測定の収縮血圧が拡張期血圧より低いかどうかチェックします。ルールが失敗した場合、ルールは AddNamedValue メソッドを呼び出してクエリを発行し、このテキストによりルールの Event 定義に指定されているデフォルトのクエリテキストが上書きされます。

```
Measure1 = Patient.GetValue(Patient.GetArgument("SysItem"), "", 0,0,0)
```

```
Measure2 = Patient.GetValue(Patient.GetArgument("DiasItem"), "", 0,0,0)
```

```
if (Measure1<Measure2) Then
```

```
    Result.RulePassed=0
```

Patient.AddNamedValue "QUERYTEXT", "収縮血圧は拡張期血圧よりも高い必要があります。

```
    Patient.AddNamedValue "QUERYSTATE", "オープン"
```

```
Else
```

```
    Result.RulePassed=1
```

```
End If
```

ClearValues()

指定したオブジェクトのプロパティとコントロール名/値ペアをクリアします。

FormHasState(Visit,Form,State)

CRF が指定した状態であるかどうかを表します。FormHasState はブール値を返します。FormHasState は、繰り返しでない通常のビジットに使用してください。予定外の繰り返しビジットの特定のインスタンスを対象とするには、FormHasStateEx を使用してください。

引数

ビジット

ビジットの RefName。

注： これは完全な REFNAMEPath ではなく、ビジットだけの REFNAME です。

フォーム

CRF の RefName。

注： これは完全な **REFNAMEPath** ではなく、**CRF** だけの **REFNAME** です。

ステータス

チェックする状態を指定する文字列。以下のものが有効です。

- **Frozen** — CRF がフリーズされていることを示します。
- **Hascomment** — CRF に CRF レベルでコメントがあることを示します。
- **Hasdata** — CRF にデータがあることを示します。
- **Locked** — CRF がロックされていることを示します。
- **Missing** — CRF のデータが欠落していることを示します。
- **Queries** — CRF にクエリがあることを示します。
- **SDVComplete**—CRF は SDV 完了としてマークされています。 .
- **SDVPartial**—CRF は未完了としてマークされています。
- **SDVReady** — CRF が SDV 実施可であることを示します。
- **Signed** — CRF が署名済みであることを示します。
- **Skipped** — CRF が CRF レベルのコメント付きで完了していないことを示します。
- **Started** — CRF が症例データ、コメント、またはクエリで開始されたことを示します。

例

たとえば、AE フォームが併用薬(ConMed)を採用したことを示す場合、併用薬(ConMed) フォームにデータがあることを確認します。If Patient.FormHasState("CommonVisit", "ConMed", "missing") then
Result.RulePassed = 0

Else
Result.RulePassed = 1

FormHasStateEx(Visit,VisitIndex,Form,State)

CRF が指定した状態であるかどうかを表します。FormHasStateEx はブール値を返します。FormHasState Ex には VisitIndex が含まれているため、繰り返しビジットの特定のインスタンスをターゲットにできることを除き、FormHasStateEx は FormHasState とまったく同じです。

引数

ビジット

ビジットの RefName。

注： これは完全な **REFNAMEPath** ではなく、ビジットだけの **REFNAME** です。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

フォーム

CRF の RefName。これは完全な REFNAMEPath ではなく、CRF だけの RefName であることに注意してください。

ステータス

チェックする状態を指定する文字列。以下のものが有効です。

- **Frozen** — CRF がフリーズされていることを示します。
- **Hascomment** — CRF に CRF レベルでコメントがあることを示します。
- **Hasdata** — CRF にデータがあることを示します。
- **Locked** — CRF がロックされていることを示します。
- **Missing** — CRF にデータが欠落していることを示します。
- **Queries** — CRF にクエリがあることを示します。
- **SDVComplete**—CRF は SDV 完了としてマークされています。
- **SDVPartial**—CRF は未完了としてマークされています。
- **SDVReady** — CRF が SV 実施可であることを示します。
- **Signed** — CRF が署名済みであることを示します。
- **Skipped** — CRF が CRF レベルコメントを付けて完了していないことを示します。
- **Started** — CRF が症例データ、コメント、またはクエリで開始されたことを示します。

例

たとえば、予定外ビジットの 2 番目のインスタンスにおける AE CRF が併用薬 (ConMed) を採用したことを示す場合、同じインスタンス内の併用薬 (ConMed) CRF にデータがあるか確認します。

```
If Patient.FormHasStateEx("UnschVisit", "2", "ConMed", "missing") then
  Result.RulePassed = 0
```

```
Else
  Result.RulePassed = 1
```

FormHasStateRF(Visit,VisitIndex,Form,FormIndex,State)

繰り返し CRF について、FormHasStateRF メソッドは、CRF が指定した状態にあるかどうかを示します。FormHasStateRF はブール値を返します。FormHasStateRF は、FormIndex を除いて、FormHasStateEx とまったく同じです。

- 繰り返し CRF の特定のインスタンスを対象とできるように、FormIndex 引数を含んでいます。
- 削除された状態かどうかをチェックすることができます。

引数

ビジット

ビジットの RefName。これは完全な REFNAMEPath ではなく、ビジットの RefName のみであることを注意してください。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

フォーム

フォームの RefName。これは完全な REFNAMEPath ではなく、フォームの RefName のみであることを注意してください。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

ステータス

チェックする状態を指定する文字列。以下のものが有効です。

- **Frozen** — CRF がフリーズされていることを示します。
- **Hascomment** — CRF に CRF レベルでコメントがあることを示します。
- **Hasdata** — CRF にデータがあることを示します。
- **Locked** — CRF がロックされていることを示します。
- **Missing** — CRF にデータが欠落していることを示します。

- **Queries** — CRF にクエリがあることを示します。
- **SDVComplete**—CRF は SDV 完了としてマークされています。
- **SDVPartial**—CRF は未完了としてマークされています。
- **SDVReady** — CRF が SV 実施可であることを示します。
- **Signed** — CRF が署名済みであることを示します。
- **Skipped** — CRF が CRF レベルコメントを付けて完了していないことを示します。
- **Started** — CRF が症例データ、コメント、またはクエリで開始されたことを示します。
- **Deleted (CRF のインスタンスの場合)** — CRF インスタンスが削除されたことを示します。

例

次の例は、有害事象の説明内容が txtAEDiag コントロールに既に入力されているかどうかをテストしています。FormHasStateRF メソッドを使用して、テストするインスタンスが削除されているかどうかを判断し、削除されたインスタンスをテストから除外することができます。

```
patient.RulePassed = 1
Patient.AddNamedValue "QUERYTEXT", "This AE already exists"
path="0.CommonCRF.AE.sctAE.0.itmAEDiag.txtAEDiag"
isValid=patient.IsValidItemPathRF (path,0,0)
strVal=patient.GetValueRF(path,"",0,0,0,0)
intcurFormCount=patient.GetCurrentFormIndex()
intFormCount=patient.GetFormCount("CommonCRF",1,"AE")

for count = 1 to intFormCount
intIsDel=Patient.FormHasStateRF("CommonCRF",1,"AE",count,"deleted")
if (cint(count) <> cint(intcurFormCount)) and intIsDel=0 then
strVal2=patient.GetValueRF("0.CommonCRF.AE.sctAE.0.itmAEDiag.txtAEDiag","", 0, 0, 0,
count)
If Ucase(strVal) = Ucase(strVal2) then
patient.RulePassed = 0
exit for
End If
End If
Next
```

GetArgument

指定されたルール引数の値を返します。引数は RULEARG エレメントで定義され、ATTACHRULESET 定義の中のルールとアイテムに関連付けられています。

例

この例では、ルールのために定義され、テストされているアイテムに関連付けられた引数の名前が「Item1」です。

```
pulse = Patient.GetValue(Patient.GetArgument("Item1"), "", 0, 0, 0)
```

GetArgumentCount()

ルールコンテキストで使用された引数の数を返します。

GetAssociationCount(RefNamePath,VisitIndex,FormIndex)

ビジットにある特定の CRF インスタンスに関連付けられている繰り返し CRF インスタンス数を返します。

引数

RefNamePath

取得する CRF を特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含んだ .xml ファイルで指定された RefName です。RefNamePath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。以下のフォーマットには RefNamePath argument 引数があります。 (0.Visit.Form)

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。CRF の最初のインスタンスでは、CRF インデックスは 1 から始まります。

例

以下の例は、GetAssociationCount が返す値を使用して、現在のフォームに関連付けられている繰り返し CRF インスタンスのグループ内の欠測値をテストする回数を設定します。GetAssociationCount は、GetAssociationValue メソッドと連携して機能します。

```
patient.RulePassed = 1
Patient.AddNamedValue "QUERYTEXT", "At least one of the Associated forms is missing data"
nAssociations = Patient.GetAssociationCount(Patient.GetCurPath(), Patient.GetCurrentVisitIndex(),
Patient.GetCurrentFormIndex())

for count = 1 to nAssociations
  Assoc = Patient.GetAssociationValue(count)
  AssocVisit = Assoc(0)
  AssocForm = Assoc(1)
```

```

AssocVisitIndex = Assoc(2)
AssocFormIndex = Assoc(3)
'Do something with the association.例：see if data missing
if(Patient.FormHasStateRF(AssocVisit, AssocVisitIndex , AssocForm, AssocFormIndex, "missing"))
then
patient.RulePassed = 0
end if
Next

```

GetAssociationValue(nAssociations)

GetAssociationCount メソッドと連動して、特定の関連 CRF インスタンスに関する情報を返します。GetAssociationValue は、GetAssociationCount メソッドの最新の呼び出しで返されたカウントに到達するまで、CRF の関連を繰り返します。GetAssociationCount 値が示す関連に達すると、GetAssociationValue は指定された関連フォームインスタンスに関する以下の値を持つ配列を返します。

- ビジットの RefName
- CRF の RefName
- VisitIndex
- FormIndex

引数

nAssociations

繰り返し処理する関連 CRF のインスタンス数。1 から始めて指定した数が入ります。

例

この例では GetAssociationValue メソッドを使用して、インスタンスが符号付きであるかどうかチェックしています。

```

nAssociations = Patient.GetAssociationCount(Patient.GetCurPath(),
Patient.GetCurrentVisitIndex(), Patient.GetCurrentFormIndex())

```

```

for count = 1 to nAssociations
  Assoc = Patient.GetAssociationValue(count)
  AssocVisit = Assoc(0)
  AssocForm = Assoc(1)
  AssocVisitIndex = Assoc(2)
  AssocFormIndex = Assoc(3)
  ' Do something with the association.例：see if signed
  □□□□ if(Patient.FormHasStateRF(AssocVisit, AssocVisitIndex , AssocForm,
AssocFormIndex, "signed") then
  ' associated to signed form
  □□□□ endif
Next

```


GetCurPath()

ルールが実行されているコンテキストの RefName パスを返します。

例

```
val = Patient.GetValue(Patient.GetCurPath(), "", 0,0,0)
```

GetCurrentFormIndex()

現在の繰り返し CRF のインスタンスのインデックス番号または 0 を返します。このメソッドを使用して、変更されている繰り返し CRF のインスタンスを特定したり、新しいインスタンスが Submit に追加されているかどうかを特定することができます。

繰り返し CRF インスタンスの最初のサブミットでは、GetCurrentFormIndex は 0 を返します。その後は、インスタンスのインデックスを返します。

例

この例では、GetCurrentFormIndex は、繰り返し CRF の現在のインスタンスのフォームインデックスを GetAssociationCount メソッドに渡します。

```
patient.RulePassed = 1
Patient.AddNamedValue "QUERYTEXT", "At least one of the Associated forms is missing data"
nAssociations = Patient.GetAssociationCount(Patient.GetCurPath(), Patient.GetCurrentVisitIndex(),
Patient.GetCurrentFormIndex())

for count = 1 to nAssociations
  Assoc = Patient.GetAssociationValue(count)
  AssocVisit = Assoc(0)
  AssocForm = Assoc(1)
  AssocVisitIndex = Assoc(2)
  AssocFormIndex = Assoc(3)
  'Do something with the association.例： see if data is missing
  if(Patient.FormHasStateRF(AssocVisit, AssocVisitIndex , AssocForm, AssocFormIndex, "missing"))
  then
    patient.RulePassed = 0
  end if
Next
```

GetCurrentISRowNum()

現在のアイテムセット行のインデックス番号を返します。メソッドは現在の行が新しい行である場合、0 を返します。このメソッドを使用して、itemset のどの行が変更されているか、または新しい行が Submit で追加されているかどうか特定することができます。

例

このルールスクリプトは有害事象 (Adverse Experience) CRF に付加されて、CRF に入力される有害事象 (Adverse Experience) に関する情報を取得します。次に、AddNamedValue メソッドを使用する実行計画に変数が渡されます。

```
ISrownum = Cint(patient.GetCurrentISRowNum())
```

```

AdverseExp =
Cstr(patient.GetValue("0.CommonCRF.AE.AE.AE.AEDESC.AEDESCTEXT"," ",0,0,0))
sitemnemonic = Cstr(patient.GetSiteMnemonic())
screeningnumber = Cstr(patient.GetScreeningNumber())
patientnumber = Cstr(patient.GetPatientNumber())
servertime = Cdate(patient.GetServerTime())
sitetime = Cdate(patient.GetSiteTime())

```

```

patient.AddNamedValue "passISrownum", ISrownum
patient.AddNamedValue "passAE", AdverseExp
patient.AddNamedValue "passitemnemonic", sitemnemonic
patient.AddNamedValue "passscreeningnumber", screeningnumber
patient.AddNamedValue "passpatientnumber", patientnumber
patient.AddNamedValue "passservertime", servertime
patient.AddNamedValue "passsitetime", sitetime

```

上記のルールのためのイベントが実行計画を作動させます。実行計画は、itemset の行番号をチェックして、レポートされる有害事象が新しいものであること(値=0)を確認します。有害事象が新しい場合、いくつか異なる部分の情報を添えて電子メールが送信されます。

```

getISrownum = Cint(global.GetNamedValue("passISrownum"))

If (getISrownum = 0) then
  getAE = cstr(global.GetNamedValue("passAE"))
  getsitemnemonic = cstr(global.GetNamedValue("passitemnemonic"))
  getscreeningnumber = cstr(global.GetNamedValue("passscreeningnumber"))
  getpatientnumber = cstr(global.GetNamedValue("passpatientnumber"))
  getservertime = cdate(global.GetNamedValue("passservertime"))
  getsitetime = cdate(global.GetNamedValue("passsitetime"))
  emailsub = "Adverse Experience Notification:" & getAE
  emailtxt = "An Adverse Experience has occurred.
  Adverse Experience = " & getAE& ",
  Site = " & getsitemnemonic& ",
  Site Time = " & getsitetime& ",
  Server Time = " & getservertime& ",
  Screening Number = " & getscreeningnumber& ",
  Subject Number = " & getpatientnumber
  global.EMailToInternetUser "user@Oracle.com", emailsub , emailtxt
end if

```

GetCurrentVisitIndex()

現在のビジットのインデックス番号を返します。予定外のビジットシリーズのどのビジットが変更されているか、または新しいビジットが Submit で追加されているかどうか決定するには、このメソッドを使用してください。

例

このスクリプトは、現在の予定外ビジットのビジット日付が前回のビジット日付より後かどうかをチェックして、現在のビジット日付よりも前ならば、クエリを生成します。

```
VisitIndex = Cint(patient.GetCurrentVisitIndex())
```

```
valnewvisit = Cdate(Patient.GetValue( Patient.GetCurPath() & "." &
    Patient.GetArgument("Controlname"), "", 0, 0, VisitIndex))
valoldvisit = Cdate(Patient.GetValue( Patient.GetCurPath() & "." &
    Patient.GetArgument("Controlname"), "", 0, 0, VisitIndex- 1))

difference = DateDiff("n", valnewvisit, valoldvisit)

if (VisitIndex = 1) then
    result.RulePassed=1
else
    if (difference > 0) then
        result.RulePassed=0
        patient.AddNamedValue "QUERYTEXT", "This date is earlier than the date
        of the previous visit."
        patient.AddNamedValue "QUERYSTATE", "Open"
    Else
        result.RulePassed=1
    end if
end if
```

GetDefaultValue(RefNamePath)

GetValue メソッドと同じですが、デフォルトのパラメータを持ちます。GetDefaultValue は、AttributeName、Index、ItemIndex、VisitIndex に対して、以下の値をデフォルトとします。

- **AttributeName="VALUE"** — メソッドは CRF コンポーネントの値を返します。
- **Index="0"** — CRF コンポーネントは多重選択コンポーネントではありません。
- **ItemsetIndex="0"** — 現在のアイテムセット行、またはアイテムがアイテムセットにありません。
- **VisitIndex="0"** — 現在のビジット。

アイテムの RefNamePath を指定するだけで、メソッドは指定された CRF コンポーネントに入力されたデータの正規化した値を返します。アイテムが正規化されていない場合、メソッドは入力された値を返します。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

例

```
Value = Patient.GetDefaultValue(Patient.GetCurPath())
```

```
If (Value>MinVal and Value<MaxVal) Then
```

```
    Result.RulePassed = 1
```

```
Else
```

```
    Result.RulePassed = 0
```

```
End If
```

GetEnteredValue(REFNAMEPath,AttributeName,Index,ItemSetIndex,VisitIndex)

指定された CRF コンポーネントに入力されたデータの値を返します。GetEnteredValue は、指定されたフォームコンポーネントに入力されたデータを正規化して返す GetValue メソッドに非常に似ています。これらのメソッドの相違は、ユニットを使用して定義されるフォームコンポーネントに適用します。

MedML インストラユーティリティでは、単位として指定する CRF コンポーネントを定義することができます。データをオンライン上で 1 つの単位として表示/データベースへ保存するために、他の単位を参照して単位 CRF コンポーネントを定義することができます。例えば、CRC がオンス単位かポンド単位のどちらかで体重を入力できるように CRF を設計し、データベース上ではオンスで保存されるように指定することも可能です。変換後にデータベースに保存したデータのユニットをベース単位と呼び、ベース単位に変換されたデータ値を正規化された値と呼びます。

データアイテムの入力値を確認する場合は、GetEnteredValue を使用してください。データアイテムの正規化された値を入手する際には、GetValue を使用してください。

注：繰り返し CRF のデータ値を確認する、または繰り返しでない CRF の値を参照するために繰り返しフォームにルールを付加するには、GetValueRF または GetEnteredValueRF を使用します。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

```
(0.Visit.Form.Section.Itemset.Item[.control[.control...]])
```

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを

取得するアイテムセットが含まれている場合があります。

- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

AttributeName

二重引用符(“)で指定された戻り値のタイプを指定するストリング。有効な AttributeNames を以下に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。空のダブルクォーテーション(“)で示すことができます。
- **DATESTRING** — DateTime コントロールの日付の部分のみ返します。
- **TIMESTRING** — DateTime コントロールの時間の部分のみ返します。
- **DATETIMESTRING** — DateTime コントロールの日付と時刻の両方の部分を返します。
- **SELECTED** — **INDEX** 属性と一緒に使用して、指定したインデックスが選択されているかどうかを示します。
- **COUNT** — コントロールで選択されるアイテムの数を指定します。例えば、コントロールが多重選択リストボックスである場合、**COUNT** は選択済みリストアイテムの数を返します。
- **UNIT** — 選択されている場合、単位のシンボルテキストを返します。

<注> 単位の **SYMBOL** 属性は変換できます。返される値は、現在の症例施設のロケールの値になります。したがって、返されたシンボルテキストをチェックするルールのロジックは、現在の症例施設のロケールを考慮する必要があります。

- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールのエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32
- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるか

どうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使⽤します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なエレメントを示すビットマスクを含む整数を返します。

- 年 — 1
- 月 — 2
- 日 — 4
- 時間 — 8
- 分 — 16
- 秒 — 32

注：日付コンポーネントの **UNKNOWNDATEMAP** 値にビットを設定した場合、対応するビットは **VALIDDATEMAP** 値に設定されません。

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

例 1: テキストコンポーネントの値をチェックする

次の例は、Vitals CRF の VS セクションにある **TEMPITEM** コンポーネントの値をチェックします。このコンポーネントでサブミットされた値が、指定された最小値と最大値の範囲内である場合、ルールの条件を満たしています。そうでなければ、ルールの条件を満たしていません。

```
Control = "0.Visit2.VITALS.VS.0.TEMPITEM"
MinVal = 95
MaxVal = 105

'-----
Value = Patient.GetEnteredValue(Control, "", 0,0,0)

If (Value>MinVal and Value<MaxVal) Then
    Result.RulePassed = 1
Else
    Result.RulePassed = 0
```


End If

例 2：コンパウンドコントロールにおけるデータの指定

この例は、グループボックスの中のテキストボックスを作ったコンパウンドコントロール内にあるデータをどう記述するかを示します。このスクリプトでは、BPREADINGITEM アイテムの BPREADINGGROUP グループにネストされている **SYSTEXT** コンポーネントと **DIASTEXT** コンポーネントの値をチェックします。

```
Control1 = "0.Visit2.VITALS.VS.0.BPREADINGITEM.BPREADINGGROUP.SYSTEXT"
```

```
Control2 = "0.Visit2.VITALS.VS.0.BPREADINGITEM.BPREADINGGROUP.DIASTEXT"
```

```
'-----
Value1 = Patient.GetEnteredValue(Control1, "", 0,0,0)
Value2 = Patient.GetEnteredValue(Control2, "", 0,0,0)
```

例 3：どのラジオボタンが選択されているかの特定

この例は、コンパウンドコントロールでどのラジオボタンが選択されているかを特定する方法を示します。

```
X = Patient.GetEnteredValue("0.Visit1.DEM.DEM.0.RACE", "SELECTED",
"RACEPULLDOWN",0,0)
Y = Patient.GetEnteredValue("0.Visit1.DEM.DEM.0.RACE", "SELECTED","RACETEXT",0,0)
```

```
If X = 1
    Then val = Patient.GetEnteredValue("0.Visit1.DEM.DEM.0.RACE","",0,0,0)
    If val = (test against integer)
    End If
```

```
Else if Y = 1
    Then val = Patient.GetEnteredValue("0.Visit1.DEM.DEM.0.RACE","",0,0,0)
    If val = (test against string)
    End If
```

End If

例 4：複数の選択リストの中の特定の選択された値を得る

この例は、WHATSMOKED 複数選択リスト内で選択されたアイテム数を返し、Cigars アイテムが選択されているかどうかを特定するため、選択された各アイテムをテストしています。

```
nCount=Patient.GetValue("0.Visit1.DEM.SH.0.WHATSMOKED", "COUNT", 0,0,0)
i=1
bCigars=false
Do Until i < nCount + 1
    val=Patient.GetValue("0.Visit1.DEM.SH.0.WHATSMOKED", "", i,0,0)
    if(val="cigars") then
        bCigars=true
    Exit Do
```

```

    end if
Loop
if(bCigars) then
    Result.RulePassed=1
else
    Result.RulePassed=0
End if

```

例 5：アイテムセットのデータアイテムを参照する

次の例は、itemset の中のデータアイテムをどのように参照するかを示します。このスクリプトでは、AE アイテムセットのデータアイテムである **MEASUREITEM** コンポーネントの値をチェックします。

```

Control = "0.Visit2.AE.AE.AE.MEASUREITEM"
MinVal = 40

'-----
'This statement gets the value of MEASUREITEM in the current row,
'the row the user just submitted or changed.

Value = Patient.GetEnteredValue(Control, "", 0,0,0)

'This statement gets the value of MEASUREITEM in the third row
'of the itemset, by using the ItemsetIndex.

'Value = Patient.GetEnteredValue(Control, "", 0,3,0)

If (Value>MinVal) Then
Result.RulePassed = 1
Else
Result.RulePassed = 0
End If

```

例 6：予定外のビジットにアクセスする

次の例は、予定外の各ビジットで発生するフォームの特定の条件にあるデータをどう参照するかを示します。この例では、2 番目の予定外ビジットにある **PULSERATETEXT** データアイテムからデータを入手します。最初の例では、予定外ビジットは現在のビジットです。2 番目の例では、予定外ビジットは現在のビジットではありません。

```

Control = "0.UnschVisit.VS.VS.0.PULSERATE.PULSERATETEXT"
Value = Patient.GetEnteredValue(Control, "", 0,0,0)

Control = "0.UnschVisit.VS.VS.0.PULSERATE.PULSERATETEXT"
Value = Patient.GetEnteredValue(Control, "", 0,0,2)

```

例 7：必須の日付が完全に入力されたかどうかを検証する

次の例は、**VALIDDATEMAP** を使用して、日付の必要部分が CRF にすべて入力されていることを確認しています。値は、完全な日付を構成する各コンポーネントに割り当てられています。次の例では、日が 4、月が 2、および年が 1 という値を合計すると 7 に

なります。これらの各コンポーネントにデータが存在する場合には、PFCOMPLETEDATE=7 となり、ルールを満たします。これらのコンポーネントの値のどれかが欠けると、PFCOMPLETEDATE は 7 とはならず、ルールの条件は満たされません。

```
strItem = "0.Visit1.DEM.DEM.0.DEMDOB"
```

```
PFYEAR=1
```

```
PFMONTH=2
```

```
PFDAY=4
```

```
PFCOMPLETEDATE = PFYEAR Or PFMONTH Or PFDAY 'PFCOMPLETEDATE=7
```

```
testDateMap = Patient.GetValue(strItem, "VALIDDATEMAP", 0, 0, 0)
```

```
If testDateMap=PFCOMPLETEDATE Then
```

```
    Result.RulePassed = 1
```

```
Else
```

```
    Result.RulePassed = 0 End If
```

例 8：ItemsetIndex をループする。

次の例では、GetValue で GetItemsetCount を使用して ItemsetIndex をループします。

```
ISidx = 1
```

```
vPgPath = Patient.GetArgument("PgCtrlPath")
```

```
vIScnt = CInt(Patient.GetItemsetCount(vPgPath,0))
```

```
Patient.RulePassed = 0 'failed unless we find a match
```

```
Do While (ISidx <= vIScnt)
```

```
    [CODE]
```

```
    ISidx = ISidx + 1
```

```
Loop
```

GetEnteredValueRF(REFNAMEPath,AttributeName,Index,ItemsetIndex,VisitIndex, FormIndex)

繰り返しフォームのインスタンスにおいて、指定された CRF コンポーネントに入力されたデータの値を返します。GetEnteredValueRF メソッドは、繰り返し CRF での使用を目的とすることを除き、GetEnteredValue メソッドと同じです。このため、繰り返し CRF の特定インスタンスの参照を可能にする追加の引数 FormIndex があります。

注： 繰り返し CRF 上の値を確認する場合、または、繰り返し CRF にルールを付加して繰り返しでない CRF の値を参照する場合のいずれかに、**GetEnteredValueRF** を使用します。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

AttributeName

二重引用符(“)で指定された戻り値のタイプを指定するストリング。有効な AttributeNames を以下に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。空のダブルクォーテーション(“)で示すことができます。
- **DATESTRING** — DateTime コントロールの日付の部分のみ返します。
- **TIMESTRING** — DateTime コントロールの時間の部分のみ返します。
- **DATETIMESTRING** — DateTime コントロールの日付と時刻の両方の部分を返します。
- **SELECTED** — **INDEX** 属性と一緒に使用して、指定したインデックスが選択されているかどうかを示します。
- **COUNT** — コントロールで選択されるアイテムの数を指定します。例えば、コントロールが多重選択リストボックスである場合、**COUNT** は選択済みリストアイテムの数を返します。
- **UNIT** — 選択されている場合、単位のシンボルテキストを返します。

<注> 単位の **SYMBOL** 属性は変換できます。返される値は、現在の症例施設のロケールの値になります。したがって、返されたシンボルテキストをチェックするルールのロジックは、現在の症例施設のロケールを考慮する必要があります。

- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールの要素を

示すビットマスクを含む整数を返します。

- 年 — 1
- 月 — 2
- 日 — 4
- 時間 — 8
- 分 — 16
- 秒 — 32
- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32

注：日付コンポーネントの **UNKNOWNDATEMAP** 値にビットを設定した場合、対応するビットは **VALIDDATEMAP** 値に設定されません。

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定しま

す。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

例

この例では、繰り返しでないフォーム上の 2 つのコントロールの値を取得し、繰り返しになっている現在の CRF でその値を使用することができます。

```
Control1 = "0.Visit2.VITALS.VS.0.BPITEM.BPGROUP.SYSTEXT"  
Control2 = "0.Visit2.VITALS.VS.0.BPITEM.BPGROUP.DIASTEXT"  
Value1 = Patient.GetEnteredValueRF(Control1, "", 0,0,0,0)  
Value2 = Patient.GetEnteredValueRF(Control2, "", 0,0,0,0)
```

GetFormCount(Visit,VisitIndex,Form)

指定されたビジットに対して作成された繰り返し CRF インスタンス数を返します。

引数

Visit

繰り返し CRF が発生するフォームセットの RefName。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

Form

繰り返し CRF の RefName。

例

以下の例は、入力された有害事象がすでに存在するかどうかをテストします。

GetFormCount メソッドを使用して、AE インスタンスの合計数を基にテストの実行回数を設定します。

```
patient.RulePassed = 1
Patient.AddNamedValue "QUERYTEXT", "This AE already exists"
path="0.CommonCRF.AE.sctAE.0.itmAEDdiag.txtAEDdiag"
isValid=patient.IsValidItemPathRF (path,0,0)
strVal=patient.GetValueRF(path,"",0,0,0,0)
intcurFormCount=patient.GetCurrentFormIndex()
intFormCount=patient.GetFormCount("CommonCRF",1,"AE")

for count = 1 to intFormCount
intIsDel=Patient.FormHasStateRF("CommonCRF",1,"AE",count,"deleted")
if (cint(count) <> cint(intcurFormCount)) and intIsDel=0 then
strVal2=patient.GetValueRF("0.CommonCRF.AE.sctAE.0.itmAEDdiag.txtAEDdiag","", 0, 0, 0,
count)
If Ucase(strVal) = Ucase(strVal2) then
patient.RulePassed = 0
exit for
End If
End If
Next
```

GetID()

現在の症例の ID を返します。ID は、データベース内で症例を認識するために内部的に

割り当てられた値です。

例

以下の例は、無作為割付キットナンバーの生成準備において、GetID メソッドを使用して Randomization オブジェクトのプロパティをロードする例です。

```
Randomization.SiteID = Patient.GetSiteID  
Randomization.Type = 1  
Randomization.Source = "SimpleCentral"  
Randomization.PatientID = Patient.GetID  
Randomization.Revision = 0  
SeqNum = Randomization.GetNextKit(KitInfo)  
Result.Result= SeqNum + " / " + KitInfo
```


GetItemsetCount(REFNAMEPath,VisitIndex)

アイテムセットに現在入力されている行の数を返します。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

例

以下の例は、AE アイテムセットに入力された行数を数えるため、AE アイテムセットの最初のデータアイテム AETEXTITEM を使用しています。GetItemsetCount の結果は整数として割り当てなければならないことに注意してください。

```
CheckItem = "0.Visit2.AE.AE.AE.AETEXTITEM"  
CheckCount = Cint(Patient.GetItemsetCount(CheckItem,0))
```

GetItemsetCountRF(REFNAMEPath,VisitIndex,FormIndex)

繰り返しフォームの指定されたインスタンスに発生するアイテムセットに現在入力されている行数を返します。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の

CRF を指定します。

- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

例

この例では、繰り返しでない AE フォームのアイテムセットの行数を数え、繰り返しになっている他の CRF でその数を使用できます。

```
CheckItem = "0.Visit2.AE.AE.AE.AETEXTITEM"
CheckCount = Cint(Patient.GetItemsetCountRF(CheckItem,1,1))
```

GetItemsetRowID(REFNAMEPath,VisitIndex)

アイテムセット行の一意の行 ID を返します。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

```
(0.Visit.Form.Section.Itemset.Item[.control[.control...]])
```

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用

されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

例

以下の例は、AE アイテムセットの最初のデータアイテムである AETEXTITEM のアイテムセット行 ID を取り出します。

```
CheckItem = "0.Visit2.AE.AE.AE.AETEXTITEM"
CheckID = Cint(Patient.GetItemsetRowID(CheckItem,0))
```

GetPatientNumber()

現在の症例に割り当てられている番号を返します。症例番号は、登録時に症例に割り当てられた値です。

例

GetCurrentISRowNum() の例を参照してください。

GetRefName(ObjType)

ルールが実行されるコンテキストの指定されたオブジェクトタイプ の RefName を文字列で返します。

引数

ObjType

次のうちの一つ:

- Visit
- Form
- Section
- Item
- Itemset

GetRefPath(ObjType)

ルールが実行されるコンテキストの指定されたオブジェクトタイプの RefName パスを文字列で返します。たとえば、"セクション" を指定すると、GetRefPath
0.VisitRefName.FormRefName.SectionRefName の RefName パスを返します。

引数

ObjType

次のうちの一つ:

- Visit
- Form
- Section
- Item
- Itemset

GetScreeningNumber()

現在の症例のスクリーニング番号を文字列で返します。

例

GetCurrentISRowNum() の例を参照してください。

GetServerTime()

現在のサーバの日時値を GMT として文字列で返します。

例

GetCurrentISRowNum() の例を参照してください。

GetSiteID()

現在の症例施設 ID を返します。施設 ID はデータベース内で施設を認識するために内部的に割り当てられた値です。

例

```
SiteID = Patient.GetSiteID()  
If SiteID...
```

GetSiteLocale()

現在の症例について、施設のスタディオケールを返します。たとえば、**en-US** や **ja-JP** などです。

例

GetSiteID() の例を参照してください。

GetSiteMnemonic()

現在の症例の施設の施設略名を文字列で返します。このメソッド（GetSiteName メソッドと同じ）は施設を特定する簡略化された方法です。

例

GetCurrentISRowNum() の例を参照してください。

GetSiteName()

現在の症例の施設名を返します。

例

```
SiteName = Patient.GetSiteName()  
If SiteName...
```

GetSiteTime()

現在の施設のタイムゾーンでサーバの日時値を返します。このメソッドは、未来の日付もしくは時間が CRF に入力されていないことを確認する場合に有効です。

例

GetCurrentISRowNum()の例を参照してください。

GetTrialName()

現在の症例が参加しているトライアルの名前を返します。

例

```
TrialName = Patient.GetTrialName()  
If TrialName...
```

GetUserName()

InForm アプリケーションに現在ログインしているユーザの名前を返します。

例

GetSiteID() の例を参照してください。

GetValue(REFNAMEPath,AttributeName,Index,ItemsetIndex,VisitIndex)

指定された CRF コンポーネントに入力されたデータの正規化された値を返します。アイテムが正規化されていない場合、メソッドは入力された値を返します。GetValue は、指定された CRF コンポーネントに入力されたデータの値を返す GetEnteredValue メソッドと同じです。

これらのメソッドの違いは、単位として定義される CRF コンポーネントにあります。

MedML インストラユーティリティでは、単位として指定する CRF コンポーネントを定義することができます。データをオンライン上で 1 つの単位として表示/データベースへ保存するために、他の単位を参照して単位 CRF コンポーネントを定義することができます。例えば、CRC がオンス単位かポンド単位のどちらかで体重を入力できるように CRF を設計し、データベース上ではオンスで保存されるように指定することも可能です。変換後にデータベースに保存したデータのユニットをベース単位と呼び、ベース単位に変換されたデータ値を正規化された値と呼びます。

入力されたデータ値をチェックする場合は、GetEnteredValue を使用してください。データアイテムの正規化された値を入手する際には、GetValue を使用してください。

注：繰り返し CRF のデータ値を確認する、または繰り返しでない CRF の値を参照するために繰り返し CRF にルールを付加するには、GetValueRF または GetEnteredValueRF を使用します。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含んだ .xml ファイルで指定された RefName です。

REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む.xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするには、チャイルドコンポーネントがネストされている各ペアレントコントロールを参照します。例えば、グループコントロール内の 2 つのテキストコントロールのうちの 1 つを指

定するには、以下のようにグループコントロールの RefName を入力し、ピリオドで区切ってから、その後にテキストコントロールの RefName を続けて入力します。

GroupControlRefname.TextControlRefname.

AttributeName

二重引用符(“)で指定された戻り値のタイプを指定するストリング。有効な AttributeNames を以下に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。空のダブルクォーテーション(“)で示すことができます。
- **DATESTRING** — DateTime コントロールの日付の部分のみ返します。
- **TIMESTRING** — DateTime コントロールの時間の部分のみ返します。
- **DATETIMESTRING** — DateTime コントロールの日付と時刻の両方の部分を返します。
- **SELECTED** — **INDEX** 属性と一緒に使用して、指定したインデックスが選択されているかどうかを示します。
- **COUNT** — コントロールで選択されるアイテムの数を指定します。例えば、コントロールが多重選択リストボックスである場合、**COUNT** は選択済みリストアイテムの数を返します。
- **UNIT** — 選択されている場合、単位のシンボルテキストを返します。

<注> 単位の **SYMBOL** 属性は変換できます。返される値は、現在の症例施設のロケールの値になります。したがって、返されたシンボルテキストをチェックするルールのロジックは、現在の症例施設のロケールを考慮する必要があります。

- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールのエLEMENTを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32
- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使用します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なELEMENTを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2

- 日 — 4
- 時間 — 8
- 分 — 16
- 秒 — 32

注：日付コンポーネントの **UNKNOWNDATEMAP** 値にビットを設定した場合、対応するビットは **VALIDDATEMAP** 値に設定されません。

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

例

GetEnteredValue メソッドで示した例を参照してください。

GetValueRF(REFNAMEPath,AttributeName,Index,ItemsetIndex,VisitIndex,FormIndex)

繰り返し CRF のインスタンスにおいて、指定された CRF コンポーネントに入力された

データの正規化された値を返します。アイテムが正規化されていない場合、メソッドは入力された値を返します。繰り返し CRF での使用を目的とすることを除き、GetValueRF メソッドは GetValue メソッドと同じです。このため、繰り返し CRF の特定インスタンスの参照を可能にする追加の引数 FormIndex があります。

注：繰り返し CRF 上の値を確認する場合、または、繰り返し CRF にルールを付加して繰り返しでない CRF の値を参照する場合のいずれかに、GetValueRF を使用します。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含んだ .xml ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。
(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む.xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするには、チャイルドコンポーネントがネストされている各ペアレントコントロールを参照します。例えば、グループコントロール内の 2 つのテキストコントロールのうちの 1 つを指定するには、以下のようにグループコントロールの RefName を入力し、ピリオドで区切ってから、その後にテキストコントロールの RefName を続けて入力します。
GroupControlRefname.TextControlRefname.

AttributeName

二重引用符(“)で指定された戻り値のタイプを指定するストリング。有効な AttributeNames を以下に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。空のダブルクォーテーション(“)で示すことができます。
- **DATESTRING** — DateTime コントロールの日付の部分のみ返します。
- **TIMESTRING** — DateTime コントロールの時間の部分のみ返します。
- **DATETIMESTRING** — DateTime コントロールの日付と時刻の両方の部分を返します。
- **SELECTED** — **INDEX** 属性と一緒に使用して、指定したインデックスが選択されているかどうかを示します。
- **COUNT** — コントロールで選択されるアイテムの数を指定します。例えば、コントロールが多重選択リストボックスである場合、**COUNT** は選択済みリストアイテムの数を返します。
- **UNIT** — 選択されている場合、単位のシンボルテキストを返します。

<注> 単位の **SYMBOL** 属性は変換できます。返される値は、現在の症例施設のロケールの値になります。したがって、返されたシンボルテキストをチェックするルールロジックは、現在の症例施設のロケールを考慮する必要があります。

- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールのエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32
- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使用します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32

注：日付コンポーネントの **UNKNOWNDATEMAP** 値にビットを設定した場合、対応するビットは **VALIDDATEMAP** 値に設定されません。

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

例

次の例は、有害事象の説明内容が txtAEDiag コントロールに既に入力されているかどうかをテストしています。GetValueRF メソッドを使用して、現在の CRF インスタンスのテキストコントロールに入力された値と、テストの対象となっている後続の各 CRF インスタンスに入っている値を取得し、値を比較できるようにしています。

```
patient.RulePassed = 1
Patient.AddNamedValue "QUERYTEXT", "This AE already exists"
```

```

path="0.CommonCRF.AE.sctAE.0.itmAEDdiag.txtAEDdiag"
isValid=patient.IsValidItemPathRF (path,0,0)
strVal=patient.GetValueRF(path,"",0,0,0,1)
intcurFormCount=patient.GetCurrentFormIndex()
intFormCount=patient.GetFormCount("CommonCRF",1,"AE")

for count = 1 to intFormCount
intIsDel=Patient.FormHasStateRF("CommonCRF",1,"AE",count,"deleted")
if (cint(count) <> cint(intcurFormCount)) and intIsDel=0 then
strVal2=patient.GetValueRF("0.CommonCRF.AE.sctAE.0.itmAEDdiag.txtAEDdiag","", 0, 0, 0,
count)
If Ucase(strVal) = Ucase(strVal2) then
patient.RulePassed = 0
exit forthat's
End If
End If
Next

```

GetVisitCount(VisitRefName)

VisitRefName 引数で指定されたビジットのインスタンス数を返します。このメソッドは繰り返しビジットに使用し、ビジットの全ての繰り返しをループして反復を数える定数を作成するために使用できます。

引数

VisitRefName

ビジット数を数えるビジットの RefName。

例

```

visitname = Cstr(Patient.GetArgument("visitrefname"))
visitcount = Cint(patient.Getvisitcount(visitname))

```

IsCBSelected(RefNamePath,ItemSetIndex, VisitIndex)

チェックボックスコントロールのチャイルドコントロールが選択されているかどうかを確認して、チャイルドコントロールの状態を示すブール値を返します。

- **True** — 選択
- **False** — 選択解除

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControl/Refname.TextControl/Refname*

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

例

以下の例は、チェックボックスコントロールの各シンプルコントロールを確認して、選択されているかどうかを判断します。すべてのチェックボックスが空の場合は、チェックボックスコントロールのラジオボタンが選択されたかどうかを確認するクエリを生成します。

```
cbselected = "False"
valradio = Cstr(Patient.GetValue( Patient.GetCurPath() & ".SMOKEGROUPRADIO", "", 0, 0, 0))

If CBool(patient.IsCBSelected(Patient.GetCurPath() &
    ".SMOKEGROUPRADIO.SMOKEGROUP.SMOKECHECKBOX.CIGARETTE",0,0)) =
    "True" then
    cbselected = "True"
End if

If CBool(patient.IsCBSelected(Patient.GetCurPath() &
    ".SMOKEGROUPRADIO.SMOKEGROUP.SMOKECHECKBOX.PIPE",0,0)) = "True" then
    cbselected = "True"
End if

If CBool(patient.IsCBSelected(Patient.GetCurPath() &
    ".SMOKEGROUPRADIO.SMOKEGROUP.SMOKECHECKBOX.CIGAR",0,0)) = "True"
then
    cbselected = "True"
End if

if valradio = "Smokes" then
    if cbselected = "False" then
        result.RulePassed=0
        patient.AddNamedValue "QUERYTEXT", "Please select at least one checkbox."
        patient.AddNamedValue "QUERYSTATE", "Open"
    Else
        result.RulePassed=1
    End if
Else
    result.RulePassed=1
End if
```

IsCBSelectedRF(RefNamePath,ItemsetIndex,VisitIndex,FormIndex)

繰り返し CRF のインスタンスにおいて、チェックボックスコントロールのチャイルドコントロールが選択されているかどうかを確認して、チャイルドコントロールの状態を示すブール値を返します。

- **True** — 選択
- **False** — 選択解除

繰り返し CRF での使用を目的とすることを除いて、IsCBSelectedRF メソッドは IsCBSelected メソッドと同じです。このため、繰り返し CRF の特定インスタンスの参照を可能にする追加の引数 **FormIndex** があります。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

例

以下の例は、CM CRF の現在のインスタンスの Mthd_GC コントロール内の各チェックボックスをテストして、オンになっているかどうかを判断します。オンになっているチェックボックスがなければ、ルールはクエリを発行します。

```
patient.RulePassed=1
If patient.GetValueRF("0.CommonCRF.CM.sctCM.0.itmCMMed.txtCMMed","",0,0,0,0)<>"" then
If ("True" = patient.IsCBSelectedRF
("0.CommonCRF.CM.sctCM.0.itmMthd.Mthd_GC.Mthd_l_CBC.l_SC",0,0,0)) or _
("True" = patient.IsCBSelectedRF
("0.CommonCRF.CM.sctCM.0.itmMthd.Mthd_GC.Mthd_i_CBC.i_SC",0,0,0)) or _
("True" = patient.IsCBSelectedRF
("0.CommonCRF.CM.sctCM.0.itmMthd.Mthd_GC.Mthd_t_CBC.t_SC",0,0,0)) then
Else
patient.RulePassed=0
patient.AddNamedValue "QUERYTEXT", "Medication is present but at least 1 method has not
been selected in item 6. Please select at least one checkbox in item 6."
End If
End If
```

IsItemSetDeleted(REFNAMEPath,VisitIndex,ItemSetIndex)

エントリ追加アイテムセットの中の行が削除されているかどうかを示します。行が削除

されていない場合は 0、削除されている場合には 1 を返します。アイテムセットの中に重複する情報がないかどうかチェックするルールを使用していて、削除した情報があるという場合、ルールにこれらの削除された行のアイテムを含めないようにします。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。

有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

例

```
if (Patient.IsItemSetDeleted(Patient.GetCurPath(),0,0)) then
    Result.RulePassed = 0
else
    Result.RulePassed = 1
End if
```

IsItemSetDeletedRF(REFNAMEPath,VisitIndex,FormIndex,ItemsetIndex)

繰り返し CRF のインスタンスにおいて、エントリ追加アイテムセットの中の行が削除されているかどうかを示します。行が削除されていない場合にはゼロ、行が削除されている場合には 1 を返します。アイテムセットの中に重複する情報がないかどうかチェックするルールを使用していて、削除した情報があるという場合、ルールにこれらの削除された行のアイテムを含めないようにします。

繰り返し CRF における使用を目的とすることを除いて、IsItemSetDeletedRF メソッドは IsItemSetDeleted メソッドと同じです。このため、繰り返し CRF の特定インスタンスの参照を可能にする追加の引数 FormIndex があります。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、

併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。

- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControl/Refname.TextControl/Refname*

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

例

この例は、繰り返しフォームインスタンス内にある現在のアイテムセット行が削除されているかどうかをチェックします。

```

if (Patient.IsItemSetDeletedRF(Patient.GetCurPath(),0,0,0)) then
Result.RulePassed = 0
else
Result.RulePassed = 1
End if

```

IsValidItemPath(RefNamePath,VisitIndex)

指定されたパスの存在をデータベースでチェックし、RefName パスが有効であるかどうかを確認します。施設のスタディバージョンが異なり、一部のスタディバージョンにのみ存在するアイテムを更新する計算を行う場合は、このメソッドを使用して、計算実行前に目的のアイテムが存在するかどうかをチェックします。さらに、このメソッドを使用して、計算で CRF の新しいバージョンを更新するか、または以前のスタディバージョンで CRF を開始した症例について代替 CRF を使用して同じデータを入手するかの判断を行うこともできます。メソッドは、パスが存在する場合は 1(True)を、パスが存在しない場合は 0(False)を返します。

注：IsValidItemPath は、計算によってアイテムを更新したい場合に、計算されたアイテムへのパスの存在するかどうかをチェックするためにのみ使用してください。**CRF** ルールにこのメソッドを使用しないでください。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を

区切ってください。 *GroupControl/Refname.TextControl/Refname*

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

例

この例では、アイテムが CRF の現行バージョンに存在している場合、計算値はそのバージョンで更新されます。そうでない場合、値は代替 CRF で更新されます。

```
dpath=0.Visit1.DEM.DEM.0.NEWITEM.NEWITEM_TB
altdpath=0.Visit1.ALTDEM.ALTDEM.0.NEWITEM.NEWITEM_TB
val="Patient.GetEnteredValue("0.Visit1.DEM.DEM.0.WEIGHT.WEIGHT*TEXT",0)"
newval=val * .625
```

```
If Patient.IsValidItemPath("dpath,0") then
    Patient.SetEnteredValue dpath,"", 0, 0, 0,newval
else Patient.SetEnteredValue altdpath,"", 0, 0, 0,newval
end if
```

IsValidItemPathRF(RefNamePath,Visit,Index,FormIndex)

繰り返しフォームのインスタンスにおいて、指定されたパスの存在をデータベースでチェックし、RefName パスが有効であるかどうかを確認します。施設のスタディバージョンが異なり、一部のスタディバージョンにのみ存在するアイテムを更新する計算を行う場合は、このメソッドを使用して、計算実行前に目的のアイテムが存在するかどうかをチェックします。さらに、このメソッドを使用して、計算で CRF の新しいバージョンを更新するか、または以前のスタディバージョンで CRF を開始した症例について代替 CRF を使用して同じデータを入手するかの判断を行うこともできます。メソッドは、パスが存在する場合は 1(True)を、パスが存在しない場合は 0(False)を返します。

繰り返し CRF における使用を目的とすることを除いて、IsValidItemPathRF メソッドは IsValidItemPath メソッドと同じです。このため、繰り返し CRF の特定インスタンスの参照を可能にする追加の引数 FormIndex があります。

注 : IsValidItemPathRF は、計算によってアイテムを更新したい場合に、計算されたアイテムへのパスの存在するかどうかをチェックするためにのみ使用してください。CRF ルールにこのメソッドを使用しないでください。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

例

この例では、アイテムが CRF の現行バージョンに存在している場合、計算値はそのバージョンで更新されます。そうでない場合、値は代替 CRF で更新されます。

```
dpath=0.vstCORE2.0.frmECG.sctECG.0.mitmQTRate.calQTRate
altdpath=0.vstCORE2.0.frmALTECG.sctALTECG.0.NEWitmQTRate.NEWcalRate
val="Patient.GetEnteredValueRF("0.vstCORE2.0.frmALTECG.sctALTECG.0.mitmQRSRate.mtxt
QRSRate",0")
newval=val * .625
If Patient.IsValidItemPathRF("dpath,0,0") then
    Patient.SetEnteredValueRF dpath,"",0,0,0,newval
else Patient.SetEnteredValueRF altdpath,"",0,0,0,newval
end if
```

ItemHasBeenNormalizedEx(REFNAMEPath,ItemsetIndex,VisitIndex)

データが正規化されたかどうか確認することによって、参照されるデータアイテムに関連する単位が選択されているかを示します。ユーザがデータ値を入力してデータをサブミットするとき、ユーザのルールと関連付けられている単位を選択した場合は、単位変換ルールの ID がルールエンジンに渡されます。ルールエンジンが変換ルール ID を取得した場合、ItemHasBeenNormalizedEx メソッドは true を返します。そうでない場合、メソッドは false を返します。

注：このメソッドを使用可能にするために、ベース単位を含むそれぞれのユニットが変換ルールに関連付けられていることを確認しなければなりません。ベース単位の変換ルールは **Data.Result=Data.BaseValue*1** である必要があります。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

```
(0.Visit.Form.Section.Itemset.Item[.control[.control...]])
```

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、

併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。

- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControl/Refname.TextControl/Refname*

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

例

```
UnitFilled = Patient.ItemHasBeenNormalizedEx("0.Visit1.DEM.DEM.0.HEIGHT",0,0)
```

ItemHasBeenNormalizedRF(REFNAMEPath,ItemsetIndex,VisitIndex,FormIndex)

繰り返し CRF の特定インスタンスで発生するアイテムについては、データが正規化されたかどうか確認することによって、参照されるデータアイテムに関連する単位が選択されているかを示します。ユーザがデータ値を入力してデータをサブミットするとき、ユーザのルールと関連付けられている単位を選択した場合は、単位変換ルールの ID がルールエンジンに渡されます。ルールエンジンが変換ルール ID を取得した場合、

ItemHasBeenNormalizedRF メソッドは true を返します。そうでない場合、メソッドは false を返します。

繰り返し CRF における使用を目的とすることを除いて、ItemHasBeenNormalizedRF メソッドは ItemHasBeenNormalizedEx メソッドと同じです。このため、繰り返し CRF の特定インスタンスの参照を可能にする追加の引数 FormIndex があります。

注：このメソッドを使用可能にするために、ベース単位を含むそれぞれのユニットが変換ルールに関連付けられていることを確認しなければなりません。ベース単位の変換ルールは **Data.Result=Data.BaseValue*1** であることが必要です。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照しま

す。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。

- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

例

この例では、ItemHasBeenNormalized を使用して、繰り返し CRF の現在のインスタンスの中のコントロールが選択されているかどうかをチェックします。

```
IsSelected =
Patient.ItemHasBeenNormalizedRF(Patient.GetCurPath(),0,0,Patient.GetCurrentFormIndex())
patient.rulepassed = 1
Patient.AddNamedValue "QUERYTEXT", "Quantity is present, but measurements are not selected;
please complete"
if IsSelected = 0 then
patient.rulepassed = 0
end if
```

LoadCRB()

現在の症例の CRB のデータをすべてメモリにロードします。これは、複数のビジットおよび CRF における複数のデータアイテムをルールでチェックする必要がある場合に、何度もデータベースを検索しなくてもデータを見つけることができるため効果を発揮します。

RemoveEmptyDynamicVisit

ビジットの CRF のいずれにデータが入力されていない場合、症例にスケジューリングされているダイナミックビジットを削除します。このメソッドの唯一の引数は、削除す

るダイナミックビジットの RefName です。

ダイナミックビジットの削除

ダイナミックビジットで症例データがどの CRF にも入力されていない場合、RemoveEmptyDynamicVisit メソッドを呼び出してビジットを削除できます。RemoveEmptyDynamicVisit にはパラメータが 1 つあります。つまり、ダイナミックビジットの RefName です。

例

この例では、ある症例がスケジューリングされたダイナミックビジットから誤って削除され、その症例の性別に該当するビジットに再スケジューリングされます。

```
MALE=1
FEMALE=2
MALEVISIT="Dose100"
FEMALEVISIT="Dose200"
Gender = Patient.GetValue("0.Visit1.DEM.DEM.0.GENDER","",0,0,0)

If Gender = MALE Then
  □□ SVisit="MALEVISIT"
  □□ RVisit="FEMALEVISIT"
Else
  □□ SVisit="FEMALEVISIT"
  □□ RVisit="MALEVISIT"
End If
Patient.RemoveEmptyDynamicVisit RVisit
Patient.SetDynamicVisitStart SVisit, "", 720
```

RemoveNotStartedDynamicForm

ダイナミックフォームにデータが未入力の場合、症例の CRF から Patient オブジェクトまたは Result オブジェクトの RemoveNotStartedDynamicForm メソッドを使用して削除することができます。

RemoveNotStartedDynamicForm には、以下のパラメータがあります。

- CRF が表示される、またはその可能性があるビジットの RefName。
- スケジュールから取り除くダイナミック CRF の RefName。
- Visit Index。これは繰り返しビジットにインデックスを指定する数です。値は以下の通りです。
 - 0 — ビジットが繰り返されていないことを示します。または、ビジットが繰り返している場合、現在のビジットのインスタンスを参照します。
 - 0 より大きい数 — ビジットの指定された箇所を明確にインデックス指定します。

例

```
MALE=1
FEMALE=2

Gender=Patient.GetValue("0.visit1.DEM.DEM.0.GENDER", "", 0, 0, 0)
```

```

If Gender=FEMALE Then
Patient.ShowDynamicForm "visit1" , "Preg", 0
Else
Patient.RemoveNotStartedDynamicForm "visit1", "Preg", 0
End If

```

症例 (症例) の性別が女性である場合、"Preg" ダイナミックフォームがビジットに表示されます。性別が女性でない場合、CRF はビジットから削除され、症例 (症例) に対してスケジューリングされません。

SetDynamicVisitStart

ダイナミックビジットに症例をスケジューリングします。メソッドの引数には、

- 予定するダイナミックビジットの RefName。
- ダイナミックビジットの前のビジットの RefName。
- ダイナミックビジットが始まった前のビジットからの時間数。前のビジットが指定されていない場合、時間数はスタディの初診から数えられます。

症例をダイナミックビジットスケジュールに割り当てるには、ビジット割り当ての基となるアイテムの値をテストする計算ルールスクリプトを作成し、割り当てを行います。Patient オブジェクトか Result オブジェクトに対しては、SetDynamicVisitStart メソッドを使用してください。

例 1

この例では、症例の性別によって、症例を Dose100 ビジットか Dose250 ビジットのどちらにスケジュールするかが決まります。このビジットはどちらも最初のビジットから 720 時間後に始まります。

```
MALE=1
```

```
FEMALE=2
```

```
Gender = Patient.GetValue("0.Visit1.DEM.DEM.0.GENDER","",0,0,0)
```

```
If Gender = MALE Then
```

```
    DVisit="Dose100"
```

```
Else
```

```
    DVisit="Dose250"
```

```
End If
```

```
Patient.SetDynamicVisitStart DVisit, "", 720
```

例 2

この例では、症例の性別によって、前のビジット Vist3 の 10 日または 2 週間後に、Visit4Gen ダイナミックビジットを開始するかどうかが決まります。

```
MALE=1
```

```
FEMALE=2
```

```
Gender = Patient.GetValue("0.Visit1.DEM.DEM.0.GENDER","",0,0,0)
```

```
If Gender = MALE Then
```

```
    StartAt=240
```

```

Else
  StartAt=336
End If

Patient.SetDynamicVisitStart "Visit4Gen", "Visit3", StartAt

```

SetEnteredValue(REFNAMEPath,AttributeName,Index,ItemsetIndex,VisitIndex,Value)

指定されたフォームコンポーネントに入力されたデータの値を使用します。SetEnteredValue は、指定されたフォームコンポーネントに入力されたデータの正規化された値を使用する SetValue メソッドに非常に似ています。これらのメソッドの違いは、単位として定義される CRF コンポーネントにあります。

MedML インストーラユーティリティでは、単位として指定する CRF コンポーネントを定義することができます。データをオンライン上で 1 つの単位として表示/データベースへ保存するために、他の単位を参照して単位 CRF コンポーネントを定義することができます。例えば、CRC がオンス単位かポンド単位のどちらかで体重を入力できるように CRF を設計し、データベース上ではオンスで保存されるように指定することも可能です。変換後にデータベースに保存したデータのユニットをベース単位と呼び、ベース単位に変換されたデータ値を正規化された値と呼びます。

<注> 単位の **SYMBOL** 属性は変換できます。メソッドを呼び出すときに、ルールまたは計算文をその変換したもののユニットシンボルの値に渡すことができます。値を設定しているフォームの **CRF** ロケールの変換に渡す必要はありません。

データアイテムの入力した値を使用する際には、SetEnteredValue を使用してください。データアイテムの正規化した値を使用しなければならないときには SetValue を使用してください。

注： SetEnteredValue は計算でのみ使用し、CRF ルールでは使用しないでください。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

```
(0.Visit.Form.Section.Itemset.Item[.control[.control...]])
```

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセット

の RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。

- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

AttributeName

二重引用符 ("") で指定された戻り値のタイプを指定するストリング。有効な AttributeNames を次に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。空のダブルクォーテーション ("") で示すことができます。
- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールのエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32
- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使用します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32

注：日付コンポーネントの **UNKNOWNDATEMAP** 値にビットを設定した場合、対応するビットは **VALIDDATEMAP** 値に設定されません。

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

値

設定する値。

例

次の例では、SetEnteredValue が取り出した施設番号に症例 ID が追加されています。

```
v2subno="0.V2.SUBID.SUBID_S.0.SUBNO"
```

```
v2scrnno="0.V2.SUBID.SUBID_S.0.SCNUM"
```

```
pn1 = cstr(Patient.GetValue(v2subno, "", 0, 0, 0))
```

```
sitenm_g=Patient.GetSiteName()
```

```
siteno = cSTR(mid(sitenm_g,2,2))
```

```
If isnumeric(pn1) and len(pn1)=2 then
```

```

Patient.SetEnteredValue v2subno,"", 0, 0, 0,siteno&pn1
else if len(pn1)=3 and cint(pn1)>600 and cint(pn1)<700 then
  Patient.SetEnteredValue v2subno,"",0,0,0,"S"&siteno&pn1
  Patient.SetEnteredValue v2scrnno,"",0,0,0,"S"&siteno&pn1
end if

end if

```

SetEnteredValueRF(REFNAMEPath,AttributeName, Index, ItemsetIndex, VisitIndex, FormIndex, Value)

繰り返しフォームのインスタンスにおいて、指定された CRF コンポーネントに入力されたデータの値を使用します。SetEnteredValueRF メソッドは、繰り返し CRF での使用を目的とすることを除き、SetEnteredValue メソッドと同じです。このため、繰り返し CRF の特定インスタンスの参照を可能にする追加の引数 FormIndex があります。

<注> 単位の **SYMBOL** 属性は変換できます。メソッドを呼び出すときに、ルールまたは計算文をその変換したもののユニットシンボルの値に渡すことができます。値を設定しているフォームの **CRF** ロケールの変換に渡す必要はありません。

SetEnteredValueRF は計算でのみ使用し、**CRF** ルールでは使用しないでください。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を

区切ってください。 *GroupControlRefname.TextControlRefname*

AttributeName

二重引用符 ("") で指定された戻り値のタイプを指定するストリング。有効な AttributeNames を次に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。
空のダブルクォーテーション ("") で示すことができます。
- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールのエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32
- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使用します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32

注：日付コンポーネントの **UNKNOWNDATEMAP** 値にビットを設定した場合、対応するビットは **VALIDDATEMAP** 値に設定されません。

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータ

アイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

値

設定する値。

例

この例では、strSetBox2Path 変数によって識別されるコントロールの値が「*****」に設定されています。

```
strSetBox2Value = Patient.GetValueRF(strSetBox2Path,"",0,0,0,0)
If strSetBox2Value <> "" then
  Patient.SetEnteredValueRF strSetBox2Path,"",0,0,0,0,"*****"
End If
```

SetNewValue(REFNAMEPath,AttributeName,Index,ItemsetIndex,VisitIndex,Value)

SetNewValue メソッドは内部 Oracle 専用です。

SetValue メソッドのように、指定された CRF コンポーネントに入力したデータの値を使

用します。SetNewValue はデータを一度も入力したことがないアイテム専用です。これは、症例登録 CRF からスタディの他の CRF にデータをマッピングする特殊な登録マッピング計算で使用することを目的としています。

<注> 単位の **SYMBOL** 属性は変換できます。メソッドを呼び出すときに、ルールまたは計算文をその変換したもののユニットシンボルの値に渡すことができます。値を設定しているフォームの **CRF** ロケールの変換に渡す必要はありません。

CRF ルールで **SetNewValue** を使用しないでください。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

AttributeName

二重引用符 ("") で指定された戻り値のタイプを指定するストリング。有効な AttributeNames を次に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。空のダブルクォーテーション ("") で示すことができます。
- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールの要素を

示すビットマスクを含む整数を返します。

年	1
月	2
日	4
時	8
分	16
秒	32

- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使用します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なエレメントを示すビットマスクを含む整数を返します。

年	1
月	2
日	4
時	8
分	16
秒	32

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

値

設定する値。

SetNewValueRF(REFNAMEPath,AttributeName,Index,ItemsetIndex,VisitIndex,FormIndex,Value)

SetNewValueRF メソッドは内部 Oracle 専用です。

SetValueRF メソッドのように、指定された CRF コンポーネントに入力したデータの値を使用します。SetNewValueRF はデータを一度も入力したことがないアイテム専用です。これは、症例登録 CRF からスタディの他の CRF にデータをマッピングする特殊な登録マッピング計算で使用することを目的としています。

SetNewValueRF メソッドは、繰り返し CRF 専用という以外 SetNewValue メソッドと同じです。このため、繰り返し CRF の特定インスタンスの参照を可能にする追加の引数 FormIndex があります。

<注> 単位の **SYMBOL** 属性は変換できます。メソッドを呼び出すときに、ルールまたは計算文をその変換したもののユニットシンボルの値に渡すことができます。値を設定しているフォームの **CRF** ロケールの変換に渡す必要はありません。

CRF ルールで **SetNewValueRF** を使用しないでください。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで指定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

AttributeName ダブルクォーテーション (") で指定された情報のタイプを指定する文字

列。有効な AttributeNames を次に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。空のダブルクォーテーション ("") で示すことができます。
- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールのエレメントを示すビットマスクを含む整数を返します。

年	1
月	2
日	4
時	8
分	16
秒	32

- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使用します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なエレメントを示すビットマスクを含む整数を返します。

年	1
月	2
日	4
時	8
分	16
秒	32

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールが

アイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。

- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

値

設定する値。

例

この例は、以前未入力だった繰り返し CRF の値を別の CRF のコントロールの値に設定します。

```
Sub CopyDateValue(strGetPath, strSetPath)
  x = Patient.GetValueRF(strGetPath, "", 0, 0, 0, 0)
  y = CStr(x)
  If y <> "" Then Patient.SetNewValueRF strSetPath, "", 0, 0, 0, x
End Sub
```

SetValue(REFNAMEPath,AttributeName,Index,ItemsetIndex,VisitIndex,Value)

指定された CRF コンポーネントに入力されたデータの正規化された値を使用します。SetValue は、指定された CRF コンポーネントに入力されたデータ値を使用する SetEnteredValue メソッドに非常に似ています。これらのメソッドの違いは、単位として定義される CRF コンポーネントにあります。

MedML インストラユーティリティでは、単位として指定する CRF コンポーネントを定義することができます。データをオンライン上で 1 つの単位として表示/データベ

スへ保存するために、他の単位を参照して単位 CRF コンポーネントを定義することができます。例えば、CRC がオンス単位かポンド単位のどちらかで体重を入力できるように CRF を設計し、データベース上ではオンスで保存されるように指定することも可能です。変換後にデータベースに保存したデータのユニットをベース単位と呼び、ベース単位に変換されたデータ値を正規化された値と呼びます。

<注> 単位の **SYMBOL** 属性は変換できます。メソッドを呼び出すときに、ルールまたは計算文をその変換したもののユニットシンボルの値に渡すことができます。値を設定しているフォームの **CRF** ロケールの変換に渡す必要はありません。

データアイテムの入力した値を使用する際には、`SetEnteredValue` を使用してください。データアイテムの正規化した値を使用しなければならないときには `SetValue` を使用してください。

注 : `SetValue` は計算でのみ使用し、**CRF** ルールでは使用しないでください。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 `RefName` は、CRF コンポーネント定義を含む XML ファイルで指定された `RefName` です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの `RefName`。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の `RefName`。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの `RefName`。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの `RefName`。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの `RefName`。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの `RefName`。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの `RefName` を入力した後で、テキストコントロールの `RefName` を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

AttributeName

二重引用符 ("") で指定された戻り値のタイプを指定するstring。有効な

AttributeNames を次に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。
空のダブルクォーテーション ("") で示すことができます。
- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールのエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32
- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使用します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32

注：日付コンポーネントの **UNKNOWNDATEMAP** 値にビットを設定した場合、対応するビットは **VALIDDATEMAP** 値に設定されません。

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。.

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事を示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

値

設定する値。

例

この例は、データアイテムの値を空にするため SetValue と特別なキーワード「EMPTY」を使用します。

```
v2subno = Cint(Patient.Getvalue(Patient.GetCurPath(),"",0,0,0))
```

```
Patient.SetEnteredValue v2subno,"", 0, 0, 0,EMPTY
```

SetValueRF(REFNAMEPath,AttributeName,Index,ItemsetIndex,VisitIndex,FormIndex,Value)

繰り返し CRF のインスタンスにおいて、指定された CRF コンポーネントに入力されたデータの正規化された値を使用します。SetValueRF メソッドは、繰り返し CRF 専用と

いう以外 SetValue メソッドと同じです。このため、繰り返し CRF の特定インスタンスの参照を可能にする追加の引数 FormIndex があります。

<注> 単位の **SYMBOL** 属性は変換できます。メソッドを呼び出すときに、ルールまたは計算文をその変換したもののユニットシンボルの値に渡すことができます。値を設定しているフォームの **CRF** ロケールの変換に渡す必要はありません。

SetValueRF は計算でのみ使用し、**CRF** ルールでは使用しないでください。

引数

RefNamePath

取得する CRF コンポーネントを特定する文字列。文字列中の各 RefName は、CRF コンポーネント定義を含む XML ファイルで指定された RefName です。REFNAMEPath には完全なパスを指定します。そのパスの症例コンポーネントだけが、現在の症例をデフォルトとします。REFNAMEPath 引数は以下の形式です。

(0.Visit.Form.Section.Itemset.Item[.control[.control...]])

- **0** — 現在の症例。
- **Visit** — ビジット定義を含む .xml ファイルで指定された、ビジットの RefName。
- **Form** — CRF 定義を含む .xml ファイルで指定された、CRF の RefName。
- **Section** — セクション定義を含む .xml ファイルで指定された、セクションの RefName。
- **Itemset** — アイテムセット定義を含む .xml ファイルで指定された、アイテムセットの RefName。アイテムセットは繰り返しデータアイテムグループです。たとえば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。
- **Item** — アイテム定義を含む .xml ファイルで指定された、アイテムの RefName。
- **Control** — コントロール定義を含んでいる .xml ファイルで特定された、コントロールの RefName。グループコントロールのコンポーネントにアクセスするためには、チャイルドコンポーネントがネスト化されているそれぞれのペアレントコントロールを参照してください。たとえば、グループコントロール内の 2 つのうちの 1 つのテキストコントロールを扱うには、グループコントロールの RefName を入力した後で、テキストコントロールの RefName を入力し、以下のようにピリオドで名前を区切ってください。 *GroupControlRefname.TextControlRefname*

AttributeName

二重引用符 ("") で指定された戻り値のタイプを指定するストリング。有効な AttributeNames を次に示します。

- **VALUE** — CRF コンポーネントの値を取得します。この属性はデフォルトです。空のダブルクォーテーション ("") で示すことができます。
- **VALIDDATEMAP** — 日付が完全であるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **VALIDDATEMAP** 属性を Patient オブジェクトで使用します。**VALIDDATEMAP** は完全な状態の DateTime コントロールのエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16

- 秒 — 32
- **UNKNOWNDATEMAP** — Unknown とマークされたコンポーネントが日付にあるかどうかを指定する数。このスクリプトでは、GetEnteredValue メソッドの **UNKNOWNDATEMAP** 属性を Patient オブジェクトで使⽤します。**UNKNOWNDATEMAP** は DateTime コントロールの不明なエレメントを示すビットマスクを含む整数を返します。
 - 年 — 1
 - 月 — 2
 - 日 — 4
 - 時間 — 8
 - 分 — 16
 - 秒 — 32

注：日付コンポーネントの **UNKNOWNDATEMAP** 値にビットを設定した場合、対応するビットは **VALIDDATEMAP** 値に設定されません。

Index

チェックボックスリストなどの複数の値を選択できるコントロールで選択したアイテムのリストのインデックスで、0 から始まります。**INDEX** 属性を使用するには、最初に、選択されるアイテム数を決定するためのコントロールのカウントを返すか、またはリストをループして選択されたアイテムを検索し、選択されたアイテムのリストを作成します。

ItemSetIndex

アイテムセットを構成するアイテムのインデックス。アイテムセットは繰り返しデータアイテムグループです。例えば、併用薬用の CRF には、症例が使用している各併用薬について同じデータアイテムを取得するアイテムセットが含まれている場合があります。有効な ItemSetIndex 値は 0 以上です。

- **0** — コントロールがアイテムセットでないことを示すか、もしくはコントロールがアイテムセットである場合アイテムセットの中の現在のデータアイテムを参照します。例えば、アイテムセットの第 3 データアイテムのデータを入力した後に、ユーザが [サブミット] をクリックした場合、0 は第 3 データアイテムにインデックスを設定します。
- **0 より大きい数** — アイテムセットの指定された箇所を明確にインデックス指定します。

VisitIndex

繰り返しフォームセットにインデックスを指定します。繰り返しフォームセットは、主にビジットが起こるたびに同一データが記録されるような、予定外のビジットに使用されます。有効な VisitIndex 値は 0 以上です。

- **0** — フォームセットが繰り返されていない事示します。またはフォームセットが繰り返している場合、現在のフォームセットを参照します。
- **0 より大きい数** — フォームセットの指定された箇所を明確にインデックス指定します。

FormIndex

繰り返し CRF にインデックスを指定します。

- **0** — CRF が繰り返すか繰り返さないかを表します。CRF が繰り返す場合、現在の CRF を指定します。
- **0 より大きい数** — 繰り返し CRF の指定された箇所を明確にインデックス指定します。

値

設定する値。

例

この例では、strSetBox2Path 変数によって識別されるコントロールの値が「*****」に設定されています。

```
strSetBox2Value = Patient.GetValueRF(strSetBox2Path,"",0,0,0,0)
```

```
If strSetBox2Value <> "" then
Patient.SetValueRF strSetBox2Path,"",0,0,0,0,"*****"
End If
```

ShowDynamicForm

他の CRF での計算結果を基にして、ビジットにダイナミック CRF を挿入します。

ShowDynamicForm には、以下のパラメータがあります。

- CRF の表示をスケジューリングするビジットの RefName。
- スケジューリングする CRF の RefName。
- Visit Index。これは繰り返しビジットにインデックスを指定する数です。値は以下の通りです。
 - **0** — ビジットが繰り返されていないことを示します。または、ビジットが繰り返している場合、現在のビジットのインスタンスを参照します。
 - **0 より大きい数** — ビジットの指定された箇所を明確にインデックス指定します。

例

この例では、症例の性別によって、ビジットに妊娠テスト CRF を表示するかどうかを決定します。

```
MALE=1
FEMALE=2

Gender=Patient.GetValue("0.visit1.DEM.DEM.0.GENDER", "", 0, 0, 0)
If Gender=FEMALE Then
Patient.ShowDynamicForm "visit1" , "Preg", 0
Else
Patient.RemoveNotStartedDynamicForm "visit1", "Preg", 0
End If
```

ルールは特殊ビジット (Special Visit) コンテキストとして DEM フォームの上の GENDER アイテムに付加されています。

第 3 章

InForm データインポートユーティリティ

この章の内容

InForm データインポートユーティリティの概要.....	342
XML ファイルのインポート.....	346
データおよびマップファイルを利用するインポート.....	367
エラーファイルのチェック.....	380
日付と時間の妥当性チェック.....	381
コマンドラインからの InForm データインポートユーティリティの実行.....	383
データインポートの拡張.....	386

InForm データインポートユーティリティの概要

インポートユーティリティの構成

項目	説明
データファイル	次を含む、パイプ区切りファイルです。 <ul style="list-style-type: none"> 行間の改行文字および復改文字 スタディデータベースにロードするデータ。
マップファイル	データファイルを InForm データベースにインポートする際に必要なすべてのマッピング情報を含むファイルです。
XML ファイル	スタディデータベースに直接インポートできる情報を含むファイルです。詳細については、「 データおよびマップファイルの作成 」（367ページ）および「 新しい臨床データの追加 」（350ページ）を参照してください。
ルール	XML ファイルがスタディ基準に従っているかどうかを確認するための、照合先となる情報です。 ルールの使用目的は、次の通りです。 <ul style="list-style-type: none"> データのクエリを出す。 データを検証する。 <注> ルールを実行すると、インポートの実行速度が落ちる場合があります。
InForm アプリケーション サーバ	InForm アプリケーションが実行されているサーバです。
スタディ データベース	スタディのためのデータがあるデータベースです。

インポート方法

InForm データインポートユーティリティでは、次のようなデータロードの方法が利用できます。

- データファイルのインポート** — マップファイルを使用してインポートされたデータと InForm データベーステーブル間のマッピングを定義します。次に、InForm データインポートユーティリティが、パイプ (|) 区切りファイルのデータを InForm データベースにロードします。この入力ファイル形式でインポートを実行するには、[InForm データ インポート] メインウィンドウで **[InForm データとマップファイル]** オプションを選択します。

この方法の場合、InForm アプリケーションサーバによって、データベースに書き込

む前にデータを検証する目的でエディットチェックが実行された後データが処理されます。

- **XML ファイルのインポート** — XML ファイルのデータを InForm データベースにロードするか、選択した症例レコードをある施設から別の施設へと移動します。この入力ファイル形式でインポートを実行するには、[InForm データインポート] メインウィンドウで [MedML ファイル] オプションを選択します。

この方法の場合、インポート中にルールを実行することができます。ルールを実行すると、アプリケーションはオンラインでデータを入力しているかのようにデータを処理します。またエディットチェックが実行され、チェックに失敗したデータに関するクエリが生成されます。

<注> InForm データインポートユーティリティでは、規制ドキュメントまたはビジットレポートのデータのインポートはできません。

要注意点

- **新規症例データのインポート** — スクリーニングおよび症例登録のプロセスを完了していない症例のデータをインポートするには、MedML ファイルオプションを使用する必要があります。

データおよびマップファイルオプションを使用して、スクリーニングおよび症例登録のデータをインポートすることはできません。

注： InForm アプリケーションでステータスが [フリーズ済] または [ロック済] のフォームに新規データをインポートすることはできません。

- **コメントのインポート** — フォームレベルコメントをインポートするには、MedML ファイルオプションを使用する必要があります。
アイテムレベルコメントをインポートするには、データおよびマップファイルオプション、MedML ファイルオプションのいずれかを使用する必要があります。
- **計算コントロールのインポート** — 計算コントロールをインポートするときは、インポートフィールドのデータタイプ定義がテキストコントロールであることが必要です。整数または浮動小数が使用されている場合、一致しているかどうかは認識されず、データは更新されません。その結果、データは新規の行として追加されます。
- **単位のインポート** — 単位データをサブミットするとき、単位がデータおよびマップファイルの直前のフィールドに関連付けられている必要があります。
- **予定外のビジットデータのインポート** — データを予定外のビジットにインポートするには、MedML ファイルオプション、データおよびマップファイルオプションのいずれかを使用します。
- **複数のサーバへのインポート** — スタディを複数のサーバ上で実行している場合は、InForm データインポートユーティリティを 1 回あたり 1 台のサーバのみで実行して、重複データがインポートされないようにします。
- **繰り返しフォームの編集** — 繰り返しフォームインスタンスを作成または編集するには、InForm データインポートユーティリティのユーザインターフェースではなく、

テキストエディタを使用して、マップファイルを作成する必要があります。次の手順に従って、マップファイルを編集します。

- マップファイルの最初の行に NOFORMNEW を適用し、InForm データインポートユーティリティに新規繰り返しフォームインスタンスを作成しないように指示します。
- マップファイルで新規 !formmatch! エレメントを使用して、他の繰り返しフォームインスタンスと比較されるアイテム (アイテムセットにはないもの) を指定します。

複数の !formmatch! エレメントを使用できます。すべてのエレメントが既存の繰り返しフォームインスタンスのいずれかと一致する場合、そのインスタンスは更新されます。一致しない場合 (または !formmatch! エレメントがない場合)、繰り返しフォームインスタンスが新規に作成されます。次にサンプルコードを示します。

```
FORMNOVISITNEWNOFORMNEW|
lcd!Site|
lcd!Patient|
!visitmatch!0.UnschVisit.DOV.DOV.0.DOV.DOV!dtdatetime!|
!formmatch!0.UnschVisit.HH.DH.0.DURATIONGROUP.DURTAIONGROUP.YRDUR
ATION!dtstring!|
!formmatch!0.UnschVisit.HH.DH.0.DURATIONGROUP.DURTAIONGROUP.MTHDU
RATION!dtstring!|
0.UnschVisit.HH.DH.0.previousgroup.previousgroup!DTSTRING!|
```

- 繰り返しフォームを作成するときは、フォームを一意に識別できるように、アイテムセットではなく通常のアイテムを含めます。フォームを一意に識別できない場合、フォームへのデータ入力は 1 回のみ可能となり、次のデータ入力は新規フォームインスタンスに対して行われることになります。
- **英数字以外の文字を含むファイルのインポート** — 日本語など英数字以外の文字を含むインポートファイルは、すべて Unicode 形式にする必要があります。他の形式はすべて ASCII として読み取られるため、間違ったエラー メッセージが作成される可能性があります。
- **エントリ追加アイテムセットへのデータのインポート** — エントリ追加アイテムセットには行を追加できます。また、既存のエントリ追加アイテムセット行にデータを追加したり、データを変更することも可能です。
 - エントリ追加アイテムセットに行を追加するには、PATIENTDATA タグを使用します。アイテムセットには、1 度に 1 行のみ追加できます。
詳細については、「**初診被験者の臨床データの追加**」(350ページ)を参照してください。
 - エントリ追加アイテムセットのアイテムにデータを追加する、またはデータを変更するには、EDITPATIENTDATA タグを使用します。
 - エントリ追加アイテムセットのアイテムにデータを追加する、またはデータを変更するには、そのデータに前からデータが含まれていたかどうかにかかわらず、EDITPATIENTDATA タグを使用します。
 - EDITPATIENTDATA タグを使用する場合、ITEMSETINDEX タグと一緒に

DATA サブタグを使用してアイテムセットデータを変更することはできません。EDITPATIENTDATA タグには ITEMSETINDEX 属性を使用する必要があります。

詳細については、「[既存の臨床データの更新](#)」(357ページ)を参照してください。

- **繰り返しデータアイテムセットへのデータのインポート** — データベースには、繰り返しデータアイテムセットの行を追加できます。また、既存の繰り返しデータアイテムセット行にデータを追加したり、データを変更することも可能です。
- データベースに繰り返しデータアイテムセットの行を作成するには、PATIENTDATA タグを使用します。繰り返しデータアイテムセットの行は、1 度に 1 行、または複数まとめて作成できます。

注： PATIENTDATA タグを使用して繰り返しデータアイテムセットの行を作成するには、ITEMSETINDEX 属性と DATA サブタグを使用する必要があります。

詳細については、「[初診被験者の臨床データの追加](#)」(350ページ)を参照してください。

- 繰り返しデータアイテムセットのアイテムにデータを追加する、またはデータを変更するには、EDITPATIENTDATA タグを使用します。
 - 繰り返しデータアイテムセットのアイテムにデータを追加する、またはデータを変更するには、そのデータに前からデータが含まれていたかどうかにかかわらず、EDITPATIENTDATA タグを使用します。
 - EDITPATIENTDATA タグを使用する場合、ITEMSETINDEX タグと一緒に DATA サブタグを使用してアイテムセットデータを変更することはできません。EDITPATIENTDATA タグには ITEMSETINDEX 属性を使用する必要があります。

詳細については、「[既存の臨床データの更新](#)」(357ページ)を参照してください。

- **マップファイルのアイテムセット [!match!] フィールドは使用しないでください。** これは、[!cd!ItemsetIndex] フィールドを使用してアイテムセット行をマッチさせるのに使用します。[!match!] フィールドは通常、マップファイルに手動で追加されます。InForm データインポートユーティリティのフィールド定義ページには表示されません。
- **エントリ追加アイテムセットの削除と削除解除** — エントリ追加アイテムセットデータを削除または削除解除するには、MedML ファイルオプションを使用する必要があります。

注： 繰り返しデータのアイテムセットは削除することも、削除したアイテムセットを元に戻すこともできません。

XML ファイルのインポート

XML ファイルのインポートの概要

XML ファイルをインポートするには、以下の手順に従います。

- 1 追加または更新する症例データが含まれている XML ファイルを作成します。
詳細については、「**XML インポートファイルの作成**」(346ページ)を参照してください。
- 2 MedML ファイルオプションを使用して、インポートを実行します。
詳細については、「**MedML ファイルオプションを使用したインポートの実行**」(365ページ)を参照してください。

XML インポートファイルの作成

MedML ファイルオプションのインポートファイルは、インポート中に実行する処理のタイプと、インポートデータのインポート先および値を指定するエレメントが含まれた XML ファイルです。ファイルには、以下の実行タイプのタグを持たせることができます。

- 症例のスクリーニングと症例登録
- 新しい症例データの追加
- 既存の症例データの更新
- 1 つの施設から別の施設への症例の移動

インポートファイルの作成に使用するエディタは、プレーンテキストファイルを作成できる任意のテキストエディタです。詳細については、「**付録 A:インポート用 XML ファイルのサンプルデータ**」(423ページ)を参照してください。

XML サブミッションの作成

- 1 xml バージョン番号を含む最初の行を作成します。この文字列は、次のように小文字にする必要があります。

```
<?xml version="1.0"?>
```

- 2 どのタイプの処理を実行するかを InForm データインポートユーティリティに指示するため、開始エレメントと終了エレメントを追加します。

```
<CLINICALDATA>
</CLINICALDATA>
```

- 3 この開始エレメントと終了エレメントの間に、InForm データインポートユーティリティで実行するそれぞれのアクティビティに対し、別の開始エレメントと終了エレメントを追加します。データをインポートする症例ごとに 1 セットのエレメントを使用します。たとえば、新規症例のスクリーニングデータをインポートするには、次のエレメントを使用します。

```
<SCREEN>
```

</SCREEN>

- 4 インポートアクティビティの開始エレメントに、そのアクティビティタイプに必要な属性と、オプションの属性を任意に追加します。たとえば、**SCREEN** エレメントの場合、略名または名前で症例の施設を指定するには、**SITEMNEMONIC** 属性または **SITENAME** 属性が必要です。症例の施設の略名が PF である場合は、次のように **SITEMNEMONIC** 属性を挿入します。

```
<SCREEN SITEMNEMONIC="PF">
```

注： InForm データインポートユーティリティを使用して施設間で症例レコードを移動する場合は、本セクションの最後の手順に進んでください。症例レコード移動インポートファイルには、**DATA** エレメントは使用されません。

- 5 インポートアクティビティを指定する開始エレメントと終了エレメントの間に、次のように各フォームコントロール用の **DATA** エレメントを挿入します。

```
<DATA/>
```

- 6 **DATA** エレメントには、次に示す必須属性が含まれます。

- **TAG** — ターゲットのデータアイテムコントロールを特定するデータベースパスで、次のような順序からなる参照名で構成されます。

```
Section.Itemset.Item[.control[.control...]]
```

- **Section** — セクションの RefName です。セクション定義を含む XML ファイルで指定された通り。
- **Itemset** — アイテムセットの RefName です。アイテムセット定義を含む XML ファイルで指定された通り。ターゲットデータアイテムがアイテムセットではなく通常の CRF アイテムである場合は、0 を入力します。
- **Item** — アイテムの RefName です。アイテム定義を含む XML ファイルで指定された通り。アイテムセットの各アイテム用の **DATA** エレメントを、個別に作成します。
- **Control** — コントロールの RefName です。コントロール定義を含む XML ファイルで指定された通り。グループコントロールのエレメントにアクセスするには、子エレメントがネストされている各ペアレントコントロールを参照します。たとえば、グループコントロール内の 2 つのうち 1 つのテキストコントロールを扱うには、グループコントロールの RefName をタイプした後にテキストコントロールの RefName をタイプし、次のようにピリオドで名前を区切ります。

```
GroupControlRefName.TextControlRefName
```

- 次の 1 つ：

- **VALUE** — コントロールにインポートするデータの値です。二重引用符で値を囲みます。

<注> 二重引用符は、属性の値を区切る目的で用いられるため、値のテキストの一部として含めることはできません。値のテキストの一部として二重引用符を含める必要がある場合は、**XML** エンティティレファレンスである **"** を使用します。

- **CHILDSELECTED** — 子コントロールがコンパウンドコントロール内にネ

ストされている場合、選択された子コントロールの参照名です。例えば、ラジオコントロールに 2 つの ドロップダウンリスト が含まれている場合は、**CHILDSELECTED** 属性を使用して、選択するラジオコントロールを示します。

- **MONTH、DAY、YEAR、HOUR、MINUTE、SECOND** — 日時コントロールの各適用可能な部分の値です。
- **UNIT** — 単位定義がターゲットコントロールの一部である場合、選択した単位の参照名です。
- **COMMENT** — アイテムレベルコメントのテキストです。
- **REASONINCOMPLETE** — アイテムが不完全である理由です。この属性を指定する場合、DATA エlementに VALUE または任意の日時コントロール属性を含めないでください。
- **NOMULTIVALUE** — カンマ (,) を含むデータ値が単一値であることを示します。この属性がないと、カンマに先行するデータのみが値として格納されます。カンマ以降のデータは、multi-value DATA エlementでは別個の値とみなされます。

既存の症例データの場合：

- **CLEARVALUE** — コントロールの既存の値をクリアする場合は TRUE を指定します。

7 ファイルを保存します。

例

次の例は、DATA エlementを使って、スクリーニングフォームで PF 施設の症例 AAA のイニシャルを [症例のイニシャル] フィールドにインポートしたことを示します。

```
<SCREEN SITEMNEMONIC="PF">
  <DATA TAG="screen.0.patientinitials.patientinitials"
    VALUE="AAA"/>
</SCREEN>
```

次の例は、DATA エlementの CHILDSELECTED 属性を使って、RACE ラジオコントロールの RACEPULLDOWN ラジオボタンが選択されていることを示します。2 番目の DATA エlementは、ドロップダウンリスト内で選択した値を示しています。

```
<PATIENTDATA PATIENTINITIALS="AAA" SITEMNEMONIC="PF"
  FORMSETREFNAME="Visit1" FORMREFNAME="DEM" COMMENT>
  <DATA TAG="DEM.0.RACE.RACEGROUP"
    CHILDSELECTED="RACEPULLDOWN"/>
  <DATA TAG="DEM.0.RACE.RACEGROUP.RACEPULLDOWN"
    VALUE="Asian"/>
</PATIENTDATA>
```

次の例は、DATA エlementが症例登録フォームへインポートされる日付を特定したことを示します。

```
<ENROLL PATIENTINITIALS="AAA" SITEMNEMONIC="PF"
  PATIENTNUMBER="BK1" ENROLL="TRUE">
  <DATA TAG="consent.0.consentdate.date"
    MONTH="1" DAY="6" YEAR="1999"/>
</ENROLL>
```

次の例は、身長が測定される単位がインチであることを単位属性の使用で明確にします。

```
<PATIENTDATA PATIENTINITIALS="AAA" SITEMNEMONIC="PF"
  FORMSETREFNAME="Visit1" FORMREFNAME="DEM">
```

```
<DATA TAG="DEM.0.HEIGHT.HEIGHTTEXT"
VALUE="67" UNIT="Inches"/>
</PATIENTDATA>
```

次の例は、カンマ (,) を含むデータ値が 1 つの値であることを NOMULTIVALUE 属性の使用で明確にします。

```
<PATIENTDATA PATIENTINITIALS="AAA" SITEMNEMONIC="PF"
FORMSETREFNAME="Visit1" FORMREFNAME="DEM">
  <DATA TAG="DEM.0.HEIGHT.ID"
    VALUE="67,11" NOMULTIVALUE=""/>
</PATIENTDATA>
```

スクリーニング

施設で症例のスクリーニングデータをインポートするには、SCREEN エlementを次の必須属性と共に使用します。

属性	定義
SITEMNEMONIC	症例がスクリーニングされている施設の略名です。

例

次のサンプルファイルは、施設 PF において症例 AAA をスクリーニングしたことを示しています。

```
<?xml version="1.0"?>
<CLINICALDATA>
<!-- Screen Patient -->
<SCREEN SITEMNEMONIC="PF">
  <DATA TAG="screen.0.patientinitials.patientinitials"
    VALUE="AAA"/>
  <DATA TAG="screen.0.eligible.eligible" VALUE="yes"/>
  <DATA TAG="screen.0.datescreened.date" MONTH="1" DAY="6"
    YEAR="1999"/>
  <DATA TAG="screen.0.dob.dob" MONTH="11" DAY="11"
    YEAR="1959"/>
</SCREEN>
</CLINICALDATA>
```

症例登録

施設で症例の登録データをインポートするには、ENROLL エlementを次の必須属性と共に使用します。

属性	定義
PATIENTINITIALS	登録されている症例のイニシャルです。
SITEMNEMONIC	症例がスクリーニングされている施設の略名です。
DUPLICATEORDER	同一の施設に登録され、同一のイニシャルをもつ症例がスクリーニングを受ける順序を指定する数値です。
PATIENTNUMBER	登録されている症例の症例番号です。
ENROLL	TRUE または FALSE を指定して、症例を登録するかどうかを指示します。

例 1

次のサンプルファイルは、施設 PF で症例 XYZ を登録するのに使用するタグを例示したものです。この例では、症例 XYZ が以前スクリーニングを受けたことがあるものと仮定しています。

```
<?xml version="1.0"?>
<CLINICALDATA>

<!-- Screen Patient -->
<SCREEN SITEMNEMONIC="PF">
<DATA TAG="screen.0.patientinitials.patientinitials" VALUE="XYZ"/>
<DATA TAG="screen.0.eligible.eligible" VALUE="yes"/>
<DATA TAG="screen.0.datescreened.date" MONTH="1" DAY="6" YEAR="1999"/>
<DATA TAG="screen.0.dob.dob" MONTH="11" DAY="11" YEAR="1959"/>
</SCREEN>

<!-- Enroll Patient -->
<ENROLL PATIENTINITIALS="XYZ" SITEMNEMONIC="PF" PATIENTNUMBER="BK-XYZ"
ENROLL="TRUE">
<DATA TAG="consent.0.consentdate.date" MONTH="1" DAY="6" YEAR="1999"/>
<DATA TAG="consent.0.patientnumber.patientnumber" VALUE="28"/>
<DATA TAG="inclusion.0.age_inc.yesno" VALUE="1"/>
<DATA TAG="inclusion.0.hyper_inc.yesno" VALUE="1"/>
<DATA TAG="inclusion.0.understand_inc.yesno" VALUE="1"/>
<DATA TAG="inclusion.0.agree_inc.yesno" VALUE="1"/>
<DATA TAG="exclusion.0.secondary_ex.yesno" VALUE="0"/>
<DATA TAG="exclusion.0.malignant_ex.yesno" VALUE="0"/>
<DATA TAG="exclusion.0.allergyhistory_ex.yesno" VALUE="0"/>
<DATA TAG="exclusion.0.myocardial_ex.yesno" VALUE="0"/>
<DATA TAG="exclusion.0.monitor_ex.yesno" VALUE="0"/>
</ENROLL>
</CLINICALDATA>
```

例 2

この例は、イニシャルが同一の症例を登録する際に、その順序の指定に **DUPLICATEORDER** 属性を使用するものです。症例ジョン・R・ドウおよびジェーン・R・ドウ（ともにイニシャルが JRD）は施設 PF でスクリーニングされます。ここでは最初にジョン、その後ジェーンの順となります。

```
<!-- This is John R. Doe-->
<ENROLL PATIENTINITIALS="JRD" SITEMNEMONIC="PF" DUPLICATEORDER="1"
PATIENTNUMBER="BK-JRD" ENROLL="TRUE">
</ENROLL>
<!-- This is Jane R. Doe-->
<ENROLL PATIENTINITIALS="JRD" SITEMNEMONIC="PF" DUPLICATEORDER="2"
PATIENTNUMBER="BK-JRD" ENROLL="TRUE">
</ENROLL>
```

新しい臨床データの追加

施設で既存の症例に対して新しい臨床データを追加するには、**PATIENTDATA** エレメントを次の属性と共に使用します。

属性	定義
PATIENTINITIALS	登録されている症例のイニシャルです。 PATIENTINITIALS 、 PATIENTNUMBER のいずれかが必要です。

属性	定義
PATIENTNUMBER	登録されている症例の症例番号です。 PATIENTINITIALS 、 PATIENTNUMBER のいずれかが必要です。
SITEMNEMONIC	症例がスクリーニングされている施設の略名です。 SITEMNEMONIC または SITENAME が必要です。
SITENAME	症例が登録されている施設の名前です。 SITEMNEMONIC または SITENAME が必要です。
FORMSETREFNAME	データをインポートするビジットの参照名です。
FORMSETINDEX	データを追加するビジットインスタンスを指定します。
FORMREFNAME	データをインポートする CRF の参照名を指定します。
REASONINCOMPLETE	フォームまたはアイテムレベルのいずれかに適用できます。値は、ラジオグループコントロールの値の 1 つです。フォームあるいはアイテムが完全な場合、この属性は無視されます。 フォームレベルで不完全な理由を追加するには、 PATIENTDATA グループに DATA エレメントを含めないでください。
FORMINDEX	(オプション) 繰り返しフォームインスタンスが存在しない場合、新規に作成されます (FORMREFNAME が繰り返しフォームの場合)。存在する場合、値は新規データが追加されるフォームインスタンスを示します。
ASSOCIATION	関連が存在することを示します。

属性	定義
ITEMSETINDEX	<p>アイテムセットを追加する行の行番号。</p> <ul style="list-style-type: none"> エントリ追加アイテムセットの場合: <ul style="list-style-type: none"> 1 度に 1 つの行番号を指定します。 エントリ追加アイテムセットに単一の行を追加します。 ITEMSETINDEX 属性は、PATIENTDATA タグまたは DATA サブタグと使用できます。 繰り返しデータアイテムセットの場合: <ul style="list-style-type: none"> 1 度に 1 行または複数の行を指定します。 繰り返しデータアイテムセットのすべての行を作成します。 繰り返しデータアイテムセットの場合、ITEMSETINDEX 属性を DATA サブタグと使用する必要があります。 <p>備考：</p> <p>PATIENTDATA タグと DATA タグの両方で ITEMSETINDEX 属性を指定することはできません。</p> <p>ITEMSETINDEX 属性は、PATIENTDATA タグと DATA サブタグのオプションの属性です。</p>

エントリ追加アイテムセットへの単一行の追加

フォーム上のエントリ追加アイテムセットに行を追加するには、次の追加属性を使用します。

属性	定義
SECTIONNAME	アイテムセットが存在するセクションの参照名を指定します。
ITEMSETNAME	アイテムセット定義の参照名を指定します。

属性	定義
ITEMSETINDEX	<p>更新するエントリの追加アイテムセット行の数です。ITEMSETINDEX 属性が空欄か、値が 0 の場合、InForm データインポートユーティリティは新規アイテムセット行を追加します。ITEMSETINDEX が 0 以外の値の場合、InForm データインポートユーティリティは指定した行を更新します。注) 既存のアイテムセットデータが変更されませんが、欠測値がマップされます。</p> <p>注： エントリ追加または繰り返しデータアイテムセットの行にデータを追加する、またはこれらのデータを変更する場合は、EDITPATIENTDATA タグを使用する必要があります。これは行にデータが含まれているかどうかとは関係ありません。</p>

繰り返しデータアイテムセットの行の作成

フォーム上に繰り返しデータアイテムセットの行を作成するには、以下の属性を使用します。

属性	定義
PATIENTDATA 属性	定義
SECTIONNAME	アイテムセットが存在するセクションの参照名を指定します。
ITEMSETNAME	アイテムセット定義の参照名を指定します。
DATA 属性	定義
ITEMSETINDEX	<p>繰り返しデータアイテムセットに対して行を作成するアイテムセットの行番号を 1 つまたは複数指定します。</p> <p>有効な値は 1～n です。ここで、n は繰り返しデータアイテムセットの INITIALROWCOUNT です。</p> <p>注： エントリ追加または繰り返しデータアイテムセットの行にデータを追加する、またはこれらのデータを変更する場合は、EDITPATIENTDATA タグを使用する必要があります。これは行にデータが含まれているかどうかとは関係ありません。</p>

予定外のビジットへのデータ追加

予定外のビジットにデータを追加するには、次の追加属性を使用します。

属性	定義
NEWUNSCHEDVISIT	データが新規の予定外のビジットに追加されているかどうかを示します。値は、TRUE か FALSE (デフォルト) です。この属性は、ビジット日フォーム (DOV の FORMREFNAME をもつ事前に定義されたフォーム) で使用します。詳細については、 例 3: 予定外のビジットへのデータ追加 (355ページ) を参照してください。
FORMSETINDEX	データを追加する予定外のビジットを指定する数値です。この数値は、予定外ビジットがスタディに追加された順序に対応します。

コメントの追加

COMMENT 属性を使用して、アイテム、アイテムセット内のアイテム、フォームのいずれかにコメントを追加できます。

- アイテムにコメントを追加するには、**DATA** 属性に、コメントのテキストとともに **COMMENT** 属性を含めます。
- フォームにコメントを追加するには、属性 **SECTIONNAME** および **ITEMSETNAME** が含まれていない **PATIENTDATA** 属性に、コメントのテキストとともに **COMMENT** 属性を含めます。

例: フォームへのデータ追加

以下は XML ファイルの一部で、施設 PF の症例 AAA 用の DEM フォームにデータを追加するために使用されるエレメントを示したものです。

```
<?xml version="1.0"?>
<CLINICALDATA>
<PATIENTDATA
  PATIENTINITIALS="VOL" SITEMNEMONIC="PF"
  FORMSETREFNAME="visit1"
  FORMREFNAME="DEM" COMMENT="This form was edited by Joe">
  <DATA TAG="DEM.0.GENDER.GENDERRADIO" VALUE="1"/>
  <DATA TAG="DEM.0.DEMDOB.dob" MONTH="2" DAY="14"
    YEAR="1961" COMMENT="As reported by patient"/>
  <DATA TAG="DEM.0.RACE.RACEGROUP"
    CHILDSELECTED="RACETEXT"/>
  <DATA TAG="DEM.0.RACE.RACEGROUP.RACETEXT"
    VALUE="African American"/>
  <DATA TAG="DEM.0.HEIGHT.HEIGHTTEXT" VALUE="67"
    UNIT="Inches"/>
  <DATA TAG="DEM.0.WEIGHT.WEIGHTTEXT" VALUE="150"
    UNIT="Pound"/>
  <DATA TAG="DEM.0.FRAME.FRAMEPULLDOWN" VALUE="1"/>
  <DATA TAG="SH.0.SMOKE.SMOKERADIO" VALUE="Y"/>
  <DATA TAG="SH.0.EVERSMOKED.SMOKERADIO" VALUE="N"/>
  <DATA TAG="SH.0.WHATSMOKED.SMOKECHECKBOX"
    VALUE="cigarette,pipe"/>
  <DATA TAG="SH.0.HOWMUCHSMOKED.SMOKERADIO2"
    CHILDSELECTED="NUMTEXT"/>
  <DATA TAG="SH.0.HOWMUCHSMOKED.SMOKERADIO2.NUMTEXT"
```

```

        VALUE="10"/>
    <DATA TAG="SH.0.YRSSMOKED.SMOKERADIO2"
        VALUE="NDElement"/>
</PATIENTDATA>
</CLINICALDATA>

```

例: エントリ追加アイテムセットへの単一行の追加

以下は XML ファイルの一部で、高血圧既往歴 (Hypertension History) フォーム上にエン
トリ追加アイテムセットの新しい行を作成するエレメントを示したものです。

```

<PATIENTDATA
  PATIENTINITIALS="AAA" SITEMNEMONIC="PF"
  FORMSETREFNAME="Visit1" FORMREFNAME="HH"
  SECTIONNAME="PT" ITEMSETNAME="PT">
  <DATA TAG="PT.PT.THERAPYTEXT.THERAPYTEXT"
    VALUE="aspirin"/>
  <DATA TAG="PT.PT.DOSAGETEXT.DOSAGETEXT"
    VALUE="1 tablet"/>
</PATIENTDATA>

```

以下の XML フラグメントは、高血圧既往歴 (Hypertension History) フォームで既存の
エントリの追加アイテムセットへのアイテムの入力を終了するために使用されるエレメ
ントを示したものです。

```

<PATIENTDATA
  PATIENTINITIALS="AAA" SITEMNEMONIC="PF"
  FORMSETREFNAME="Visit1" FORMREFNAME="HH"
  SECTIONNAME="PT" ITEMSETNAME="PT" ITEMSETINDEX="1">
  <DATA TAG="PT.PT.DOSDATE.DOSDATE" MONTH="12" DAY="23"
    YEAR="1998"/>
  <DATA TAG="PT.PT.DISCULLDOWN.DISCULLDOWN"
    VALUE="Not Effective"/>
</PATIENTDATA>

```

例: 繰り返しデータアイテムセットの行の作成

以下は XML ファイルの一部で、投与 VS2 フォーム上にサンプル収集アイテムセットの
行を作成するエレメントを示したものです。

繰り返しデータアイテムセットには、値が事前に定義された 2 つの行が含まれています。

- Hour 10, Minute 0
- Hour 10, Minute 2

Hour 10, Minute 0 データポイントには、値 50.000 が入力されます。

```

<PATIENTDATA PATIENTINITIALS='RDF' SITEMNEMONIC='01' FORMSETREFNAME='CommonCRF'
  FORMREFNAME='frmDOSEVS2' FORMINDEX='0'
  SECTIONNAME='sctSampleCollection' ITEMSETNAME='sctSampleCollection'>
  <DATA TAG='sctSampleCollection.sctSampleCollection.itmTime.itmTime'
    ITEMSETINDEX='1' HOUR='10' MINUTE='0' />
  <DATA
    TAG='sctSampleCollection.sctSampleCollection.itmMeasurement.itmMeasurement'
    ITEMSETINDEX='1' VALUE='50.00' />
  <DATA TAG='sctSampleCollection.sctSampleCollection.itmTime.itmTime'
    ITEMSETINDEX='2' HOUR='10' MINUTE='2' />
</PATIENTDATA>

```

例: 予定外のビジットへのデータ追加

以下は XML ファイルの一部で、初回および 2 回目の予定外のビジットに含まれる
DOV フォームおよびバイタルサインフォームにデータを追加するために使用される
エレメントを示したものです。

```

<?xml version="1.0"?>
<CLINICALDATA>

```

```

<!-- DOV form -->
<PATIENTDATA
  PATIENTINITIALS="ABC" SITEMNEMONIC="PF"
  FORMSETREFNAME="UnschVisit" FORMREFNAME="DOV"
  NEWUNSCHEDVISIT="TRUE">
  <DATA TAG="DOV.0.DOV.DOV" MONTH="2" DAY="1"
    YEAR="1999"/>
</PATIENTDATA>
<!-- VitalSigns form -->
<PATIENTDATA
  PATIENTINITIALS="ABC" SITEMNEMONIC="PF" FORMSETINDEX="1"
  FORMSETREFNAME="UnschVisit" FORMREFNAME="VS">
  <DATA TAG="VS.0.DATEASSESS.COMMONDATE" MONTH="3" DAY="1"
    YEAR="1999"/>
  <DATA TAG="VS.0.WEIGHT.PFWT_TC" VALUE="150"
    UNIT="Pound"/>
  <DATA TAG="VS.0.TEMPTEXT.TEMPTEXT" VALUE="98.7"
    UNIT="Fahrenheit"/>
  <DATA TAG="VS.0.BPREADING.BPREADINGGROUP.SYSTEXT"
    VALUE="130"/>
  <DATA TAG="VS.0.BPREADING.BPREADINGGROUP.DIASTEXT"
    VALUE="85"/>
</PATIENTDATA>
<!-- DOV form -->
<PATIENTDATA
  PATIENTINITIALS="ABC" SITEMNEMONIC="PF"
  FORMSETREFNAME="UnschVisit" FORMREFNAME="DOV"
  NEWUNSCHEDVISIT="TRUE">
  <DATA TAG="DOV.0.DOV.DOV" MONTH="2" DAY="2"
    YEAR="1999"/>
</PATIENTDATA>
<!-- VitalSigns form -->
<PATIENTDATA
  PATIENTINITIALS="ABC" SITEMNEMONIC="PF" FORMSETINDEX="2"
  FORMSETREFNAME="UnschVisit" FORMREFNAME="VS">
  <DATA TAG="VS.0.DATEASSESS.COMMONDATE" MONTH="3" DAY="2"
    YEAR="1999"/>
  <DATA TAG="VS.0.WEIGHT.PFWT_TC" VALUE="150"
    UNIT="Pound"/>
  <DATA TAG="VS.0.TEMPTEXT.TEMPTEXT" VALUE="98.7"
    UNIT="Fahrenheit"/>
  <DATA TAG="VS.0.BPREADING.BPREADINGGROUP.SYSTEXT"
    VALUE="130"/>
  <DATA TAG="VS.0.BPREADING.BPREADINGGROUP.DIASTEXT"
    VALUE="85"/>
</PATIENTDATA>

```

例:REASONINCOMPLETE の指定

以下は XML ファイルの一部で、[バイタルサイン] (VS) フォームで脈拍アイテムにデータを追加するために使用されるタグを示したものです。

```

<PATIENTDATA
  PATIENTINITIALS="A3" SITEMNEMONIC="PF"
  FORMSETREFNAME="DV1" FORMREFNAME="VS"
  REASONOTHER="test reason">
  <DATA TAG="VS.0.PULSERHYTHM.PULSERHYTHMRADIO"
    COMMENT="irregular REASONINCOMPLETE="NAElement"/>
</PATIENTDATA>

```

例:関連項目の新規作成

以下は XML ファイルの一部で、関連項目の新規作成に使用されるタグを示したものです。

```

<PATIENTDATA
  PATIENTINITIALS="pjb" SITEMNEMONIC="PF"
  FORMSETREFNAME="Visit6" FORMREFNAME="VS"
  FORMINDEX="2"
  REASONPULLDOWN="3"
  <ASSOCIATION FORMSETREFNAME="Visit1" FORMREFNAME="DEM"
    FORMINDEX="4"/>

```

</PATIENTDATA>

既存の臨床データの更新

施設で症例の既存の臨床データを修正するには、EDITPATIENTDATA エlementを次の属性と共に使用します。

属性	定義
PATIENTINITIALS	登録されている症例のイニシャルです。 PATIENTINITIALS 、 PATIENTNUMBER のいずれかが必要です。
PATIENTNUMBER	登録されている症例の症例番号です。 PATIENTINITIALS 、 PATIENTNUMBER のいずれかが必要です。
SITEMNEMONIC	症例がスクリーニングされている施設の略名です。 SITEMNEMONIC 、 SITENAME のいずれかが必要です。
SITENAME	症例が登録されている施設の名前です。 SITEMNEMONIC 、 SITENAME のいずれかが必要です。
FORMSETREFNAME	データをインポートするビジットの参照名です。
FORMREFNAME	データをインポートする CRF の参照名を指定します。
REASONPULLDOWN	データ値フォームの [変更理由] ドロップダウンリストにリストされる、変更理由の事前定義の値を指定します。
REASONOTHER	データ値フォームの [変更のための別の理由] フィールドに入力される、変更理由のテキストを指定します。
REASONINCOMPLETE	フォームまたはアイテムレベルのいずれかに適用できます。値は、ラジオグループコントロールの値の 1 つです。フォームまたはアイテムが完全な場合、この属性は無視されます。
CLEARCRF	TRUE の場合、以降に続くすべての <Data Tags>を無視します。
FORMINDEX	繰り返しフォームの必須属性です。 更新する繰り返しフォームを指定します。値は、更新するフォーム項目に対応させます。
ASSOCIATION	関連が存在することを示します。
ACTION	ADD または REMOVE。関連を追加するか削除するかを指定します。

属性	定義
ITEMSETINDEX	データを変更または追加するエン트리追加または繰り返しデータアイテムセットのアイテムの行番号。 注： ITEMSETINDEX を EDITPATIENTDATA タグの DATA サブタグと使用することはできません。

エン트리追加アイテムセットのデータの編集

フォームのエン트리追加アイテムセットのデータを編集するには、次の追加属性を使用します。

属性	定義
SECTIONNAME	アイテムセットが存在するセクションの参照名を指定します。
ITEMSETNAME	アイテムセット定義の参照名を指定します。
ITEMSETINDEX	更新するエントリの追加アイテムセット行の数です。ITEMSETINDEX 属性が空欄か、値が 0 の場合、InForm データインポートユーティリティは新規アイテムセット行を追加します。ITEMSETINDEX が 0 以外の値の場合、InForm データインポートユーティリティは指定した行を更新します。注) 既存のアイテムセットデータが変更されませんが、欠測値がマップされます。 注： エン트리追加または繰り返しデータアイテムセットの行にデータを追加する、またはこれらのデータを変更する場合は、EDITPATIENTDATA タグを使用する必要があります。これは行にデータが含まれているかどうかとは関係ありません。

繰り返しデータアイテムセットのデータの編集

フォームの繰り返しデータアイテムセットのデータを編集するには、次の追加属性を使用します。

属性	定義
SECTIONNAME	アイテムセットが存在するセクションの参照名を指定します。
ITEMSETNAME	アイテムセット定義の参照名を指定します。

属性	定義
ITEMSETINDEX	<p>繰り返しデータアイテムセットのアイテムのデータを追加または変更します。</p> <p>データを変更または追加する繰り返しデータアイテムセットのアイテムの行番号を指定します。</p> <p>有効な値は 1～n です。ここで、n は繰り返しデータアイテムセットの <code>INITIALROWCOUNT</code> です。</p> <p>注： エントリ追加または繰り返しデータアイテムセットの行にデータを追加する、またはこれらのデータを変更する場合は、<code>EDITPATIENTDATA</code> タグを使用する必要があります。これは行にデータが含まれているかどうかとは関係ありません。</p>

コメントの編集

COMMENT 属性を使用して、アイテム、アイテムセット内のアイテム、またはフォームのコメントを編集できます。

- アイテムのコメントを編集するには、**DATA** エlementにコメントのテキストと共に **COMMENT** 属性を含めます。
- アイテムセットのコメントを編集するには、**SECTIONNAME** および **ITEMSETNAME** 属性 (**EDITPATIENTDATA** Elementがアイテムセットを更新することを示す属性) を含む **EDITPATIENTDATA** Elementに、コメントのテキストと共に **COMMENT** 属性を含めます。
- フォームのコメントを編集するには、**SECTIONNAME** および **ITEMSETNAME** 属性を含まない **EDITPATIENTDATA** Elementに、コメントのテキストと共に **COMMENT** 属性を含めます。

例:データ値の変更

次のサンプルファイルの例は、症例背景 (DEM) フォームで身長アイテムの値を変更するために使用されるタグを示したものです。変更理由は、データ値フォームの事前定義された変更理由以外の文字列です。

注： **UNIT** の値は修正しなくても入力する必要があり、入力しない場合は削除されます。

```
<?xml version="1.0"?>
<CLINICALDATA>
<!-- Demographics form -->
<EDITPATIENTDATA
  PATIENTINITIALS="XYZ"
  ITEMNEMONIC="PF"
  FORMSETREFNAME="Visit1"
  FORMREFNAME="DEM"
  REASONOTHER="received new data">
  <DATA TAG="DEM.0.HEIGHT.PFHT_TC" VALUE="75"
    UNIT="Inches" />
</EDITPATIENTDATA>
</CLINICALDATA>
```

例:データ値の消去

以下の XML フラグメントは、**DATA** エレメントで **CLEARVALUE** 属性を使用した場合を示したものです。複合ラジオコントロールの選択内容を変更する場合、元のチャイルドコントロール値をクリアする必要があります。

```
<EDITPATIENTDATA
  PATIENTINITIALS="PE1" SITEMNEMONIC="PF"
  FORMSETREFNAME="Visit1" FORMREFNAME="DEM"
  REASONPULLDOWN="New Information">
  <DATA TAG="RACEGROUP.RACETEXT" CLEARVALUE="TRUE"/>
</EDITPATIENTDATA>
```

例:エントリ追加アイテムセット行の変更

以下の XML フラグメントは、有害事象 (AE) フォーム上のエントリ追加アイテムセットを変更するものです。

```
<EDITPATIENTDATA
  PATIENTINITIALS="EDT" SITEMNEMONIC="PF"
  FORMSETREFNAME="CommonCRF" FORMREFNAME="AE"
  SECTIONNAME="AE" ITEMSETNAME="AE"
  ITEMSETINDEX="5" REASONOTHER="additional description">
  <DATA TAG="AE.AE.AEDESC.AEDESCTEXT"
    VALUE="Temp spike"/>
</EDITPATIENTDATA>
```

例:繰り返しデータアイテムセットの変更

以下の XML フラグメントは、投与 VS2 フォーム上の繰り返しデータアイテムセットを変更するものです。

```
<EDITPATIENTDATA PATIENTINITIALS='RDF' SITEMNEMONIC='01'
  FORMSETREFNAME='CommonCRF' FORMREFNAME='frmDOSEVS2' FORMINDEX='1'
  SECTIONNAME='sctSampleCollection' ITEMSETNAME='sctSampleCollection'
  ITEMREFNAME='itmTime' ITEMSETINDEX='1' REASONPULLDOWN='1'>
  <DATA TAG='sctSampleCollection.sctSampleCollection.itmTime.itmTime'
    HOUR='11' MINUTE='11' />
</EDITPATIENTDATA>
<EDITPATIENTDATA PATIENTINITIALS='RDF' SITEMNEMONIC='01'
  FORMSETREFNAME='CommonCRF' FORMREFNAME='frmDOSEVS2' FORMINDEX='1'
  SECTIONNAME='sctSampleCollection' ITEMSETNAME='sctSampleCollection'
  ITEMREFNAME='itmTime' ITEMSETINDEX='2' REASONPULLDOWN='1'>
  <DATA
    TAG='sctSampleCollection.sctSampleCollection.itmTime.itmTime' HOUR='11'
    MINUTE='13' />
</EDITPATIENTDATA>
```

例:REASONINCOMPLETE の指定

以下の XML フラグメントは、バイタルサイン (VS) フォームの脈拍アイテムへの修正を示しています。

```
<EDITPATIENTDATA
  PATIENTINITIALS="A3" SITEMNEMONIC="PF"
  FORMSETREFNAME="DV1" FORMREFNAME="VS"
  REASONOTHER="test reason">
  <DATA TAG="VS.0.PULSERHYTHM.PULSERHYTHMRADIO"
    COMMENT="me too REASONINCOMPLETE="NAElement"/>
</EDITPATIENTDATA>
```

例:CRF の消去

以下の XML フラグメントは、CRF をクリアするために使用するタグを示したものです。

```
<EDITPATIENTDATA
  PATIENTINITIALS="DD" SITEMNEMONIC="PF"
  FORMSETREFNAME="VISIT7" FORMREFNAME="PREIM"
```



```

    REASONOTHER="clear CFR test">
    CLEARCRF="TRUE"
    <DATA TAG="DEM.0.DTV.DTV_DC" MONTH="4" DAY="22"
        YEAR="2000"
    </EDITPATIENTDATA>

```

例:関連項目の削除

以下の XML フラグメントは、関連項目の削除に使用するタグを示したものです。

```

<EDITPATIENTDATA
    PATIENTINITIALS="DD" SITEMNEMONIC="PF"
    FORMSETREFNAME="VISIT7" FORMREFNAME="PREIM"
    FORMINDEX="2"
    REASONPULLDOWN="Transcription Error">
    <ASSOCIATION FORMSETREFNAME="Visit6" FORMREFNAME="PE" FORMINDEX="3"
    ACTION="REMOVE"
</EDITPATIENTDATA>

```

例:繰り返しフォームの削除

以下の XML フラグメントは、繰り返しフォームの削除に使用するタグを示したものです。

```

<?xml version="1.0"?>
<CLINICALDATA xmlns="PhaseForward/ImportXML/Inform4">
<EDITPATIENTDATA
    PATIENTINITIALS="XYZ"
    SITEMNEMONIC="PF"
    FORMSETREFNAME="vstAECM"
    FORMREFNAME=" LAE1 "
    FORMINDEX="1"
    <!-- Form Status action:
        DELETE to delete the form
        UNDELETE to restore a deleted form
    -->
    FORMSTATUS="DELETE"
    REASONOTHER="Deleting crf" />
</CLINICALDATA>

```

例:繰り返しフォームの復元

以下の XML フラグメントは、繰り返しフォームの復元に使用するタグを示したものです。

```

<?xml version="1.0"?>
<CLINICALDATA xmlns="PhaseForward/ImportXML/Inform4">
<EDITPATIENTDATA
    PATIENTINITIALS="XYZ"
    SITEMNEMONIC="PF"
    FORMSETREFNAME=" vstAECM "
    FORMREFNAME=" LAE1"
    FORMINDEX="1"
    <!-- Form Status action:
        DELETE to delete the form
        UNDELETE to restore a deleted form
    -->
    FORMSTATUS="UNDELETE"
    REASONOTHER="Undeleting CRF" />
</CLINICALDATA>

```

症例レコードの移動

InForm ユーザインターフェースを使用して、1 回に 1 人分ずつ症例情報を移動できます。また、InForm データインポートユーティリティを使用して、一括移動することもできます。

症例情報が移動できるのは次のような場合です。

- スタディが終了する前に症例の住所に変更があった場合
- スタディ期間を通じて症例が複数の住居に居住していた場合
- 最初の段階で、症例が誤った施設やスタディにはもう携わっていない担当医に割り当てられた場合

症例情報を個別に移動する場合でも、一括移動する場合でも、次の点に留意する必要があります。

- InForm アプリケーションでは、移動先施設のスタディバージョンが現在の施設のスタディバージョンと同一またはそれ以降の場合にのみ、症例情報を移動できます。

施設のスタディバージョンは、InForm ユーザインターフェイス の管理インタフェースで確認できます。

登録が完全な症例のみ移動できます。スクリーニングを受けていても症例登録していないか、または症例登録に不備がある症例の情報は移動できません。

症例レコードを施設間で移動するには、次のエレメントを使用します。

- PATIENTSITECHANGE
- NEWSITE
- CURRENTSITE

PATIENTSITECHANGE

PATIENTSITECHANGE は、移動する症例をそれぞれ特定するためのエレメントです。移動する症例ごとに、**PATIENTSITECHANGE** の開始エレメントと終了エレメントのペアを 1 つ使用します。**PATIENTSITECHANGE** エレメントで、1 人の症例、現在の施設、および移動先施設に関するすべての情報を囲みます。

PATIENTSITECHANGE エレメントの必須属性は、次の 1 つのみです。

属性	定義
REASON	症例を移動する理由を記述したテキストです。

PATIENTSITECHANGE エレメント内には、次のエレメントを含めます。

- NEWSITE
- CURRENTSITE

NEWSITE

NEWSITE エレメントは、症例の移動先施設に関する情報を提供するためのものです。**NEWSITE** エレメントは、**PATIENTSITECHANGE** の開始エレメントと終了エレメントの間で使用します。

NEWSITE エレメントには、移動先施設を指定する属性が 1 つ必要です。次の属性のいずれかを使用できます。

属性	定義
SITEMNEMONIC	転送先施設の略名
SITENAME	転送先施設の施設名

NEWSITE エレメントには、オプションの属性が 1 つあります。

属性	定義
PATIENTNUMBER	移動先施設での症例の症例番号。この属性は、移動先の施設で症例番号の重複が発生するために、番号を現在の施設のものから変更する必要がある場合にのみ使用します。

CURRENTSITE

CURRENTSITE エレメントは、症例の現在の、または前の施設に関する情報を提供するためのものです。**CURRENTSITE** エレメントは、**PATIENTSITECHANGE** の開始エレメントと終了エレメントの間で使用します。

CURRENTSITE エレメントには、現在の施設を指定するための属性と、移動する症例を指定するための属性の 2 つの必須属性があります。

次の属性のいずれかを使用して、現在の施設を特定します。

属性	定義
SITEMNEMONIC	転送先施設の略名
SITENAME	転送先施設の施設名

次の属性のいずれかを使用して、移動する症例を特定します。

属性	定義
PATIENTINITIALS	移動する症例のイニシャル。現在の施設に、指定したものとイニシャルの同じ症例が他にもいることがわかっている場合は、この属性を使用する際、オプションの DUPLICATEORDER 属性を適用して、移動する症例を指定します。

属性	定義
PATIENTNUMBER	現在の施設に存在する、移動する症例の症例番号

PATIENTINITIALS エレメントで指定したイニシャルが当該施設で重複する場合は、**CURRENTSITE** エレメントと共に次の属性を適用します。

属性	定義
DUPLICATEORDER	現在の施設で症例のイニシャルが重複している場合に、移動する症例を指定します。イニシャルが同一の症例が複数いる場合、InForm アプリケーションはこれらの症例のスクリーニング番号をチェックし、 DUPLICATEORDER タグで指定した順番に合致している症例を移動します。 この属性は、 PATIENTINITIALS を使用して症例を指定する場合にのみ適用します。

例

下のサンプルは、複数の症例を移動するのに使用される、MedML ファイルのエレメントを示したものです。各ペアの **PATIENTSITECHANGE** 属性は、それぞれ 1 人の症例の現在の施設および移動先施設の情報を定義していることに注意してください。

```
<?xml version="1.0"?>
<CLINICALDATA>
  <PATIENTSITECHANGE REASON="new patient address">
    <NEWSITE SITEMNEMONIC="MCLEAN"/>
    <CURRENTSITE SITEMNEMONIC="PF" PATIENTNUMBER="1003"/>
  </PATIENTSITECHANGE>
<!--For subject1001 the file specifies a new subject number because a subject already exists at the
McLean Hospital site with the same subject number.>

  <PATIENTSITECHANGE REASON="new patient address">
    <NEWSITE SITENAME="McLean Hospital" PATIENTNUMBER="1002"/>
    <CURRENTSITE SITEMNEMONIC="PF" PATIENTNUMBER="1001"/>
  </PATIENTSITECHANGE>
<!--The DUPLICATEORDER attribute indicates that subject DDD is the second subject with
those initials to be screened at the McLean Hospital site.>

  <PATIENTSITECHANGE REASON="new patient address">
    <NEWSITE SITEMNEMONIC="MCLEAN"/>
    <CURRENTSITE SITENAME="McLean Hospital" PATIENTINITIALS="DDD"
      DUPLICATEORDER="2"/>
  </PATIENTSITECHANGE>
</CLINICALDATA>
```

その他の InForm データインポートユーティリティ属性

属性	説明	値
CLEARCRF	指定したフォームをクリアするかどうかを示します。	<ul style="list-style-type: none"> TRUE FALSE

属性	説明	値
COMMONFORM	フォームが共通 CRF であるかどうかを示します。	<ul style="list-style-type: none"> • TRUE • FALSE
DELETEITEMSET	エントリ追加アイテムセットを削除するかどうかを示します。	<ul style="list-style-type: none"> • TRUE • FALSE
FORMSTATUS	フォームのステータスを変更します。	<ul style="list-style-type: none"> • FREEZE • UNFREEZE • LOCK • UNLOCK • DELETE • UNDELETE
OVERRIDE	症例スクリーニングまたは登録が条件付かどうかを示します。	<ul style="list-style-type: none"> • TRUE • FALSE
UNDELETEITEMSET	削除されたエントリ追加アイテムセットの削除を解除するかどうかを示します。	<ul style="list-style-type: none"> • TRUE • FALSE

MedML ファイルオプションを使用したインポートの実行

注： データベースにデータをインポートするには、**InForm** データインポートユーティリティを起動する前に **InForm** サーバが稼働している必要があります。

XML ファイルをデータベースにインポートし、InForm アプリケーションサーバを介してそれをサブミットするには、以下の手順に従います。

- 1 [スタート] > [すべてのプログラム] > [Oracle Health Sciences] > [InForm 6.0] > [InForm データインポート] の順にクリックします。
InForm データインポートのメインウィンドウが表示されます。
- 2 [MedML ファイル] をクリックします。
- 3 [トライアル名] フィールドに、ファイルをインポートするスタディの名前を入力します。ドロップダウンリストには、最近アクセスした 10 件のスタディが表示されます。
- 4 [次へ] をクリックします。
- 5 [MedML ファイルの選択] フィールドに、インポートする XML ファイルのフルパス名を入力するか、[参照] をクリックしてファイルの場所を特定します。以下のオプションが有効です。
 - 1 つの XML ファイルの名前
 - スペースで区切った、複数の XML ファイルの名前
 - 複数の XML ファイルを含むレスポンスファイルの名前。次の形式で、1 行に

1 ファイルずつ適用します。

@filename

- 6 必要に応じて、データベースにデータを反映せずに、MedML ファイルの構文解析のみを行う場合は、[解析のみ] を選択します。この機能は、実際に MedML ファイルをデータベースにインポートする前に、構文をテストするのに使用します。

- 7 [次へ] をクリックします。

ダイアログボックスが表示され、InForm のユーザ名とパスワードを入力するよう指示されます。

注： [解析のみ] を選択した場合、ユーザ名とパスワードを指定する必要はありません。

- 8 [名前] フィールドに、インポートするデータに関して適切な権限を有する InForm ユーザの名前を入力します。

インポートの目的	ユーザに必要な権限
症例の臨床データを追加	CRF へのデータ入力
症例の臨床データを更新	CRF のデータ編集

- 9 [パスワード] フィールドにユーザのパスワードを入力し、[次へ] をクリックします。サマリ画面が表示されます。

- 10 次のいずれかを選択する方法もあります。

- **エラーで停止する** — エラーが発生した場合、InForm データインポートユーティリティに停止するように指示します。
- **詳細** — ファイル処理に合わせて詳細なメッセージを生成するように、InForm データインポートユーティリティに指示します。
- **出力ファイルを使用する** — ファイル名を指定して、出力ファイルをテキストファイルとして保存します。

- 11 [スタート] をクリックします。

InForm データインポートユーティリティはインポートファイル进行处理し、指定されている場合はメッセージ領域および出力ファイルにメッセージを出力します。また、データベースにデータを追加するかデータを更新します。

- 12 InForm データインポートユーティリティを閉じます。

データおよびマップファイルを利用するインポート

データおよびマップファイルの指定

データおよびマップファイルをインポートするには、次の手順に従います。

- 1 次のいずれかを実行します。
 - インポートするマップファイルを作成します。
 詳細については、「**データおよびマップファイルの作成**」(367ページ)を参照してください。
 - インポートする既存のマップファイルを編集します。
 詳細については、「**既存のデータおよびマップファイルの編集**」(369ページ)を参照してください。
- 2 [InForm データおよびマップファイル] オプションを指定してインポートを実行します。
 詳細については、「**[InForm データおよびマップファイル] オプションを使用したインポートの実行**」(378ページ)を参照してください。

データおよびマップファイルの作成

直接インポートする方法で使用するデータおよびマップファイルは、テキストファイルである必要があります。プレーンテキストファイルを作成できるものであれば、どのようなテキストエディタを使用しても、また生データを自動的にフォーマットする変換ツールを作成してもかまいません。

ファイルのデータを複数の CRF のターゲットコントロールと紐付けることができます。しかし、データ行のインポート先であるアイテムセットごとに、異なるファイルを使用する必要があります。また、CRF アイテムセットにインポートするデータは、通常の CRF アイテムにインポートするデータと区別して、別のファイルに含まれている必要があります。

インポートファイルは、以下の特徴を備えている必要があります。

- 各インポート行に、次のいずれかが含まれている必要があります。
 - 以下に示す形式で、症例の番号およびイニシャルを指定するフィールド:
subject_number (subject_initials)
 複数の症例が同一の症例番号/イニシャルをもつ可能性がある場合、症例の施設の略名を指定するフィールドも含める必要があります。
 - 症例のデータベース ID。
- インポートフィールドは、すべてパイプ記号 (|) で区切られている必要があります。
- 二重引用符 (") を、データファイルに含めないようにします。

また、一部のデータタイプを特別な形式にする必要があります。下に示すインポート関連のフィールドおよびコントロールのセットアップ方法については、本セクション以降

のセクションで説明します。

- **日付フィールド** (368ページ)
- **時間フィールド** (368ページ)
- **日時フィールド** (368ページ)
- **ネスト化されたコントロール** (369ページ)
- **チェックボックスと複数選択ドロップダウンリスト** (369ページ)
- **単位** (369ページ)
- **アイテム コメント** (369ページ)

日付フィールド

日付は、「月|日|年」の形式（年は 4 桁）の 3 つのフィールドで構成されている必要があります。その際、次の点に注意してください。

- 日付の一定のエレメントを指定しない場合は、次に示すようにその部分を空欄にします。

月||年

- 日付の一定のエレメントが不明な場合は、次に示すように UNK という文字を使用します。

月|UNK|年

時間フィールド

時間は、「時|分|秒」の形式（24 時間形式）の 3 つのフィールドで構成されている必要があります。その際、次の点に注意してください。

- 時間の一定の要素を指定しない場合は、次に示すようにその部分を空欄にします。

時|分|

- 不明な要素については、次に示すように UNK という文字を使用します。

時|UNK|UNK

日時フィールド

日時は、「月|日|年|時|分|秒」という形式（年は 4 桁、時間は 24 時間形式）の 6 つのフィールドで構成されている必要があります。その際、次の点に注意してください。

- いずれかのエレメントを指定しない場合は、その部分を空欄にします。
- いずれかのエレメントが不明である場合は、UNK という文字を使用します。

例：

```
PF|001 (CJB)|Sep|23|1975|1|1|215210|ラボデータ|臨床化学:|AG
Ratio|XGR|Nov|UNK|1998|0.8-2||2.0|N|||
PF|001 (CJB)|Sep|23|1975|1|1|215210|ラボデータ|臨床化
学:|ALAT(SGPT)|XGP|Nov|UNK|1998|0-48|U/L|42|N|||
PF|001 (CJB)|Sep|23|1975|1|1|215210|ラボデータ|臨床化学:|アルブミン
|XAL|Nov|UNK|1998|3.2-5|G/DL|4.3|N|||
PF|001 (CJB)|Sep|23|1975|1|1|215210|ラボデータ|臨床化学:|アルカリホスファターゼ
|XLK|Nov|UNK|1998|20-125|U/L|63|N|||
```


ネスト化されたコントロール

ネスト化されたコントロールはパイプ記号によって切り離された各コントロールによって構成されている必要があります。これらのフィールドのうち 1 つのみに、各データ行の情報が含まれることになります。

例:

```
PF|001 (CJB)|1975|Sep|23|1|1|215210|LABDATA| 血液学 |||
PF|001 (CJB)|1975|Sep|23|1|1|215210|LABDATA|| 尿検査 ||
PF|001 (CJB)|1975|Sep|23|1|1|215210|LABDATA||| 毒物学 |
```

チェックボックスと複数選択ドロップダウンリスト

チェックボックスと複数選択ドロップダウンリストは、カンマで区切られたゼロまたは 1 つ以上のコントロールのリストを含むフィールドで構成されている必要があります。

例:

```
PF|001 (CJB)|1975|Sep|23| ラボデータ |洞頻脈, 期外収縮, 洞性徐脈, 期外収縮(心房拍動) |
```

単位

単位を含むフィールドは、その単位で表されている数値の直後にある必要があります。

下に、体重およびポンドの例を示します。1 つのフィールドに体重の数値 (177) が含まれ、その単位 (ポンド) が直後のフィールドに含まれています。

```
PF|001 (CJB)|1975|Sep|23|70|in|177|lb|
```

アイテムコメント

アイテムコメントは、アイテムに関するテキストコメントを含むフィールドで構成されている必要があります。

例:

```
PF|001 (CJB)|1975|Sep|23|70|177| 靴とソックスをつけたまま測定された体重
```

既存のデータおよびマップファイルの編集

既存のインポートマップファイルを編集するには、以下の手順に従います。

注: マップファイルの定義を変更し、[キャンセル] ([完了] ではなく) をクリックした場合、[マップ] フィールドで指定したパスおよびファイル名をもつ空のマップファイルが保存されます。

- 1 [スタート] > [すべてのプログラム] > [Oracle Health Sciences] > [InForm 6.0] > [InForm データインポート] の順にクリックします。
InForm データインポートのメインウィンドウが表示されます。
- 2 [InForm データとマップファイル] を選択します。
- 3 [トライアル名] フィールドに、ファイルをインポートするスタディの名前を入力します。
ドロップダウンリストには、最近アクセスした 10 件のスタディが表示されます。
- 4 [Next] をクリックします。

インポートするデータおよびマップファイルを指定するためのダイアログボックス

が表示されます。

- 5 [データ] フィールドにインポートするデータファイルのフルパス名をタイプするか、[参照] をクリックします。
- 6 [マップ] フィールドにインポートするマップファイルのフルパス名をタイプするか、[参照] をクリックします。

注： ファイル名を入力し、マップファイルエディタを開くと、[開始] または [停止] をクリックするしないにかかわらず、ファイルが作成されます。このファイルは、**pfimport.exe** ファイルと同じ場所に保存されます。

- 7 [マップファイルの編集] をクリックします。
[フィールド定義] ダイアログボックスが表示されます。
- 8 マップファイルを編集するには、以下の手順に従います。
 - a CRF アイテム、CRF アイテムセットのどちらをデータの対象にするかを指定します。詳細については、「**サブミッションタイプの指定**」(371ページ)を参照してください。
 - b 入力フィールドタイプを指定します。詳細については、「**入力フィールドタイプの指定**」(371ページ)を参照してください。
 - c 各マップファイルの定義を、1 回に 1 フィールドずつ指定します。詳細については、「**アイテムパスの作成**」(372ページ)を参照してください。
 - d データタイプを指定します。詳細については、「**データタイプの指定**」(374ページ)を参照してください。
 - e インポートファイルの値と、InForm データベースのターゲットデータフィールドに定義された値とのマッピングを指定します。詳細については、「**文字列とチャイルドコントロールのマッピング**」(374ページ)を参照してください。
 - f データに複数の選択アイテムを含めるかどうかを指示します。詳細については、「**複数選択アイテムを含むデータの表示**」(376ページ)を参照してください。
 - g アイテムセット内で重複する情報を確認します。詳細については、「**アイテムセット内で重複した情報の確認**」(376ページ)を参照してください。
 - h 予定外のビジットへの情報のインポート詳細については、「**予定外のビジットへの情報のインポート**」(377ページ)を参照してください。
 - i インポートファイルにインポートフィールドを挿入するか、インポートファイルからインポートフィールドを削除します。詳細については、「**インポートフィールドの挿入と削除**」(377ページ)を参照してください。
 - j インポートファイルに対してマップをチェックします。詳細については、「**インポートファイルに対してマップをチェックする**」(378ページ)を参照してください。
- 9 マップファイルを保存したらマップファイルエディタを終了し、InForm データインポートユーティリティ画面に戻ってデータおよびマップファイルをインポートします。詳細については、「**データとマップファイルを利用したインポートの実行**」(378ページ)を参照してください。

サブミッションタイプの指定

1 次のいずれかの方法を選択してください。

- **フォーム** — インポートファイルのすべてのデータを、通常の CRF アイテムの対象とする場合に選択します。
- **アイテムセット** — インポートファイルのすべてのデータを、CRF アイテムセットの対象とする場合に選択します。既存のアイテムセットを更新するデータのみをインポートするように指定する別の方法として、[新しいアイテムセット行を許可しない] を選択することもできます。

2 既存の予定外のビジットのみを更新するデータをインポートする場合、[新しい繰り返しビジットインスタンスを許可しない] をチェックします。

3 [理由] フィールドに、データがデータロードで更新された場合に [データ値] 画面の [変更理由] セクションに表示されるテキストをタイプします。デフォルトでは、「ラボインポート」となります。

4 [続行] ボタンをクリックします。

[フィールド定義] ダイアログボックスが表示されます。

マップフィールドを定義するには、「**入力フィールドタイプの指定**」 (371ページ) で示されている手順に従います。

入力フィールドタイプの指定

[フィールド定義] ダイアログボックスを使用して、各マップフィールドを 1 つずつ定義します。各マップフィールドはインポートファイルの中のデータアイテムに対応しています。

[フィールド定義] ダイアログボックスで、次のフィールドタイプのいずれかを選択します。

- **InForm アイテムパス** — CRF のデータアイテムを対象とするデータが含まれます。このオプションでは、インポートフィールドに関する追加情報を提供する必要があります。詳細については、「**アイテムパスの作成**」 (372ページ) を参照してください。
- **被験者フィールド-番号 (イニシャル)** — 被験者_数 (被験者_イニシャル) の形式か症例データベース ID のいずれかによる、症例識別情報が含まれます。インポートデータを使用して症例データベース ID によって各症例を識別する場合は、[既知の PatientID を含むフィールド] チェックボックスをオンにします。

このオプションを選択すると、マップフィールドの定義が完了します。別のマップフィールド定義に移るか、マップファイルを保存してマップエディタを終了します。詳細については、「**マップファイルへのナビゲート**」 (378ページ) を参照してください。

- **施設の略名フィールド** — 症例が登録されている施設の略名が含まれます。このオプションを選択すると、マップフィールドの定義が完了します。別のマップフィールド定義に移るか、マップファイルを保存してマップエディタを終了します。詳細については、「**マップファイルへのナビゲート**」 (378ページ) を参照してください。
- **アイテムセットインデックスフィールド** — 1 を基準とする行番号のアイテムセッ

トインデックスを含んでいます。エントリ追加または繰り返しデータアイテムセットの既存の行を特定するのに使用します。このオプションを選択すると、マップフィールドの定義が完了します。別のマップフィールド定義に移るか、マップファイルを保存してマップエディタを終了します。詳細については、「**マップファイルへのナビゲート**」(378ページ)を参照してください。

- **このフィールドを無視** — フィールドをインポートしないように指示します。このオプションを選択すると、マップフィールドの定義が完了します。別のマップフィールド定義に移るか、マップファイルを保存してマップエディタを終了します。詳細については、「**マップファイルへのナビゲート**」(378ページ)を参照してください。
- **前のフィールドに対する単位のシンボル** — インポートファイルおよびマップファイルの前のフィールドに対する単位の記号が含まれます。CRF の単位記号は単位を特定するテキストであり、適切な XML ファイルの UNIT 定義の SYMBOL 属性で定義されたものです。このオプションを選択すると、マップフィールドの定義が完了します。別のマップフィールド定義に移るか、マップファイルを保存してマップエディタを終了します。詳細については、「**マップファイルへのナビゲート**」(378ページ)を参照してください。
- **直前のフィールドのコメントフィールド** — フォームまたはアイテムセットに関するコメントテキストが含まれます。このオプションを選択すると、マップフィールドの定義が完了します。別のマップフィールド定義に移るか、マップファイルを保存してマップエディタを終了します。詳細については、「**マップファイルへのナビゲート**」(378ページ)を参照してください。

必要に応じて、次のチェックボックスを 1 つ以上オンにします。

- **カンマで区切られた複数の値** — インポートファイルに、ラジオボタンコントロール、チェックボックス、ドロップダウンリストなどの複数選択コントロールに入力するデータを含めます。
- **アイテムセットのインスタンスをこのフィールドと照合する** — InForm アプリケーションにインポートする前に、InForm データインポートユーティリティでインポートファイルに重複したデータがないかどうかを確認します。
- **繰り返しビジットのインスタンスをこのフィールドと照合する** — インポートファイルでこのフィールドのビジット日付が重複していないかどうか、および InForm データベース内の繰り返しビジットのビジット日付を InForm データインポートユーティリティで確認します。

アイテムパスの作成

アイテムタイプとして **InForm** アイテムパスを選択した場合、インポートファイルのフィールドにターゲット CRF データアイテムの RefName パスを指定する必要があります。

アイテムパスを作成するには、以下のいずれかを実行します。

- **[InForm アイテムパス]** テキストボックスにパスを入力します。
詳細については、「**アイテムパスの明示的な入力**」(373ページ)を参照してください。
- **[データベースからパスを作成する]** ダイアログボックスのドロップダウンリストか

ら各 RefName を選択します。

詳細については、「[\[データベースからパスを作成する\] ダイアログボックスの使用](#)」(374ページ) を参照してください。

アイテムパスの明示的な入力

アイテムパスを指定するには、次のアイテムパスを使用します。

```
0.Visit.Form.Section.Itemset.Item[.control[.control...]]
```

アイテムパスの各エレメントはエレメントの定義に使用される RefName です。

- **0** — 現在の症例を示します。
- **Visit** — ビジットの RefName です。ビジット定義を含む XML ファイルで指定された通り。
- **Form** — CRF または他のフォームの RefName です。フォーム定義を含む XML ファイルで指定された通り。
- **Section** — セクションの RefName です。セクション定義を含む XML ファイルで指定された通り。
- **Itemset** — アイテムセットの RefName です。アイテムセット定義を含む XML ファイルで指定された通り。マップフィールド定義を作成するインポートデータが通常の CRF アイテムである場合 (アイテムセットでない場合)、0 (ゼロ) をタイプします。
- **Item** — アイテムの RefName です。アイテム定義を含む XML ファイルで指定された通り。アイテムセット内のアイテムごとに、別々のマップフィールドを作成します。
- **Control** — コントロールの RefName です。コントロール定義を含む XML ファイルで指定された通り。グループコントロールのエレメントにアクセスするには、子エレメントがネストされている各ペアレントコントロールを参照します。たとえば、グループコントロール内の 2 つのうち 1 つのテキストコントロールを扱うには、グループコントロールの RefName をタイプした後にテキストコントロールの RefName をタイプし、次のようにピリオドで名前を区切ります。
GroupControlRefName.TextControlRefName

例

- ビジット名が VISIT1、フォームが VSL、セクションが VS、アイテムが TEMPTEXT、フィールドが TEMPTEXT であるアイテムパスは、次のようになります。

```
0.Visit1.VSL.VS.0.TEMPTEXT.TEMPTEXT
```

- アイテムセット内のアイテムのアイテムパスは、次のようになります。

```
0.Visit1.SS.SECTION2.SS2GROUP.ONSETDATE.ONSETDATE
```

このコントロールは、ビジット名が VISIT1、フォームが SS、セクションが SECTION2、アイテムセットが SS2GROUP、アイテムが ONSETDATE の、ONSETDATE という名前の日付コントロールです。

- ビジット名が VISIT1、フォームが ECG、セクションが CHESTXRAY の VIEW ラジオコントロールアイテムにネストされているテキストコントロールのアイテムパスは、次のようになります。

```
0.Visit1.ECG.CHESTXRAY.0.VIEW.VIEWRADIO.OTHERTEXT
```

アイテムは VIEW という RefName によって、ラジオコントロールは VIEWRADIO

によって、テキストコントロールは OTHERTEXT によって、それぞれ特定されます。

[データベースからパスを作成する] ダイアログボックスの使用

- 1 [フィールド定義] ダイアログボックスで、**[InForm アイテムパス]** を選択します。
- 2 [パスの作成] をクリックします。
[データベースからパスを作成する] ダイアログボックスが表示されます。
- 3 [ビジット] ドロップダウンリスト で、ターゲットビジットの RefName を選択します。
- 4 [フォーム] ドロップダウンリスト で、ターゲット CRF の RefName を選択します。
- 5 [セクション] ドロップダウンリスト で、ターゲットセクションの RefName を選択します。
- 6 [アイテムセット] ドロップダウンリスト でアイテムセットの RefName を選択して、インポートデータをアイテムセットにロードする方法もあります。
- 7 [アイテム] ドロップダウンリスト で、ターゲットアイテムの RefName を選択します。

注：アイテムセットのアイテムごとに、別々のマップフィールド定義を作成する必要があります。

- 8 [コントロール] および [子(1~4)] ドロップダウンリスト で、ターゲットのグループおよび子コントロールの RefName を選択します。
- 9 **[OK]** をクリックします。

データタイプの指定

InForm アイテムパスのフィールドとして作成する各インポートフィールド定義には、データタイプを指定する必要があります。

インポートフィールドのデータタイプを指定するには、以下の手順に従います。

- [データタイプ] セクションで適切なボタンをクリックします。

InForm データインポートユーティリティは、次のいずれかが存在する場合、エラーを表示します。

- 無効な整数フィールド。無効な整数フィールドは、整数値に完全変換されません。例えば、123\$ は無効な整数フィールドです。
- 小数点前後の許容桁数が CRF コントロールの仕様に合致しない浮動小数として定義された文字列。
- CRF テキストコントロールの定義されたサイズ範囲内でないテキストコントロールとして特定された文字列。

文字列と子コントロールのマッピング

インポートファイルのフィールド値を、事前に定義された値をもつ CRF コントロールのデータベース定義にマップするには、InForm データインポートユーティリティを使用

します。このマッピング機能は、個々の子コントロールとそれらのデータベース ID パス間のコンパウンドコントロールについても、マッピングを生成します。

文字列のマッピング

インポートのターゲットとなる入力データフィールドが、オンラインユーザが事前に定義した値から選択する形式のコントロールである場合、インポートフィールドの値はデータベースで定義されている値（選択値）と同じである必要があります。これらの値では、大文字と小文字が区別されます。

インポートファイルが定義されているデータベース値と一致しない場合、インポートファイルを変換してそれらの値に適合させることができます。

また、InForm データインポートユーティリティの文字列マッピング機能を使用して、インポートファイルの値と InForm データベースのターゲットデータフィールドに定義されている値とのマッピングを指定することもできます。

マップファイルフィールド定義の作成中にこの機能を利用するには、次の手順に従います。

- 1 [フィールド定義] ダイアログボックスで、[文字列のマップ] をクリックします。
[文字列マップ] ダイアログボックスが開きます。
- 2 上のフィールドに、インポートファイルに表示される、コントロールの適当な値を入力します。
- 3 下のフィールドに、データベースで定義されているようにコントロールの値を入力します。この定義は、一般に VALUE 属性によって、フォームおよびデータアイテムの定義をデータベースにロードする際に使用される XML ファイルに指定されるものです。
- 4 [マップ先] をクリックします。
InForm データインポートユーティリティにより、値のペアが [現在マップされている文字列] フィールドに移動されます。たとえば、Yes をコントロールの定義値とし、ファイルに表示される値として Y と入力した場合、[現在マップされている文字列] フィールドには、「Yes は Y としてマップされる」のようにマッピング内容が表示されます。
- 5 それぞれの組み合わせ可能な値（フィールドがインポートファイルとコントロール定義を持つことができる）のマッピング定義を繰り返します。
- 6 [マップの更新] をクリックします。

子コントロールのマッピング

ターゲットコントロールが別のコントロール（ラジオボタンのリスト内のフィールドなど）にネストされている場合は、グループコントロールおよびグループ内の各子コントロールについて、別々のマップフィールドを作成する必要があります。同様に、インポートファイルには、グループコントロールおよび選択可能な各子コントロールのフィールドが含まれている必要があります。

InForm データインポートユーティリティは、選択されたグループコントロールに特定の値を割り当てるため、子コントロールの名前をそれらのデータベース ID パスにマップ

します。

グループコントロールマップフィールドに子コントロールマッピングを生成するには、以下の手順を実行します。:

- 1 [フィールド定義] ダイアログボックスで、[文字列のマップ] をクリックします。
[文字列マップ] ダイアログボックスが開きます。
- 2 [子コントロールマップの作成] をクリックします。
[現在マップされている文字列] フィールドには、子コントロールの RefName とそれらのデータベース ID との間のマッピング内容が表示されます。グループコントロールを定義しているマップフィールドに対応するインポートファイルフィールドに、データの提供先とする子コントロールの RefName をタイプします。
- 3 [マップの更新] をクリックします。

データに複数選択アイテムを含める指示

複数選択コントロール、またはチェックボックス、複数選択 ドロップダウンリストにデータをインポートするには、以下の手順を実行します。

- 1 インポートファイルに、1 つのフィールドに適用する選択をすべて含めます。選択したものを、それぞれパイプ(|)で区切ります。
例えば、タバコおよび葉巻、なしを含むリストからタバコおよび葉巻のチェックボックスが選択されるようにし、それらに定義された値が「タバコ」および「葉巻」、「なし」である場合には、インポートファイルには次の値が含まれることになります。
|タバコ,葉巻|
- 2 マップフィールドを、チェックボックスまたはドロップダウンリストのペアレントコントロールを参照する InForm アイテムパスフィールドとして定義します。
- 3 [フィールド定義] ダイアログボックスで、[カンマで区切られた複数の値] チェックボックスをオンにします。

アイテムセット内での重複した情報の確認

InForm データインポートユーティリティは、データベースの既存のアイテムセットにマップしようとしているデータアイテムセットと比較することによって、データがデータベースにすでに存在しているかどうかを判断することができます。

たとえば、ラボ情報と症例名をインポートしようとしており、そのデータおよび適合させようとしているテストデータがデータベースにすでに存在している場合、データの重複が検出されることがあります。この重複データが検出された場合、InForm データインポートユーティリティはデータベースに、その 2 番目のデータインスタンスを追加しません。

この機能を使用するには、以下の手順に従います。

- 1 [サブミッションタイプ] 画面で、[アイテムセット] を選択します。
- 2 [続行] ボタンをクリックします。

[フィールド定義] ダイアログボックスが表示されます。

- 3 [次へ] をクリックして、データオブジェクトに移動します。
- 4 データベース内のアイテムセットと照合するデータオブジェクトの **InForm** アイテムパスを選択します。
- 5 このフィールドにある [アイテムセットのインスタンスをこのフィールドと照合する] を選択します。
- 6 比較したいデータアイテムが他にある場合は、上記の手順を繰り返します。

予定外のビジットへの情報のインポート

- 1 [サブミッションタイプ] 画面で、[アイテムセット] を選択します。
- 2 [続行] ボタンをクリックします。
[フィールド定義] 画面が表示されます。
- 3 データベース内の繰り返しビジットと照合するビジットの日付のアイテムパスを選択します。
- 4 このフィールドにある [繰り返しビジットのインスタンスをこのフィールドと照合する] を選択します。
- 5 インポートしたい予定外のビジットが他にある場合は、上記の手順を繰り返します。

インポートフィールドの挿入と削除

- 1 [フィールド定義] ダイアログボックスで、[フィールドの挿入] をクリックします。
InForm データインポートユーティリティは、[フィールド定義] ダイアログボックスのデータ入力フィールドをクリアします。
- 2 新規フィールドの定義を入力します。
- 3 [次へ]、[戻る]、[完了] のいずれかを、必要に応じてクリックします。
InForm データインポートユーティリティは、[フィールドの挿入] をクリックしたときに表示されていたフィールドの直前に、新規のフィールド定義を挿入します。
- 4 マップ定義を保存するには、[完了] をクリックします。

注:[完了] または [フィールドの削除] をクリックして新規フィールド定義を作成または削除するまで、他の作業はできません。

インポートフィールド定義を削除するには、以下の手順に従います。:

- 1 [次へ] または [戻る] をクリックして、削除するフィールド定義を特定します。
- 2 [フィールドの削除] をクリックします。
- 3 マップ定義を保存するには、[完了] をクリックします。

注:マップファイルの定義を変更し、[キャンセル] ([完了] ではなく) をクリックした場合、[マップ] フィールドで指定したパスおよびファイル名をもつ空のマップファイルが保存されます。

インポートファイルに対してマップをチェックする

実際に処理するインポートファイルに対してマップフィールド定義をチェックするには、InForm データインポートユーティリティマップファイルエディタを使用します。

インポートファイルの最初の行に対してマップを照合チェックするには、次の手順を実行します。

- [フィールド定義] ダイアログボックスでフィールド定義をナビゲートし、[サンプルデータ] フィールドに表示されているデータをインポートファイルのデータと比較します。

マップファイルのナビゲート

マップファイルのフィールドを作成するとき、フィールドの定義を前後に動かすことが出来るように、フィールドの定義はお互いに順序でつなぎ合わされます。

[フィールド定義] ダイアログボックスの下側にあるコントロールボタンを使用して、以下を実行します。

- マップファイルの前のフィールドを表示するには、[戻る] をクリックします。
- マップファイルの次のフィールドを表示するには、[次へ] をクリックします。
- 新規フィールド定義を作成するには、マップファイルの最後のフィールドに進み、[次の作成] をクリックします。

注: [完了] または [削除] をクリックして新規フィールド定義を作成または削除するまで、他の作業はできません。

[InForm データおよびマップファイル] オプションを使用したインポートの実行

注: データベースにデータをインポートするには、InForm データインポートユーティリティを起動する前にサーバが稼働している必要があります。インポートされたデータが症例ステータスアイコンに反映されているかどうかを確認するには、インポート後にサーバを停止し、再起動する必要があります。

InForm データおよびマップファイルを InForm データベースにインポートするには、以下の手順に従います。

- 1 [スタート] > [すべてのプログラム] > [Oracle Health Sciences] > [InForm 6.0] > [InForm データインポート] の順にクリックします。
InForm データインポートのメインウィンドウが表示されます。
- 2 [InForm データとマップファイル] を選択します。
- 3 [トライアル名] フィールドに、ファイルをインポートするスタディの名前を入力します。
ドロップダウンリストには、最近アクセスした 10 件のスタディが表示されます。
- 4 [Next] をクリックします。

インポートするデータおよびマップファイルを指定するためのダイアログボックスが表示されます。

- 5 [データ] フィールドにインポートするデータファイルのフルパス名をタイプするか、[参照] をクリックします。
- 6 [マップ] フィールドにインポートするマップファイルのフルパス名をタイプするか、[参照] をクリックします。
- 7 オプションで、マップファイルを編集するには、[マップファイルの編集] をクリックします。詳細については、「**既存のデータとマップファイルの編集**」（369ページ）を参照してください。
- 8 [次へ] をクリックします。

ダイアログボックスが表示され、InForm のユーザ名とパスワードを入力するよう指示されます。

注： [解析のみ] を選択した場合、ユーザ名とパスワードを指定する必要はありません。

- 9 [名前] フィールドに、インポートするデータに関して適切な権限を有する InForm ユーザの名前を入力します。

インポートの目的	ユーザに必要な権限
症例の臨床データを追加	CRF へのデータ入力
症例の臨床データを更新	CRF のデータ編集

- 10 [パスワード] フィールドにユーザのパスワードを入力し、[次へ] をクリックします。
サマリ画面が表示されます。
- 11 次のいずれかを選択する方法もあります。
 - **エラーで停止する** — エラーが発生した場合、InForm データインポートユーティリティに停止するように指示します。
 - **詳細** — ファイル処理に合わせて詳細なメッセージを生成するように、InForm データインポートユーティリティに指示します。
 - **出力ファイルを使用する** — ファイル名を指定して、出力ファイルをテキストファイルとして保存します。
- 12 [スタート] をクリックします。

InForm データインポートユーティリティはインポートファイルを処理し、指定されている場合はメッセージ領域および出力ファイルにメッセージを出力します。また、データベースにデータを追加するかデータを更新します。

- 13 InForm データインポートユーティリティを閉じます。

エラーファイルのチェック

- 指定したファイル名をもつ出力ファイルを、指定したディレクトリで開きます。

注：出力ファイルのディレクトリとファイル名を指定しなかった場合、エラーファイルは **InForm** データインポートユーティリティの実行ファイル **PFImport.exe** が格納されているディレクトリに保存されます。インポートの完了後に、**ERR** ファイルの存在をチェックし、ファイルにエラーがないかどうかを確認します。

日付と時間の妥当性チェック

InForm データインポートユーティリティは、日時データについて、以下の妥当性チェックを実行します。

妥当性チェック	説明
CRF 日時コントロールの DISPLAY 属性	MedML の DISPLAY 属性は、日付、時間または日時の両方をインポートする CRF 日時コントロールの各コンポーネントについて、True に設定されている必要があります。 InForm データインポートユーティリティは、 DISPLAY 属性が False となっている日付/時間コンポーネントを無視するため、インポートは行われません。
インポートフィールドの年の値	InForm データインポートユーティリティでは、インポートされた年が CRF 日時コントロールで許容される範囲外にある場合、エラーが表示されます。
インポートフィールドの日の値	年と月の値が有効であれば、InForm データインポートユーティリティは日の値が妥当かどうかをチェックします。
CRF 日時コントロールの REQUIRED 属性	InForm データインポートユーティリティでは、CRF 日時コントロールで、任意の日時コンポーネントが REQUIRED としてコード化されており、インポートファイルで欠落している場合、エラーが表示されます。
CRF 日時コントロール の UNKNOWN 属性	InForm データインポートユーティリティでは、インポートした日時コンポーネントが UNK としてコード化されているにもかかわらず、CRF 日時コントロールの UNKNOWN 属性が有効になっていない場合、エラーが表示されます。
一貫性チェック	InForm データインポートユーティリティでは、CRF 日時コントロールで一貫性チェックが有効になっており、日時コンポーネントが上位コンポーネントまたは UNK 値なしでインポートファイルに存在する場合、次のような形でエラーが表示されます。 ss:mm:hh dd/mm/yy たとえば、インポートした日時に月なしで [日] フィールドが含まれていると、InForm データインポートユーティリティではエラーが表示されます。

妥当性チェック	説明
インポートフィールドの 時間および分の値	時間と分の両方の値が有効な範囲内にある か、UNK としてコードされている必要があ ります。

コマンドラインからの InForm データインポートユーティリティの実行

マップファイルの作成中または編集時は、InForm データインポートユーティリティを対話モードで使用します。マップファイルの完成後は、PFConsole ユーティリティでコマンドラインからユーティリティを実行するか、定期的に実行されるバッチファイルにインポートコマンドを挿入できます。

注： Oracle では、PFConsole ユーティリティを使用して InForm データインポートユーティリティを実行することを強く推奨しています。詳細については、「**コマンドラインからの PFConsole ユーティリティの実行**」(15ページ)を参照してください。

使用するコマンドラインパラメータは、以下の通りです。パラメータ名とその値の間には、スペースを挿入する必要があります。

```
PFImport [-?][-autorun] [-verbose] [-parse] [-trial <trialname>] [-errstop]
-norules
[[/accountparams:path_to_password_file]
[@<rsp_file>] [-xml
<xml_file>...<xml_file>]] [-template <map_file>] [-import <data_file>]
```

注： すべてのコマンドラインアプリケーション(MedML インストーラユーティリティ、InForm データインポートユーティリティ、および InForm データエクスポートユーティリティ)は、デフォルトプロダクトロケールを使用します。デフォルトプロダクトロケールは InForm のインストール中、または PFAdmin コマンドで設定されます。

パラメータ	変数	説明
PFImport		InForm データインポートユーティリティを起動します。
-?		コマンドヘルプを表示します。
-autorun		コマンド画面で、InForm データインポートユーティリティを実行します。
-verbose		InForm データインポートユーティリティは入力ファイルの処理中に詳細なメッセージを出力します。
-parse		実際にデータをデータベースに反映することなく、インポートが実行されます。この機能は、実際に MedML ファイルをデータベースにインポートする前に、構文をテストするのに使用します。
-trial	<i>trialname</i>	データのインポート先とするスタディを指定します。スタディのフルパス名を使用します。

パラメータ	変数	説明
-errstop		InForm データインポートユーティリティは、エラーが発生すると処理を停止します。指定していない場合、エラーを含むタグはスキップされ、インポートはファイル内の次のデータタグから続行されます。
-norules		インポート中にはルールが実行されません。このパラメータは必須です。このオプションは、同期接続が定義されていない場合にのみ有効です。
/accountparams:" path_to_password_file"		<p>指定された場合、コマンドの実行に必要なユーザ名とパスワードを含むテキストファイルへのパスが含まれます。</p> <p>accountparams オプションが指定されていない場合、コマンドは必要なユーザ名とパスワードを入力するよう要求します。</p> <p>パラメータファイルの形式は「パラメータ=値」です。パラメータごとに改行し、行中ではスペースを使用しません。</p> <p>注： 手動でテキストファイルを作成する必要があります。</p>
-validate		必要なすべての XML タグが存在し、指定されたコントロールパスが特定できるかどうかを、データをロードすることなくチェックします。このパラメータを使用すれば、インポートする前に XML ファイルを検証することもできます。
@	RSP_file	処理する XML ファイルの名前を含むレスポンスファイル (拡張子 .rsp) をサブミットします。レスポンスファイルのフルパス名を指定します。このオプションか -xml オプションもしくは -template と -import のオプションを使用します。
-xml	xml_file	処理する 1 つまたは複数の XML ファイルをサブミットします。複数の XML ファイルを指定する場合はスペースで区切ります。このオプションか @ オプション、もしくは -template と -import のオプションを使用します。

パラメータ	変数	説明
-template	<i>map_file</i>	データおよびマップファイルオプションを使用している場合、次にくるパラメータがマップファイル名になるように指示します。マップファイルのフルパス名を指定します。このオプション、@ オプション、-xml オプションのいずれかを使用します。マップファイルを指定する場合、-import オプションによってインポートファイルも指定する必要があります。
-import	<i>data_file</i>	次に続くパラメータがインポートファイル名であることを示します。データファイルのフルパス名を指定します。このオプション、@ オプション、-xml オプションのいずれかを使用します。インポートファイルを指定する場合、-template オプションによってマップファイルも指定する必要があります。
-datetimeformat		<p>日時情報に使用する形式。日付順形式のスタディ構成の設定を上書きするにはこのパラメータを使用します。有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> • MMDDYYYY • DDMMYYYY • YYYYMMDD
-output	<i>output_file</i>	ファイルのフルパスとファイル名。このパラメータを使用すると、出力画面に出力メッセージは表示されません。

データインポートの拡張

- **InForm アプリケーションサーバからのログオフ** — インポートを実行する前に InForm アプリケーションサーバからログオフします。サーバは稼動状態のままにしておくことができます。
- **WWW Publishing Service の停止** — インポート中に他のユーザがデータを入力できないように、WWW Publishing Service を停止し、再起動します。
- **Oracle update statistics スクリプトの実行** — ファイルをインポートした直後に、Oracle update statistics スクリプトを実行します。
- **ホームページの変更** — インポートを開始する前に、インポートの実行中はシステムパフォーマンスが低下する可能性があることを、スタディのホームページでユーザに警告します。
- **症例別データ編成** — 可能な場合、インポートするデータを症例 ID 順にソートします。

第 4 章

InForm データエクスポート ユーティリティ

この章の内容

InForm データエクスポートユーティリティの概要	390
InForm データエクスポートユーティリティの実行	391
CDD へのデータのエクスポート	392
コントロール名/値のペアのエクスポート	395
コマンドラインからの InForm データエクスポートユーティリティの実行	398

InForm データエクスポートユーティリティの概要

InForm データエクスポートユーティリティの出力オプション

- **CDD** — 顧客定義データベースにデータをエクスポートします。
- **コントロール名/値ペア** — データパス名とデータ値で構成されるパイプ区切りファイルを作成します。

注：InForm データエクスポートユーティリティの使用は、症例数が **10** 万人未満のスタディに制限されています。

CRF Submit アプリケーションを使用すると、データを PDF 形式でエクスポートできます。CRF Submit アプリケーションは、単独で動作するユーティリティで、次の操作を行うことができます。

- FDA (食品医薬品局) に提出する目的の症例データの PDF を作成します。たとえば、症例データの PDF を作成して、新薬申請 (NDA) と共に FDA に提出することができます。
- 臨床データの PDF を内部のアーカイブに保存します。たとえば、症例を新たな施設に移す前に、その症例のすべての臨床データをアーカイブできます。

InForm データエクスポートユーティリティの実行

InForm データエクスポートユーティリティオプションは、すべて同一の実行ファイルで実行されます。

注： InForm データエクスポートユーティリティを実行する際に、InForm アプリケーションサーバが稼動している必要はありません。

InForm データエクスポートユーティリティを実行するには、次の手順を実行します。

- 1 [スタート]>[プログラム]>[Oracle Health Sciences]>[InForm 6.0]>[InForm データエクスポート] の順にクリックします。

InForm データエクスポートのメイン画面が表示されます。

- 2 [CRB] セクションで、次のいずれかを選択します。

- すべての CRB のエクスポート
- フリーズ済 CRB をエクスポートする
- ロック済 CRB をエクスポートする

- 3 [出力オプション] セクションで、エクスポートに使用する形式を選択します。

- CDD
- コントロール名／値ペア

詳細については、「**InForm データエクスポートユーティリティ出力オプション**」(392 ページ)を参照してください。

- 4 [トライアル名] フィールドに、情報のエクスポート元となるスタディの名前を入力します。
- 5 [ログ出力名] フィールドに、エクスポート中に作成されるログファイルのパスとファイル名を入力するか、[参照] をクリックしてログファイルとして使用するファイルの場所を特定します。
- 6 [次へ] をクリックします。

下記のセクションでは、各エクスポートタイプの入力画面について説明しています。

- **CDD へのデータのエクスポート** (392ページ)
- **名前／値のエクスポート** (397ページ)

CDD へのデータのエクスポート

CDD へのデータエクスポートの概要

CDD 出力オプションは CDD マッピングパラメータと InForm データベースのすべてのデータを読み取り、CDD に次の順序で入力します。

- 施設データ
- コメント
- 症例情報
- CRF データ

これは、スタディ開始後に CDD 定義が変更されたときに、それをデータベースに再入力する場合に便利な機能です。

CDD へのデータの移動

- 1 Central Designer アプリケーションを使用して、InForm データベースのデータを CDD にマップする方法を指定したパラメータを含む CDD 定義を生成します。
- 2 CDD 出力オプションによって InForm データエクスポートユーティリティを実行し、InForm データベースのデータを読み取って CDD データベースに入力します。

詳細については、「**CDD データ用エクスポートの実行**」(392ページ)を参照してください。

CDD データ用エクスポートの実行

注：CDD データ用 InForm データエクスポートユーティリティを実行する前に、まず InForm アプリケーションサーバを停止する必要があります。

- 1 既存の CDD をバックアップします。
- 2 [CRB] セクションで、以下のいずれかを選択します。
 - すべての CRB のエクスポート
 - フリーズ済 CRB をエクスポートする
 - ロック済 CRB をエクスポートする
- 3 [出力オプション] セクションで、[CDD] をクリックします。
- 4 [トライアル名] フィールドに、トライアル名を入力します。
- 5 [ログ出力名] フィールドで、エクスポートデータを保存するログファイルの名前を入力するか、[参照] をクリックしてファイルの場所を特定します。
[CDD エクスポートオプション] ダイアログボックスが表示されます。
- 6 [DSN] フィールドで、以下のいずれかを実行します。

- データの出力先の CDD について定義されている ODBC DSN の名前を入力します。
 - 新しい DSN を作成します。
詳細については、「**エクスポート用の新規 DSN の作成**」（393ページ）を参照してください。
- 7 [ユーザ名] および [パスワード] フィールドに、CDD へのアクセスに使用するユーザ名とパスワードを入力します。
 - 8 現在のデータベースを破棄し、DSN に関連付けられている RefName に基づいて新しいスキーマを作成するには、[スキーマの新規作成] チェックボックスをオンにします。
 - 9 [テーブルスペース] フィールドに、新規スキーマを置くテーブルスペースの名前を入力します。
 - 10 [次へ] をクリックします。
[エクスポート概要] 画面が表示されます。

警告: 次の手順に進む前に、入力した情報がすべて正しいことを確認してください。
InForm データエクスポートユーティリティを実行すると、現在のデータベースのデータはすべて失われます。

- 11 [終了] をクリックします。
InForm データエクスポートユーティリティは CDD へのデータ入力を開始し、その進捗状況を示すメッセージを表示します。

エクスポート用の新規 DSN の作成

- 1 [CDD エクスポートオプション] 画面に、以下の情報を入力します。
 - [DSN] フィールドに、作成する ODBC DSN の名前を入力します。
 - [ユーザ名] および [パスワード] フィールドに、CDD へのアクセスに使用するユーザ名とパスワードを入力します。
- 2 [DSN の作成] をクリックします。
[CDD データソース] 画面が表示されます。[CDD DSN] フィールドに DSN が表示されます。
- 3 [CDD 参照名] ドロップダウンリスト で、新しい DSN と関連付ける RefName を選択します。
- 4 [データベースサーバ] フィールドに、使用するサーバの名前を入力します。
- 5 [OK] をクリックします。

[CDD エクスポートオプション] 画面が再表示され、DSN、ユーザ名、およびパスワードがそれぞれのフィールドに表示されます。

注：エクスポート用 **DSN** を新規に作成する場合、新しい **DSN** は作成時に実行する **InForm** データインポートセッションでのみ有効となります。したがって、**InForm** データエクスポートユーティリティツールを閉じると削除されます。**InForm** データエクスポートユーティリティツール外でも使用できる **DSN** を作成するには、**ODBC** マネージャまたは **PFAAdmin CONFIG CDD Setup** コマンドを使用します。**PFAAdmin** コマンドの詳細については、『スタディとレポート設定ガイド』を参照してください。

コントロール名/値のペアのエクスポート

コントロール名/値のペアのエクスポートの概要

コントロール名/値出力オプションは、InForm データベースのデータをプレーンテキストファイルにエクスポートする機能です。変換または分析を行うために、データを抽出したいときに便利です。

注: コントロール名/値出力オプションは、症例データで発生する名前付き値のみをエクスポートします。規制ドキュメントまたはビジットレポートフォームの値をエクスポートすることはありません。

出力ファイルの形式

コントロール名/値ペア用のエクスポートを実行することによって作成される出力ファイルでは、フィールドはパイプで区切られ、不適切なフィールドは空白状態にされます。その際、次に示すデータが使用されます。

コントロール名/値 フィールド	説明
症例識別子	症例イニシャルと、それに続く、括弧で囲まれた症例番号
RefName パス	次の順で示される RefName のパス <ul style="list-style-type: none"> • ビジット • 繰り返しフォームセットインデックス — 繰り返された予定外ビジットのうち、どのインスタンスを参照するかを指定するのに使用されます。ビジットが予定外のビジットでない場合、値は 1.000 です。 • フォーム • 繰り返しフォームインデックス。 • セクション • アイテムセット • アイテムセットインデックス — 参照するアイテムセット行を指定するのに使用されます。 • アイテム • コントロール
標準化された値	データベースで指定されている基本単位に変換された後の値
入力値	コントロールのために入力された値

関連フォームの出力ファイル形式

関連フォームのエクスポートを実行することによって作成される出力ファイルでは、フィールドはパイプで区切られ、不適切なフィールドは空白状態にされます。その際、次に示すデータが使用されます。

関連フォームフィールド		説明
症例識別子		症例イニシャルと、それに続く、括弧で囲まれた症例番号
RefName パス		次の順で示される RefName のパス <ul style="list-style-type: none">• ビジット• 繰り返しフォームセットインデックス — 予定外ビジットのどのインスタンスを参照するかを指定するのに使用されます。ビジットが予定外のビジットでない場合、値は 1.000 です。• フォーム• 繰り返しフォームインデックス。 最初のフォームのパスがこの順に表示され、関連フォームのパスは、最初のフォームの後に二重引用符で囲まれた状態で表示されます。
標準化された値		データベースで指定されている基本単位に変換された後の値
入力値		コントロールのために入力された値

例

次の出力ファイル例は、症例 PA1(1) に関するエクスポートデータの一部を示したものです。

```
PA1(1)|Visit1|1|EYE|VISION||0|LNEAR.EYEGROUP||
PA1(1)|Visit1|1|EYE|VISION||0|LNEAR.EYEGROUP.EYE_TC|20.000000|20
PA1(1)|Visit1|1|EYE|VISION||0|LNEAR.EYEGROUP.UNEYE_TC|20.000000|20
PA1(1)|Visit1|1|EYE|VISION||0|RNEAR.EYEGROUP||
PA1(1)|Visit1|1|EYE|VISION||0|RNEAR.EYEGROUP.EYE_TC|20.000000|20
PA1(1)|Visit1|1|EYE|VISION||0|RNEAR.EYEGROUP.UNEYE_TC|20.000000|20
PA1(1)|Visit1|1|EYE|VISION||0|LDIST.EYEGROUP||
PA1(1)|Visit1|1|EYE|VISION||0|LDIST.EYEGROUP.EYE_TC|20.000000|20
PA1(1)|Visit1|1|EYE|VISION||0|LDIST.EYEGROUP.UNEYE_TC|20.000000|20
PA1(1)|Visit1|1|EYE|VISION||0|RDIST.EYEGROUP||
PA1(1)|Visit1|1|EYE|VISION||0|RDIST.EYEGROUP.EYE_TC|20.000000|20
PA1(1)|Visit1|1|EYE|VISION||0|RDIST.EYEGROUP.UNEYE_TC|20.000000|20
PA1(1)|Visit1|1|EYE|RETINAL||0|RWOOL.WOOLGROUP||N
PA1(1)|Visit1|1|EYE|RETINAL||0|LWOOL.WOOLGROUP||N
PA1(1)|Visit1|1|EYE|RETINAL||0|RNICK.NICKRADIO||N
PA1(1)|Visit1|1|EYE|RETINAL||0|LNICK.NICKRADIO||N
PA1(1)|Visit1|1|HH|PT|PT|1|THERAPYTEXT.THERAPYTEXT||Name of a doctor
```

次のエクスポートファイルフラグメントは、繰り返しフォームの例です。

```
ABC(101)|CommonCRF|1.000|L_ConMeds|291130301735008.000|L_ConMeds||0.000|CMHiddenJ.CMHiddenJ||test
```

次のエクスポートファイルフラグメントは、関連の例です。

```
BBB(111)|CommonCRF|1.000|LAE1|319514859682043.000| "BBB(111)|CommonCRF|1.000|L_ConMeds|319587403827450.000"
```

コントロール名/値ペア用のエクスポートの実行

- 1 [CRB] セクションで、以下のいずれかを選択します。
 - すべての CRB のエクスポート
 - フリーズ済 CRB をエクスポートする
 - ロック済 CRB をエクスポートする
- 2 [出力オプション] セクションで、[コントロール名/値ペア] をクリックします。
- 3 [トライアル名] フィールドに、トライアル名を入力します。
- 4 [ログ出力名] フィールドで、エクスポートデータを保存するログファイルの名前を入力するか、[参照] をクリックしてファイルの場所を特定します。
[コントロール名/値ペアのエクスポートオプション] ダイアログボックスが表示されます。
- 5 [ファイル名] フィールドに出力ファイル名を NV 拡張子を含めて入力するか、[参照] をクリックしてファイルの場所を特定します。
- 6 [削除されたアイテムセット行] セクションで、次のいずれかを選択します。
 - InForm データエクスポートユーティリティに、削除済みのフォームまたはアイテムセット行からデータを出力しないように指示するには、[削除済みのフォームまたはアイテムセット行からはデータを出力しない] を選択します。
 - 削除済みのフォームまたはアイテムセット行からデータをエクスポートするには、以下の手順に従います。
 - 1 [削除済みのフォームまたはアイテムセット行からデータを出力する] を選択します。
 - 2 [削除された接頭辞] ボックスで、InForm データエクスポートユーティリティがエクスポートする、削除されたアイテムセット行のそれぞれに追加する接頭辞を指定します。
- 7 [Next] をクリックします。

InForm データエクスポートユーティリティは出力ファイルをビルドし、データ抽出の進捗状況を示すメッセージを表示します。

コマンドラインからの InForm データエクスポートユーティリティの実行

InForm データエクスポートユーティリティをコマンドラインから実行するには、次のパラメータを使用します。

注： Oracle では、PFConsole ユーティリティを使用して InForm データエクスポートユーティリティを実行することを強く推奨しています。詳細については、「**コマンドラインからの PFConsole ユーティリティの実行**」（15ページ）を参照してください。

注： すべてのコマンドラインアプリケーション（MedML インストーラユーティリティ、InForm データインポートユーティリティ、および InForm データエクスポートユーティリティ）は、デフォルトプロダクトロケールを使用します。デフォルトプロダクトロケールは InForm のインストール中、または PFAdmin コマンドで設定されます。

パラメータ	変数	説明
PFConsole ユーティリティ		PFConsole ユーティリティを起動します。
PfExport		InForm データエクスポートユーティリティを起動します。
/accountparams:"path_to_ password_file"		<p>指定された場合、コマンドの実行に必要なユーザ名とパスワードを含むテキストファイルへのパスが含まれます。</p> <p>accountparams オプションが指定されていない場合、コマンドは必要なユーザ名とパスワードを入力するよう要求します。</p> <p>パラメータファイルの形式は「パラメータ=値」です。パラメータごとに改行し、行中ではスペースを使用しません。</p> <p>注： 手動でテキストファイルを作成する必要があります。</p>
-autorun		<p>コマンド画面で、InForm データエクスポートユーティリティを実行します。</p> <p>詳細については、「コマンドラインからの PFConsole ユーティリティの実行」（15ページ）を参照してください。</p>
-?		使用ステートメントを表示します。
-help		InForm データエクスポートユーティリティのオンラインヘルプを表示します。
-Trial	トライアル名	データをエクスポートするスタディの名前。

パラメータ	変数	説明
-outfile	output log file name	次のパラメータが出力ファイルであることを示します。 InForm データエクスポートユーティリティによって表示されるメッセージのテキストを含む出力ファイルのフルパス名です。オプション。
-cdd	ODBC DSN Name	データを CDD にエクスポートすることを指定します。 CDD 出力ツールのエクスポート対象として使用する CDD の ODBC DSN 名が必要です。
-CreatDSN	OraConnStr	CDD エクスポートにのみ適用され、CDD データベース内で新しい DSN を作成します。
-RefName	CDDRefName	CDD エクスポートにのみ適用され、現在のユーザをドロップして RefName によって定義される新しいスキーマを作成します。
-TBSP	OraTabSpace	CDD エクスポートにのみ適用され、CDD データベース内で新規 DSN を作成する際に新しいスキーマを置くテーブルスペースの名前です。
-crfhelp		CRF ヘルプをエクスポートするように指示します。
-nvfile	output nv file name	コントロール名/値ペアについて InForm データエクスポートユーティリティを実行することを指定します。 -nvfile フラグを指定する場合、名前/値ペアについてエクスポートツールを実行する際に作成されるエクスポートファイルの名前が必要です。
-DelPrefix	string to prefix data marked as deleted	エクスポートに先立って、データが InForm アプリケーションから削除されたことを示すために、エクスポートファイルのデータ行に追加される文字列です。
-OutputUN		不明の日付フィールド (ある場合) に UN を出力します。

パラメータ	変数	説明
-sysadmin		CRF ヘルプをエクスポートする権限を有しているシステム管理者の名前です。 次にくるパラメータがシステム管理者 ID であることを示します。この ID は CRF のエクスポートに必要なものです。
-outputtype	ALL EPIC STANDARD	エクスポートのタイプを指定します。
-customheader	text of header	PDF 形式にのみ適用されます。オプションです。カスタムヘッダーの名前を指定できます。

例

以下は、コマンドラインから InForm データエクスポートユーティリティを実行する構文の例です。

```
PFConsole
PFExport [-autorun] [-?][-help] [-Trial trial name] [-outfile outfile name] [-cdd
ODBC DSN Name] [/accountparams:path_to_password_file]

[-CreatDSN OraConnStr][-RefName CDDRefName][-TBSP OraTabSpace]
[-crfhelp]
[-nvfile nv file name [-DelPrefix string to prefix data marked as deleted]]

[-OutputUN]]
```

DEL ファイル形式

この DEL ファイルのセクションでは、6666 から 5555 に変更された担当医キーのレコードが示されています。この削除ファイルは、カンマ区切り値が使用されている状態でエクスポートされたものです。これは、この担当医の症例のすべてのビジット/ビジット日について、削除レコードが生成されることを意味します。

```
6666,002,1111,,V10,0,20000802,,,,,,,,0,1,Investigator key has changed from '6666' to
'5555',DELETE,
6666,002,1111,,V5,0,20011108,,,,,,,,0,1,Investigator key has changed from '6666' to
'5555',DELETE,
```


第 5 章

InForm Performance Monitor

ユーティリティ

この章の内容

InForm Performance Monitor ユーティリティの概要	402
InForm Performance Monitor ユーティリティの起動	403
パフォーマンス統計の取得	404
InForm Performance Monitor ユーティリティの出力オプション	406
InForm Performance Monitor ユーティリティデータの管理	407
InForm Performance Monitor ユーティリティの使用例	408

InForm Performance Monitor ユーティリティの概要

InForm Performance Monitor ユーティリティは、InForm アプリケーションサーバが実行されているデスクトップ上で実行されます。InForm アプリケーションのセッションがアクティブな状態にあるとき、InForm Performance Monitor ユーティリティは InForm アプリケーションサーバの特定の種類の InForm アクティビティに関するメッセージを取得し、それらを画面に表示します。この画面ではメッセージの整理や保存が可能です。このユーティリティを使用して、以下を実行できます。

- スタディの開発および実施中に、パフォーマンスチューニングを支援する情報を提供する
- トラブルシューティングに備え、パフォーマンスデータを取得する

注：InForm Performance Monitor ユーティリティは、時間の経過とともにサーバパフォーマンスに影響を及ぼす可能性があるため、稼働環境での長時間にわたる使用には適していません。
Oracle では、5 ～ 10 分おきに InForm Performance Monitor ユーティリティを実行することを推奨しています。

InForm Performance Monitor ユーティリティの起動

- [スタート] > [プログラム] > [Oracle Health Sciences] > [InForm 6.0] > [InForm Performance Monitor] の順にクリックします。

[パフォーマンスモニタ] 画面が開きます。マシン上で 1 つまたは複数の InForm サーバが実行されている場合、InForm Performance Monitor ユーティリティは InForm アプリケーションサーバメッセージの記録を開始します。

注： InForm Performance Monitor ユーティリティは、スタディのパフォーマンスを評価および強化し、各アクティビティに必要な時間を定量化することによってサーバのアクティビティを検証する目的で、Oracle スタッフによって使用されます。「パフォーマンス統計の取得」(404ページ) に、出力されるメッセージに関する簡単な説明がありますが、InForm Performance Monitor ユーティリティの使用と解釈については Oracle のスタッフに一任されています。

パフォーマンス統計の取得

InForm アプリケーションセッションの各アクティビティについて、InForm Performance Monitor ユーティリティは以下のデータを取得し、[パフォーマンスモニタ] 画面に表示します。

- **トライアル ID** — InForm Performance Monitor ユーティリティメッセージに関連付けられているスタディを示します。メッセージがスタディ固有のものでない場合、含まれる値はゼロです。
- **要求 ID** — 数値が含まれています。同一の RequestID 値（ただしゼロを除く）をもつ InForm Performance Monitor ユーティリティメッセージは、ある共通のアクティビティに関連付けられています。
- **時間 (分)** — ユーザがリクエストを発行してからサーバが応答するまでの経過時間（単位：ミリ秒）です。
- **トランザクションの日時** — メッセージが送信された日時（グリニッジ標準時 (GMT)）です。
- **タグ** — メッセージに関連付けられているアクティビティを記述したテキスト文字列です。
- **メッセージ** — アクティビティに関連する詳細情報を提供するテキスト文字列です。

特定のサブシステムのメッセージの表示

- 1 [表示] > [オプション] の順に選択します。
[パフォーマンスモニタオプション] 画面が表示されます。
- 2 [表示するメッセージ] セクションで、指定したサブシステムのみからメッセージを受け取るためのフィルタを次の中から選択します。
 - 複数のサブシステムを選択するには、**Ctrl** キーを押したまま、クリックします。
 - すべてのオプションを選択するには、[すべて選択] をクリックします。
 - オプションのリストをクリアするには、[すべてクリア] をクリックします。

フィルタ	サブシステム
IIS Request Data	InForm ISAPI
Trial package	InForm スタディ MTS パッケージ (InForm アプリケーションサーバによる)
Rule package	InForm ルールパッケージ (InForm アプリケーションサーバによる)
CDD package	InForm CCD パッケージ (InForm アプリケーションサーバによる)
UDA	InForm データベース

フィルタ	サブシステム
ODBC	InForm ODBC
同期化	サポートされていません。
レポート	サポートされていません。
Listing	InForm リスティング
InForm service	InForm サービス
PFImport	InForm データインポートユーティリティ

- 3 **【最小時間】** セクションに、メッセージのリストに含めるのに要する最小時間(ミリ秒)を入力します。
- 4 **【OK】** をクリックします。

特定の InForm サーバのメッセージの表示

- 1 [表示] > [サーバ] の順に選択します。
[InForm サーバ選択リスト] ダイアログボックスが表示されます。
- 2 メッセージの取得先となるサーバを選択します。
 - 複数のサーバを選択するには、**Ctrl** キーを押したまま、クリックします。
 - すべてのサーバを選択するには、**【すべて選択】** をクリックします。
 - サーバのリストをクリアするには、**【すべてクリア】** をクリックします。
- 3 **【OK】** をクリックします。

InForm Performance Monitor ユーティリティの出力オプション

注：出力オプションを選択する前に、「パフォーマンス統計の取得」（404ページ）の説明に従い、取得する統計を選択します。

次に挙げる InForm Performance Monitor ユーティリティメッセージの出力オプションの中から、いずれかを選択します。

出力オプション	説明	操作
出力をオンラインで表示する	[パフォーマンスモニタ]画面にメッセージを表示します。 (デフォルト)	[出力] > [ウィンドウに出力]の順に選択
出力をファイルに取り込む	メッセージを [パフォーマンスモニタ] 画面に表示せずに (または表示することに加えて)、ファイルに送信します。	[出力] > [ファイルにストリーム出力] の順に選択
パフォーマンスログに保存する	カンマ区切りテキストのパフォーマンスログを、指定したファイルに保存します。	[ファイル] > [名前を付けて保存] の順に選択

InForm Performance Monitor ユーティリティデータの管理

作業	操作
カラムのソート	カラムヘッダーバーをクリックします。
単一メッセージの選択	選択するメッセージをクリックします。
隣接する複数のメッセージの選択	Shift キーを押したまま、選択するメッセージをクリックします。
隣接しない複数のメッセージの選択	Ctrl キーを押したまま、選択するメッセージをクリックします。
すべてのメッセージの選択	[編集] > [すべて選択] の順に選択します。
選択したメッセージのコピー	[編集] > [コピー] の順に選択するか、ツールバーの [コピー] をクリックします。
選択したメッセージの削除	[編集] > [削除] の順に選択するか、ツールバーの [削除] ボタンをクリックします。
すべてのメッセージのクリア	[編集] > [リセット] の順に選択します。

InForm Performance Monitor ユーティリティの使用例

ルールスクリプトの効率のテスト

スタディ用に一連のルールを開発し、症例データをテストデータベースに追加した後であれば、InForm Performance Monitor ユーティリティを使用して、以下を実行できます。

- ルールが、ルールコンテキストに基づいて予定通りに実行されているかどうかを検証する
- 通常よりも長いルール実行時間をチェックする
- トランザクションのサブミット時間に、ルール処理が占めている割合を判断する

ルール処理データの取得

- 1 [パフォーマンスモニタ] 画面で、[表示] > [オプション] の順に選択します。
[パフォーマンスモニターオプション] ダイアログボックスが表示されます。
- 2 ルールデータを取得するメッセージフィルタを設定するために、ルールパッケージを除くすべてのフィルタをクリアします。
- 3 **[OK]** をクリックします。
- 4 スタディに使用する InForm アプリケーションを起動します。
- 5 ルールに関連付けられているアイテムに移動します。
- 6 アイテムを編集します。**[サブミット]** をクリックしないでください。
- 7 [パフォーマンスモニタ] 画面で **[編集]>[リセット]** の順にクリックし、表示をクリアします。
- 8 InForm アプリケーションで、**[サブミット]** をクリックします。
- 9 [パフォーマンスモニタ] 画面に戻り、メッセージをチェックします。

以下の内容を示すメッセージが表示されます。

- ルールが予定どおりに実行されたこと
- ルールコンテキストが存在したこと

InForm Performance Monitor ユーティリティは、依存性のチェックに要した時間、ルールと計算の実行に要した時間、およびフォームでルール処理に要した合計時間を記録します。

注：メッセージをリクエスト時間の長さでソートするには、**[時間 (分)]** カラムヘッダーをクリックします。複数のルールに関するデータを収集する際、このソートによって、通常よりも長い処理時間を要したルールを特定することができます。

SQL クエリパフォーマンスのレビュー

スタディデータベースにデータが蓄積されていった結果、応答時間が遅くなった場合に

は、InForm Performance Monitor ユーティリティを使用して長い SQL クエリを分割することができます。長い SQL クエリが見つかった場合、以下のいずれかを実行します。

- テーブルスペースの割り当てを変更する
- 特定のデータベーステーブルのインデックス作成方法を改善する
- データベースで統計更新を実行する

注：InForm Performance Monitor ユーティリティは、時間の経過とともにサーバパフォーマンスに影響を及ぼす可能性があるため、稼働環境での長時間にわたる使用には適していません。**Oracle** では、5 ～ 10 分おきに **InForm Performance Monitor ユーティリティ**を実行することを推奨しています。

SQL クエリパフォーマンスデータの取得

- 1 [パフォーマンスモニタ] 画面で、[表示] > [オプション] の順に選択します。
[パフォーマンスモニターオプション] ダイアログボックスが表示されます。
- 2 [UDA] を除くすべてのフィルタをクリアして、ルールデータを取得するメッセージフィルタを設定します。
- 3 [OK] をクリックします。
- 4 InForm アプリケーションを起動します。
- 5 クエリに関連付けられているアイテムに移動します。
- 6 アイテムを編集します。[サブミット] をクリックしないでください。
- 7 [パフォーマンスモニタ] 画面で [編集] > [リセット] の順に選択し、表示をクリアします。
- 8 [出力] > [ファイルにストリーム出力] の順に選択して InForm Performance Monitor ユーティリティにメッセージをファイルに取り込むように指示し、それらのメッセージを保存するファイルの名前を指定します。
- 9 InForm アプリケーションで、[サブミット] をクリックします。
- 10 [パフォーマンスモニタ] 画面に戻り、メッセージをチェックします。

Oracle サポートから、保存したメッセージファイルを送信するように要請があれば、それに従ってください。

注： リクエストの実行に要する時間を基にメッセージをフィルタ処理するには、[Performance Monitor オプション] ダイアログの [最小時間] セクションを使用します。**InForm Performance Monitor ユーティリティ**は、指定した時間（ミリ秒）を上回る時間を処理に必要としたリクエストに関するメッセージのみを表示します。

通常の要求時刻を判断するには、まず [最小時間] セクションの [次より短いアクションを破棄する] フィールドに 0（ゼロ）を入力し、その後 [パフォーマンスモニタ] 画面の [時間] フィールドのデータを、取得されたメッセージごとに比較します。

第 6 章

InForm Report Folder Maintenance

ユーティリティ

この章の内容

InForm Report Folder Maintenance ユーティリティの概要	412
複数のスタディまたはスポンサーのフォルダ構造.....	413
複数スポンサーまたは複数スタディ用のフォルダ構造のセットアップ	418
レポートフォルダのコピー	421

InForm Report Folder Maintenance ユーティリティの概要

InForm Report Folder Maintenance ユーティリティツールは、以下を実行できる Windows アプリケーションです。

- レポーティングと分析フォルダとその内容を目的のフォルダにコピーする
- 1 つのレポーティングと分析サーバを次のいずれかに使用している場合、フォルダ構造を定義および作成する：
 - 複数のスタディ
 - 複数のスポンサー
 - 複数スポンサー内の複数スタディ
- 目的の場所にコピーされている InForm 標準レポートのドリルダウンレポートへのリンクを更新する
- 目的の場所にある新しいレポーティングと分析パッケージに、コピーしたレポートを関連付ける

注：これには、保存されたレポートと標準レポート定義も含まれます。

InForm Report Folder Maintenance ユーティリティを実行するには、システム構成の変更権限が必要です。

InForm Report Folder Maintenance ユーティリティは、レポーティングと分析のインストールおよび構成の一部として、レポーティングと分析 (Cognos 8 Business Intelligence) サーバにインストールされています。次のパスにあります。

¥c8¥bin¥PFMTRSetupUtil.exe

InForm Report Folder Maintenance ユーティリティは、レポートとレポート定義のみをコピーします。InForm トライアルマネジメント、または公開されているスタディ固有の臨床パッケージが含まれるフォルダはコピーしません。レポートとレポートパッケージ間の関連付けの詳細については、「**レポートとパッケージの関連付け**」(419ページ)を参照してください。

注：この章で示した手順を実行する前に、レポーティングと分析でスタディを設定する必要があります。詳細については、『スタディとレポート設定ガイド』を参照してください。

複数のスタディまたはスポンサーのフォルダ構造

フォルダ構成の初期セットアップ

- 1 標準レポートアーカイブをインポートします。このアーカイブには、InForm トライアルマネジメントのほか、すべての標準レポートが含まれています。
- 2 臨床レポートに使用するスタディ固有の臨床パッケージを公開します。

これらの手順を完了すると、標準レポートおよびレポートリングパッケージ用のフォルダが InForm Report Folder Maintenance ユーティリティによってレポートリングサーバに作成されます。これらのフォルダは、レポートリングと分析ポータルの [共有フォルダ] タブに表示されます。最初に 1 つのスタディをセットアップすると、以下のフォルダが表示されます。

- CRF レポート
- アイテムレポート
- クエリのレポート
- 症例のレポート
- 監査証跡レポート
- InForm トライアル管理パッケージ
- `<study_name>`臨床パッケージ

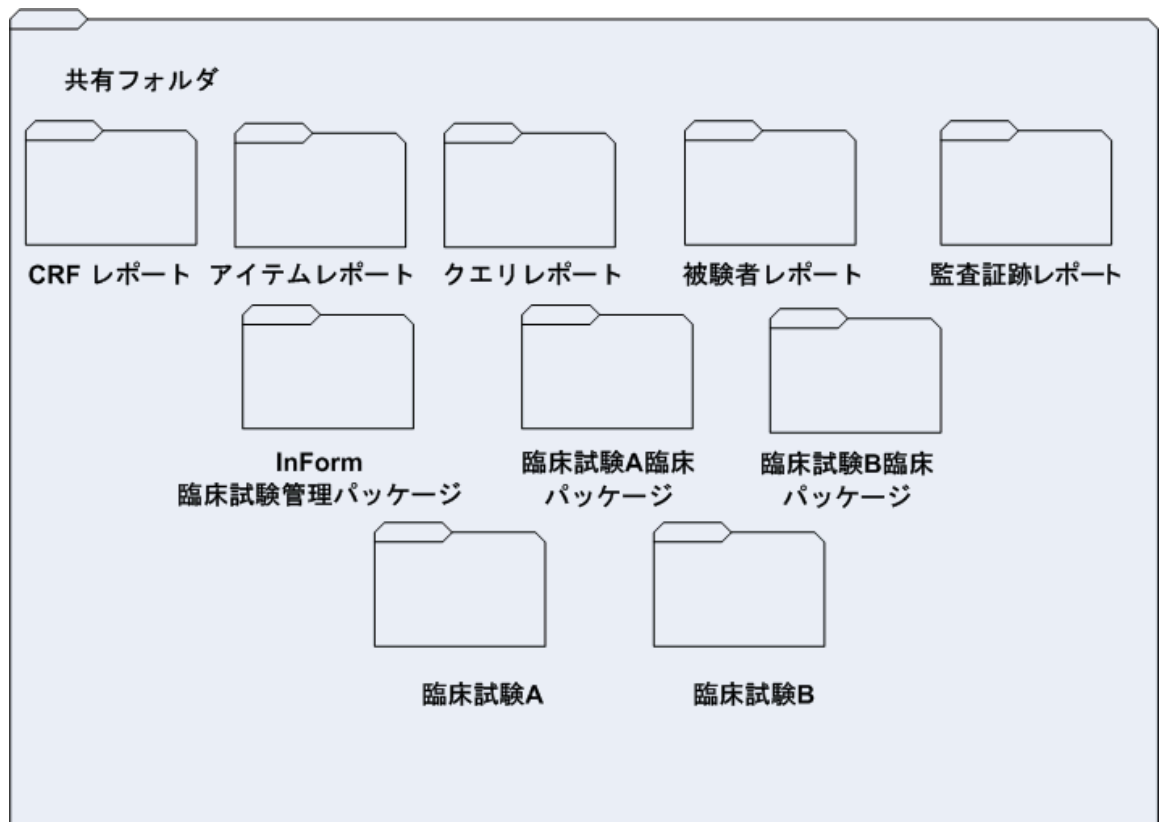
複数スタディのフォルダ構造

単一のスポンサーのスタディを複数セットアップする場合、スタディごとに別のフォルダを作成することを検討します。この処理は、レポートिंगと分析のセットアップ時に実行する PFRInit ユーティリティによって自動的に行われます。

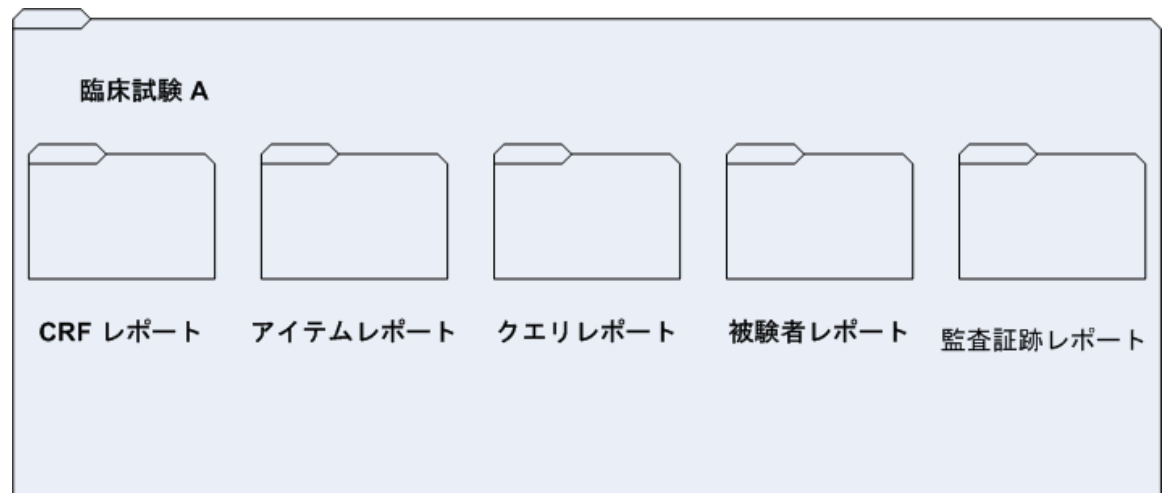
下の図は、2 つのスタディ (Study A と Study B) 用のフォルダを格納しているレポートングと分析ポータルの共有フォルダポータルの内容を示したものです。

この設定では、次の 3 つの異なるパッケージを使用しています。

- **InForm トライアルマネジメント** — Study_A と Study_B が共有。
- **Study__A_Clinical package** — Study_A のみに使用。
- **Study__B_Clinical package** — Study_B のみに使用。



各スタディのフォルダは、デフォルトのレポートिंगフォルダ構成でセットアップされます。



この構造では、指定したレポートについて、プロンプト値やレポートスケジュールなどスタディ固有のプロパティを保存できます。

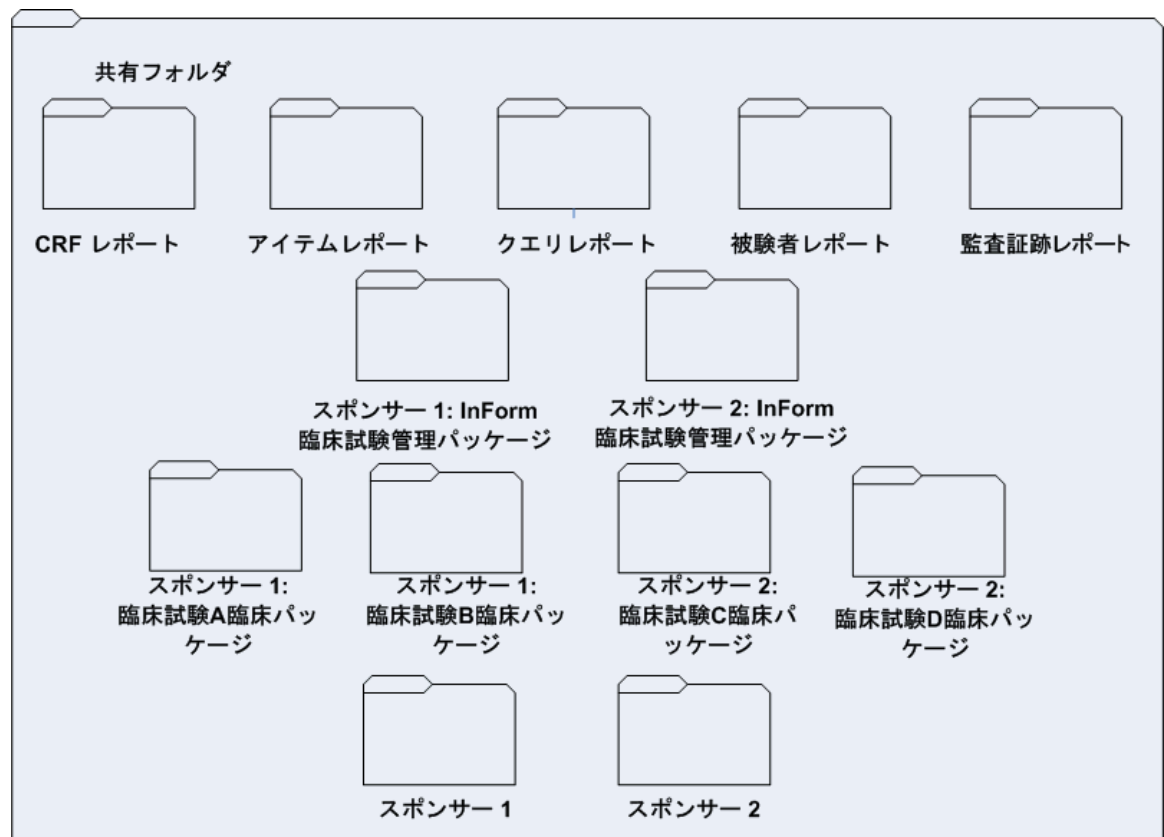
複数スポンサーと複数スタディのフォルダ構造

1 つのレポートニングサーバに複数の依頼者用フォルダをセットアップする場合、[共有フォルダ] タブの下層に各依頼者用のサブフォルダを作成することを検討します。これは、InForm Report Folder Maintenance ユーティリティを使用して構成のすべてのステップを実行する場合にのみ可能です。

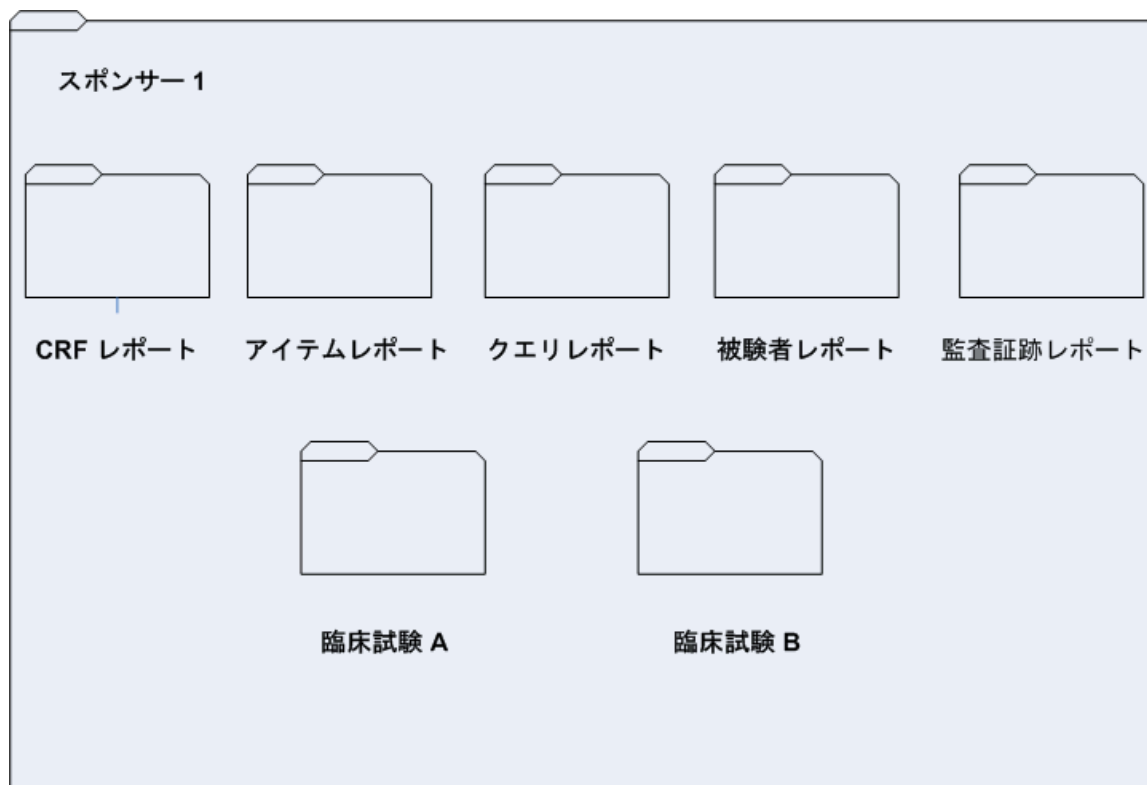
下の図は、2 つの異なるスタディをホストしている 2 人の異なる依頼者用のフォルダを格納しているレポートニングと分析ポータルの共有フォルダの内容を例示したものです。

具体的には、次のような構成になっています。

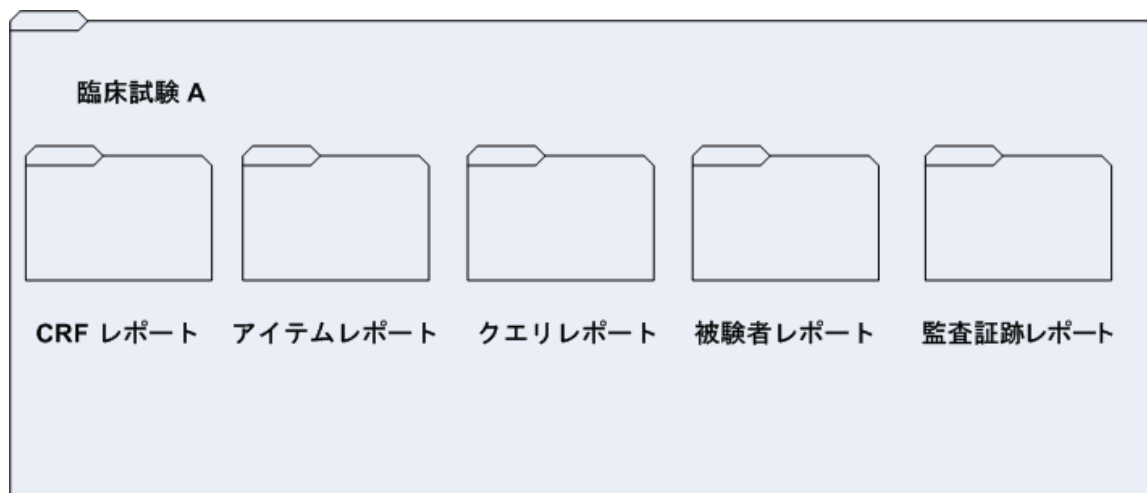
- 4 つの臨床パッケージ：Sponsor 1 に 2 つと Sponsor 2 に 2 つ
- 2 つの InForm トライアルマネジメント：Sponsor 1 に 1 つと Sponsor 2 に 1 つ



各スポンサーのフォルダは、1 つのスタディのデフォルトのレポートフォルダ構成でセットアップされます。次の図は、Sponsor 1 フォルダの内容を例示したものです。



Study_A フォルダには、デフォルトの標準レポートフォルダがすべて含まれています。



複数スポンサーまたは複数スタディ用のフォルダ構造のセットアップ

レポートパッケージのセットアップ

すべてのレポートパッケージは、共有フォルダレベルにのみ存在している必要があります。サブフォルダにはパッケージを追加できません。このため、複数の依頼者やスタディに複数のパッケージを使用する場合、レポーティングと分析ポータルの [共有フォルダ] タブには、各パッケージ用のフォルダが表示されます。

スタディ固有の臨床パッケージ

スタディごとに、スタディ固有の一意の臨床パッケージを使用します。スタディ固有の臨床パッケージを公開する方法の詳細は、『スタディとレポート設定ガイド』を参照してください。

InForm トライアルマネジメント


依頼者が同じスタディは、InForm トライアルマネジメントを共有できます。これにより、パッケージに加えられたすべての改訂または更新が、そのスポンサーのすべてのスタディに自動的に適用されます。

InForm トライアルマネジメントをインポートするには、管理モデルとレポート展開アーカイブをインポートします。このアーカイブをレポーティングと分析インストールの一部としてインポートしている場合、標準レポートフォルダと InForm トライアルマネジメントの両方がインポートされています。

ただし、個々のスタディのパッケージを改訂するには、スタディごとにスタディ管理パッケージをインポートする必要があります。

複数の異なる依頼者または複数の異なるスタディ用の InForm トライアルマネジメントをインポートするには、以下の手順に従います。

- 1 InForm システム管理者として、スタディにログインします。
- 2 [レポート] をクリックします。
- 3 [起動] > [レポート管理] を選択します。
[管理] ページが表示されます。
- 4 [コンフィギュレーション] タブをクリックします。
- 5 [コンテンツ管理] をクリックします。
- 6 Op パッケージとレポートエントリで、[プロパティを設定] をクリックします。
- 7 [インポート] タブをクリックします。
- 8 [InForm トライアル管理] チェックボックスをオンにします。他のすべてのオプションが非選択状態になっていることを確認します。

- 9 [InForm トライアル管理] の横にある鉛筆のアイコンをクリックします。
- 10 パッケージを使用する依頼者またはスタディを特定しやすいように、InForm トライアルマネジメントの名前を変更します。
- 11 [OK] をクリックします。
- 12 [戻る] をクリックします。
[管理] ページが表示されます。
- 13 作成したパッケージについて、[オプションを指定して実行 - [パッケージ名]] アイコン () をクリックします。
[オプションを指定して実行 - [パッケージ名]] ページが表示されます。
- 14 [時刻] セクションで、インポートを実行する時刻を指定します。
- 15 [Report Specification upgrade (レポート仕様のアップグレード)] セクションで、以下のいずれかを選択します。
 - すべてのレポート仕様を最新バージョンにアップグレードする
 - 既存のレポート仕様バージョンを保持する
- 16 [実行] をクリックします。

複数スタディまたはスポンサーの新規フォルダの作成

レポートとパッケージの関連付け

各レポートは、それを作成するために使用されたパッケージ (InForm トライアルマネジメントまたはスタディ固有のパッケージ) に関連付けられています。

InForm Report Folder Maintenance ユーティリティを使用してレポートをコピーする場合、必要に応じて使用するパッケージ名を指定できます。新しい場所にコピーしたレポートは、スタディまたはそのレポートを処理するスポンサーが使用するパッケージに関連付けられている必要があります。

InForm トライアルマネジメントで作成したレポートに関する留意点:

- **複数のスタディ、単独のスポンサーの場合** — 新規パッケージ名を指定する必要はありません。InForm トライアルマネジメントは、すべてのスタディとともにインストールされます。
- **複数のスタディ、複数のスポンサーの場合** — 各スポンサーは、InForm トライアルマネジメントのインスタンスを提供する必要があります。したがって、目的の場所にコピーしたレポートについて、新しいパッケージ名を指定する必要がある場合があります。

スタディ固有の臨床パッケージによって作成したレポートについては、目的の場所にコピーされたレポートに新しいパッケージ名を指定する必要があります。

レポートの検証

レポートをコピーした場合、InForm Report Folder Maintenance ユーティリティはコピーさ

れたレポートと関連付けられているパッケージとの照合チェックを実行します。したがって、コピー操作を始める前にパッケージが存在する必要があります。詳細については、「**レポートパッケージのセットアップ**」(418ページ)を参照してください。

InForm Report Folder Maintenance ユーティリティは、エラーが発生すると停止します。エラーを修正し、目的のフォルダ内にあるすべてのオブジェクトを削除し、ユーティリティを再び実行してすべてのレポートをコピーし直す必要があります。たとえば、臨床レポートエレメントを含むレポート定義はスタディ間でコピーできますが、レポートイングと分析データベースの臨床部分のスキーマは各スタディに固有なものです。したがって、スタディごとに生成される臨床パッケージも固有です。臨床レポートに、コピー先のパッケージに存在しないレポートエレメントが含まれていると、そのレポートはコピー後に検証できません。

レポートのコピーに関する注意事項のまとめ

コピー元パッケージ	レポートに含まれるデータのタイプ	新しいパッケージ名が必要か	コピー先でのレポートの変更は必要か
スタディ固有の臨床パッケージ	スタディ管理データのみ	はい*	いいえ
スタディ固有の臨床パッケージ	臨床データ、または臨床とスタディ管理の混合データ	はい*	臨床パッケージとそのエレメントに依存します。レポートを検証して目的の場所にコピーが実行できるように、レポートを変更する必要がある場合があります。
単独のスポンサーと複数のスタディ			
InForm トライアルマネジメント	スタディ管理データのみ	いいえ	いいえ
複数のスポンサーと複数のスタディ			
InForm トライアルマネジメント	スタディ管理データのみ	はい。各スポンサーフォルダに、専用の InForm トライアルマネジメントが存在している必要があります。	いいえ

*フォルダをコピーする前に、パッケージが存在する必要があります。

レポートフォルダのコピー

- 1 レポーティングと分析サーバで、次のファイルをダブルクリックします。

\crn\bin\PFMTRSetupUtil.exe

InForm Report Folder Maintenance 画面が表示されます。

- 2 以下の情報を入力してください。

- **[ユーザ名]** と **[パスワード]** — レポーティングと分析サーバへのログインに使用するユーザ名とパスワード
- **LDAP 名前空間** — Sun One Directory Server での認証に使用される LDAP ネームスペースの名前
- **コピー元** — **[標準レポートフォルダ]** をクリックして標準レポートフォルダをコピーするか、**[カスタムフォルダ]** をクリックしてカスタムレポートフォルダをコピーします。
- **Path** — コピーする既存のフォルダ構成のパスを入力します。ここではフォルダ名を入力しないようにします。
ソースフォルダのプロパティから、ソースパスをコピーできます。
- **Folder prefix** — コピーするフォルダを特定している接頭辞を指定します。たとえば、Study 1 に属するすべてのフォルダの接頭辞として S1 を使用する場合、S1 を **[フォルダのプレフィックス]** フィールドに指定して InForm Report Folder Maintenance ユーティリティに接頭辞 S1 を持つフォルダをコピーするように指示します。
- **Path** — コピー先フォルダのパスを入力します。ここではフォルダ名を入力しないようにします。
- **New Folder prefix** — コピー先フォルダ名の接頭辞を指定します。たとえば、Study 2 のフォルダ構成を作成する場合、新しいフォルダのプレフィックスとして S2 と入力します。
- **新規フォルダの名前** — コピー元フォルダの名前と異なる場合、コピー先フォルダの名前を入力します。新しいフォルダ名を指定（フォルダが新しく作成されます）するか、既存のフォルダを指定します。
- **パッケージ名** — スタディ固有のレポートへの関連付けが必要な新しいパッケージの名前を入力します。各スタディ用臨床パッケージは固有であるため、臨床（スタディ固有）データを含むレポートは、新しい場所にコピーするとき、新しい臨床パッケージに関連付ける必要があります。

注： コピーを実行する前に、パッケージが存在している必要があります。

- 3 **[作成]** をクリックします。

InForm Report Folder Maintenance ユーティリティが、新しい場所へのフォルダ構成のコピーを開始します。[ステータス]セクションに現在のステータスが表示され、コピーが完了するとそのことが通知されます。

注： ユーザーインターフェイスの [管理] > [システム構成] > [レポーティングユーザールート] フィールドで会社のトップレベルのレポートフォルダを指定できます。レポーティングと分析ポータルにログインすると、このフォルダが表示されます。

付 録 **A** インポート用 **XML** ファイルの サンプルデータ

この付録の内容

インポート用 XML ファイルのサンプルデータの概要	424
スクリーニングおよび症例登録データのインポート	425
新規症例の臨床データのインポート	426
既存の症例の臨床データの更新	427
[エントリの追加] アイテムセットの削除	428
[エントリの追加] アイテムセットの削除取り消し	429
予定外のビジットへのデータ追加	430
症例レコードの移動	431

インポート用 XML ファイルのサンプルデータの概要

MedML ファイルオプションのインポートファイルは、インポート中に実行する処理のタイプと、インポートデータのインポート先および値を指定するタグが含まれた XML ファイルです。サンプルをカットアンドペーストし、プレーンテキストファイルを作成できる任意のテキストエディタを使用してデータを編集することによって、独自のインポートファイルを作成することができます。

- **スクリーニングおよび症例登録データのインポート** (425ページ)
- **新規症例の臨床データのインポート** (426ページ)
- **既存の症例の臨床データの更新** (427ページ)
- **アイテムセットの削除** (428ページ)
- **削除したアイテムセットを元に戻す** (429ページ)
- **予定外のビジットへのデータ追加** (430ページ)
- **症例レコードの移動** (431ページ)

スクリーニングおよび症例登録データのインポート

下のサンプルファイルには、施設 PF の症例 XYZ に関するスクリーニングおよび登録データのインポートに必要なデータタグが含まれています。

```
<?xml version="1.0"?>
<CLINICALDATA>
<!--

Feature: Screen and Enroll
Description: This file sets up the user to test other XML features
-->

<!-- Screen Patient -->
<SCREEN SITEMNEMONIC="PF">
<DATA TAG="screen.0.patientinitials.patientinitials" VALUE="XYZ"/>
<DATA TAG="screen.0.eligible.eligible" VALUE="yes"/>
<DATA TAG="screen.0.datescreened.date" MONTH="1" DAY="6" YEAR="1999"/>
<DATA TAG="screen.0.dob.dob" MONTH="11" DAY="11" YEAR="1959"/>
</SCREEN>

<!-- Enroll Patient -->
<ENROLL PATIENTINITIALS="XYZ" SITEMNEMONIC="PF" PATIENTNUMBER="BK-XYZ"
ENROLL="TRUE">
<DATA TAG="consent.0.consentdate.date" MONTH="1" DAY="6" YEAR="1999"/>
<DATA TAG="consent.0.patientnumber.patientnumber" VALUE="BK-XYZ"/>
<DATA TAG="inclusion.0.age_inc.yesno" VALUE="1"/>
<DATA TAG="inclusion.0.hyper_inc.yesno" VALUE="1"/>
<DATA TAG="inclusion.0.understand_inc.yesno" VALUE="1"/>
<DATA TAG="inclusion.0.agree_inc.yesno" VALUE="1"/>
<DATA TAG="exclusion.0.secondary_ex.yesno" VALUE="0"/>
<DATA TAG="exclusion.0.malignant_ex.yesno" VALUE="0"/>
<DATA TAG="exclusion.0.allergyhistory_ex.yesno" VALUE="0"/>
<DATA TAG="exclusion.0.myocardial_ex.yesno" VALUE="0"/>
<DATA TAG="exclusion.0.monitor_ex.yesno" VALUE="0"/>
</ENROLL>
</CLINICALDATA>
```

新規症例の臨床データのインポート

下のサンプルファイルには、施設 PF の症例 XYZ に関する臨床データのインポートに必要なデータタグが含まれています。

```
<?xml version="1.0"?>
<CLINICALDATA>

<!--
Feature: Form and Item comments
Description: This example demonstrates basic form and item comments
Requirements: PF_XYZ-Enroll.xml
-->

<!-- Demographics form -->
<PATIENTDATA
  PATIENTINITIALS="XYZ"
  SITEMNEMONIC="PF"
  FORMSETREFNAME="Visit1"
  FORMREFNAME="DEM"
  COMMENT="This is the Demographics Form Comment">
  <DATA TAG="DEM.0.GENDER.GENDERRADIO" VALUE="1" COMMENT="Gender Comment"/>
  <DATA TAG="DEM.0.DEMDOB.dob" MONTH="2" DAY="14" YEAR="1961" COMMENT="Date Of Birth
  Comment"/>
  <DATA TAG="DEM.0.RACE.RACEGROUP" CHILDSELECTED="RACETEXT" COMMENT="Race Text
  Comment"/>
  <DATA TAG="DEM.0.RACE.RACEGROUP.RACETEXT" VALUE="10k" COMMENT="Race Text Value
  Comment"/>
  <DATA TAG="DEM.0.HEIGHT.PFHT_TC" VALUE="74" UNIT="Inches" COMMENT="Height
  Comment"/>
  <DATA TAG="DEM.0.WRISTCIRC.PFWC_TC" VALUE="6.0" UNIT="Inches" COMMENT="Height Unit
  Comment"/>
  <DATA TAG="DEM.0.FRAME.FRAME_CC" VALUE="1"/>
  <DATA TAG="SH.0.SMOKE.SMOKERADIO" VALUE="Y"/>
  <DATA TAG="SH.0.EVERSMOKED.SMOKERADIO" VALUE="N"/>
  <DATA TAG="SH.0.WHATSMOKED.SMOKEGROUPRADIO" CHILDSELECTED="SMOKEGROUP"/>
  <DATA TAG="SH.0.WHATSMOKED.SMOKEGROUPRADIO.SMOKEGROUP" VALUE="SMOKES"/>
  <DATA TAG="SH.0.WHATSMOKED.SMOKECHECKBOX" VALUE="cigarette,pipe"/>
  <DATA TAG="SH.0.HOWMUCHSMOKED.SMOKERADIO2" CHILDSELECTED="NUMTEXT"/>
  <DATA TAG="SH.0.HOWMUCHSMOKED.SMOKERADIO2.NUMTEXT" VALUE="10"/>
  <DATA TAG="SH.0.YRSSMOKED.SMOKERADIO2" VALUE="NDElement"/>
</PATIENTDATA>

</CLINICALDATA>
```

既存の症例の臨床データの更新

下のサンプルファイルには、症例に関する既存データの編集または削除に必要なデータタグが含まれています。

```
<?xml version="1.0"?>
<CLINICALDATA>

<!--
Feature: Edit an existing patient's form record
Description: This example demonstrates editing one item, then clearing an item's value
Requirements: The following must be run prior to this script
PF_XYZ-Enroll.xml
PF_XYZ-DEMANDcommentts.xml
-->

<!-- Demographics form -->
<EDITPATIENTDATA
PATIENTINITIALS="XYZ"
SITEMNEMONIC="PF"
FORMSETREFNAME="Visit1"
FORMREFNAME="DEM"
REASONOTHER="updated data">
<DATA TAG="DEM.0.HEIGHT.PFHT_TC" VALUE="75" UNIT="Inches" />
</EDITPATIENTDATA>

<!-- Demographics form -->
<EDITPATIENTDATA
PATIENTINITIALS="XYZ"
SITEMNEMONIC="PF"
FORMSETREFNAME="Visit1"
FORMREFNAME="DEM"
REASONOTHER="updated data">
<DATA TAG="DEM.0.WRISTCIRC.PFWC_TC" CLEARVALUE="TRUE"/>
</EDITPATIENTDATA>
</CLINICALDATA>
```

[エントリの追加] アイテムセットの削除

下のサンプルファイルには、既存の [エントリの追加] アイテムセットからのデータ削除に必要なデータタグが含まれています。

注： 繰り返しデータのアイテムセットは削除することも、削除したアイテムセットを元に戻すこともできません。

```
<?xml version="1.0"?>
<CLINICALDATA>

<!--
Feature:Delete an Itemset record (DELETEITEMSET tag)
Description:This example demonstrates to Delete an existing itemset record
Requirements:The following must be run prior to this script
PF_XYZ-Enroll.xml
PF_XYZ-ItemSet.xml
PF_XYZ-ItemSetDelete.xml
-->

<!-- non explicit itemset index (append to existing itemsets if exists) -->
<EDITPATIENTDATA
PATIENTINITIALS="XYZ"
SITEMNEMONIC="PF"
FORMSETREFNAME="Visit1"
FORMREFNAME="HH"
SECTIONNAME="PT"
ITEMSETNAME="PT"
DELETEITEMSET="TRUE"
ITEMSETINDEX="4"
REASONOTHER="test reason"
</EDITPATIENTDATA>

</CLINICALDATA>
```

[エントリの追加] アイテムセットの削除取り消し

下のサンプルファイルには、既存の [エントリの追加] アイテムセットから削除されたデータの復元に必要なデータタグが含まれています。

注： 繰り返しデータのアイテムセットは削除することも、削除したアイテムセットを元に戻すこともできません。

```
<?xml version="1.0"?>
<CLINICALDATA>

<!--
Feature:Undelete an Itemset record (UNDELETEITEMSET tag)
Description:This example demonstrates to Undelete a previously deleted itemset.
Requirements:The following must be run prior to this script
PF_XYZ-Enroll.xml
PF_XYZ-ItemSet.xml
PF_XYZ-ItemSetDelete.xml
-->

<EDITPATIENTDATA
PATIENTINITIALS="XYZ"
SITEMNEMONIC="PF"
FORMSETREFNAME="Visit1"
FORMREFNAME="HH"
SECTIONNAME="PT"
ITEMSETNAME="PT"
UNDELETEITEMSET="TRUE"
ITEMSETINDEX="4"
REASONOTHER="test reason"
</EDITPATIENTDATA>

</CLINICALDATA>
```

予定外のビジットへのデータ追加

下のサンプルファイルには、特定の症例の予定外のビジットにデータを追加する場合に必要なデータタグが含まれています。

```
<?xml version="1.0"?>
<CLINICALDATA>

<!--
Feature: Unscheduled Visits
Description: This example demonstrates the sequence of XML submits to use Unscheduled
Visits
Requirements: PF_XYZ-Enroll.xml
-->

<!-- DOV form -->
<PATIENTDATA
PATIENTINITIALS="XYZ" SITEMNEMONIC="PF"
FORMSETREFNAME="UnschVisit" FORMREFNAME="DOV" NEWUNSCHEDVISIT="TRUE">
<DATA TAG="DOV.0.DOV.DOV" MONTH="2" DAY="1" YEAR="1999"/>
</PATIENTDATA>

<!-- VitalSigns form -->
<PATIENTDATA
PATIENTINITIALS="XYZ" SITEMNEMONIC="PF" FORMSETINDEX="1"
FORMSETREFNAME="UnschVisit" FORMREFNAME="VS">
<DATA TAG="VS.0.DATEASSESS.COMMONDATE" MONTH="3" DAY="1" YEAR="1999"/>
<DATA TAG="VS.0.WEIGHT.PFWT_TC" VALUE="150" UNIT="Pound"/>
<DATA TAG="VS.0.TEMPTEXT.TEMPTEXT" VALUE="98.7" UNIT="Fahrenheit"/>
<DATA TAG="VS.0.BPREADING.BPREADINGGROUP.SYSTEXT" VALUE="130"/>
<DATA TAG="VS.0.BPREADING.BPREADINGGROUP.DIASTEXT" VALUE="85"/>
</PATIENTDATA>

<!-- DOV form -->
<PATIENTDATA
PATIENTINITIALS="XYZ" SITEMNEMONIC="PF"
FORMSETREFNAME="UnschVisit" FORMREFNAME="DOV" NEWUNSCHEDVISIT="TRUE">
<DATA TAG="DOV.0.DOV.DOV" MONTH="2" DAY="2" YEAR="1999"/>
</PATIENTDATA>

<!-- VitalSigns form -->
<PATIENTDATA
PATIENTINITIALS="XYZ" SITEMNEMONIC="PF" FORMSETINDEX="2"
FORMSETREFNAME="UnschVisit" FORMREFNAME="VS">
<DATA TAG="VS.0.DATEASSESS.COMMONDATE" MONTH="3" DAY="2" YEAR="1999"/>
<DATA TAG="VS.0.WEIGHT.PFWT_TC" VALUE="150" UNIT="Pound"/>
<DATA TAG="VS.0.TEMPTEXT.TEMPTEXT" VALUE="98.7" UNIT="Fahrenheit"/>
<DATA TAG="VS.0.BPREADING.BPREADINGGROUP.SYSTEXT" VALUE="130"/>
<DATA TAG="VS.0.BPREADING.BPREADINGGROUP.DIASTEXT" VALUE="85"/>
</PATIENTDATA>

</CLINICALDATA>
```

症例レコードの移動

下のサンプルは、複数の症例を移動するのに使用される、MedML ファイルのエレメントを示したものです。各ペアの PATIENTSITECHANGE 属性は、それぞれ 1 人の症例の現在の施設および移動先施設の情報を定義していることに注意してください。

```
<?xml version="1.0"?>
<CLINICALDATA>
  <PATIENTSITECHANGE REASON="new subject address">
    <NEWSITE SITEMNEMONIC="MCLEAN"/>
    <CURRENTSITE SITEMNEMONIC="PF" PATIENTNUMBER="1003"/>
  </PATIENTSITECHANGE>
```

<!--症例 1001 について、ファイルは新しい症例番号を指定しています。McLean 病院施設に同じ症例番号の患者がすでに存在しているためです。>

```
    <PATIENTSITECHANGE REASON="new subject address">
      <NEWSITE SITENAME="McLean Hospital" PATIENTNUMBER="1002"/>
      <CURRENTSITE SITEMNEMONIC="PF" PATIENTNUMBER="1001"/>
    </PATIENTSITECHANGE>
```

<!--DUPLICATEORDER 属性は、症例 DDD が Phase Forward 施設においてスクリーニングされる症例の中で、同一のイニシャルを持つ 2 人目の症例であることを示しています。>

```
    <PATIENTSITECHANGE REASON="new subject address">
      <NEWSITE SITEMNEMONIC="MCLEAN"/>
      <CURRENTSITE SITENAME="Phase Forward" PATIENTINITIALS="DDD"
        DUPLICATEORDER="2"/>
    </PATIENTSITECHANGE>
</CLINICALDATA>
```