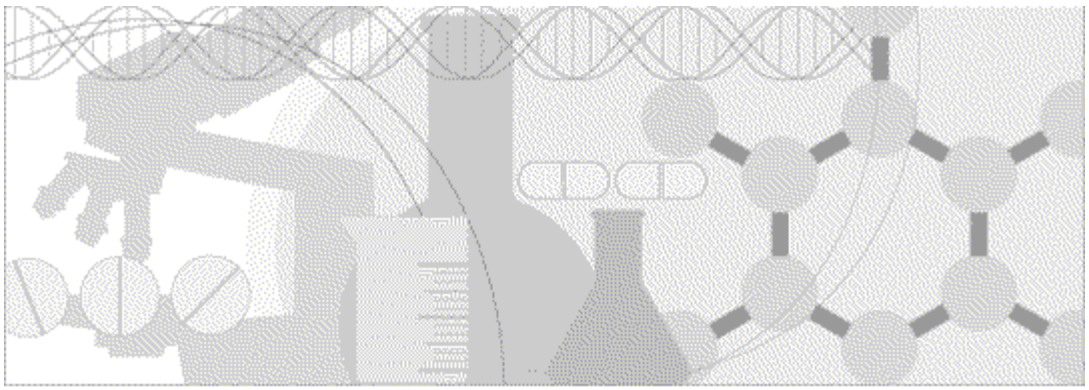
Utilities Guide

Oracle® Health Sciences InForm 6.1.1



ORACLE

Part number: E61328-01

Contents

CHAPTER 1

PFConsole utility

In this chapter

Overview of the PFConsole utility

The PFConsole utility consolidates all the activities of each utility within one window from which you can run:

- The InForm Data Import utility.
- The InForm Data Export utility.
- The MedML Installer utility.

Note: Oracle recommends that you use the PFConsole utility to run these utilities from the command line, and that you do not run them individually in the command line window.

Running the PFConsole utility from the command line

Use the following command to open the PFConsole utility and run an InForm utility.

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

where:

- *pfconsole* starts the PFConsole utility.
- <*application*> is one of the following:
  - *pfmminst*—MedML Installer utility.
  - *pfimport*—InForm Data Import utility.
  - *pfexport*—InForm Data Export utility.
- *-autorun* is a required parameter of the PFConsole utility.
- <*parameters*> are the variables for the individual utility.

For specific parameters for each utility, see:

- ***Running the MedML Installer utility from the command line*** (on page 32).
- ***Running the InForm Data Import utility from the command line*** (on page 165).
- ***Running the InForm Data Export utility from the command line*** (on page 177).

Note: All command line applications, such as the MedML Installer utility, the InForm Data Import utility, and the InForm Data Export utility use the default product locale specified during the InForm installation, or through PFAdmin commands.

Example

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

Using PFConsole utility within a script

If you are running several imports or exports, you can generate a script that runs the utilities in batch mode. To pause the script until the application completes and then moves to the next command, use the following command:

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

where:

<*pfconsole command line*> is the string of commands you specify when you run the PFConsole utility.

For more information, see ***Running the PFConsole utility from the command line*** (on page 3).

CHAPTER 2

MedML and the MedML Installer utility

In this chapter

MedML and the MedML schema

Creating an XML file

You can use any text editor, such as Notepad or TextPad, to create and modify XML files.

Note: Oracle suggests that you save the sample XML files that are delivered as part of the InForm installation media under a new name and edit those files. The files are available at: \\<*Installation_Directory*>\InForm \SampleStudy\Admin.

To create an XML file for MedML component definitions for administrative data:

1 Create a first line that contains the xml version number. This string must be lower case:

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

2 Add an opening and closing tag for the MedMLData component and include the appropriate version string as the value of the xmlns attribute:

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

3 Between the MedMLDATA start and end elements, enter definitions for the elements to install.

4 Install study data definitions in the following order:

a SequenceType

b Sequence

c Right

d User

e UserImage

f Site, including references to users who serve as site contacts, if applicable.

g SignCRF

Component definitions

An XML component definition is the basic building block of an XML file. Each component definition is made up of elements and attributes. Component definitions have a hierarchical relationship to each other (that is, a component can have child components).

XML elements

An element consists of the component name surrounded by angle brackets (< >).

When you define a compound component (a component with children), the parent component has both opening and closing elements, and the child components have an opening element that ends with a slash.

Components with no children have no separate closing element. Instead, the component definition ends with a slash and a closing angle bracket, following any attributes that you specify.

XML attributes

A set of attributes uniquely identifies the component and describes how the component appears online. In component definitions, the attributes follow the component name and precede the closing angle bracket or slash and closing angle bracket. The format of an attribute is:

ATTRIBUTE_NAME = "*attribute_value*", where

- ATTRIBUTE_NAME is the name of the attribute as defined in the MedML schema.
- *attribute_value* is the value assigned to the attribute in the component definition. Enclose this value in double quotation marks.

Child components

Child components, or children, in a component definition allow you to define controls in which multiple control definitions are nested to create a compound control. Using child components allows you to reuse previously defined components in multiple locations, by referring to them in child component definitions. For example, to define a drop-down list, you define a set of selections as components and then define the list control, including the individual selections in the list control definition as child components.

In the MedML schema, many child component names end with the letters REF (for example, Userref) to indicate their use as references to previously defined components that are nested within the definition of another component.

When you include child components in the definition of a compound control, the child component definitions follow the opening element of the parent component, after the parent component attributes.

Ordering component definitions

When you create a set of component definitions, you should work from the bottom up, and define child components before defining the components in which they are referenced.

The order in which component definitions are created is important when you load the component definitions into the database using the MedML Installer utility. When the MedML Installer utility processes a compound component, the child components referenced in the parent component definition must be present in the database. If the child components are not in the database, the MedML Installer utility cannot create the parent component.

You can organize study component definitions in any way that is convenient, as long as you follow the component order described in *Creating an XML file* (on page 6), and you can create multiple study component definition XML files. If you separate study component definitions into multiple files, load them in a sequence that maintains the prescribed component order.

Working with RSP files

Response files, which are text files with the RSP extension, list XML file names in the order in which they should be processed. You can create or modify RSP files using any text editor.

You might want to use or modify RSP files when you load multiple files at the same time, for example, when you:

- Load a subset of components from a study.
- Load base components from a library of standard study components.
- Prepare to deploy a study (the RSP file should reference all relevant files).
- Perform mid-study changes (the RSP file should include only those study components that have changed).

You can use an RSP file to load all the files for a study by specifying the paths for the XML files.

To create RSP files, from a Command Prompt window, in the directory containing the XML files, type:

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

This command creates an RSP file in the folder in which you executed the command. The resulting RSP file contains all the XML files in the current directory in alphabetical order.

Note: You might have to rearrange the order of the files in the RSP file if there are dependencies. To exclude an XML file from the RSP file, remove the file or comment it out in the RSP file using the pound sign (#).

RefNames

RefNames are names that are used to identify component definitions. In a Form definition window, the hierarchical representation of a form and its components shows each component by RefName. RefNames are also used to uniquely identify the path to a specific control when attaching a rule definition.

The following rules apply to the use of RefNames:

- RefNames can have a maximum of 63 characters.
- RefNames must be unique within a study and component type. For example, no two item definitions in a study could have the same RefName. However, a CRF and section can have the same RefName.
- RefNames are case-sensitive.
- After a definition containing a RefName is installed in the study database, you cannot change the RefName. After this point, to create a new RefName, you must create a new object.

- The following terms are disallowed for use in RefNames because they will cause a failure during the Reporting and Analysis database set up:
  - CD_COUNT
  - AFROWID
  - SUBJECTID
  - SITEID
  - STUDYVERSIONID
  - SUBJECTVISITID
  - SUBJECTVISITREV
  - VISITID
  - VISITINDEX
  - FORMID
  - FORMREV
  - FORMINDEX
  - SUBJECTINITIALS
  - SITEMNEMONIC
  - VISITMNEMONIC
  - FORMMNEMONIC
  - VISITORDER
  - SITENAME
  - SITECOUNTRY
  - SECTIONID
  - ITEMSETID
  - ITEMSETINDEX
  - ITEMSETIDX
  - DELETEDITEM
  - DELETEDFORM
  - FORMIDX

Reserved words for Reporting and Analysis

The reserved words below cannot be used for the following item properties:

- Short question.
- Default question or Question if a Short question is not provided, or is populated with a reserved word.

If you specify a reserved word for the item Default question or Question you must provide a Short question that is not a reserved word, or the clinical model in the Reporting and Analysis module fails to generate.

Note: You cannot use the English reserved words or Japanese reserved characters for any locale.

For example, to design an item for which a reserved word appears as the question on the form in the InForm application, specify the following:

- Question—Reserved word such as Subject Number.
- Default question or Short question—A word that is not on the reserved words list, such as Subj Number.

Subject Number appears as the question on the form, and Subj Number appears in the clinical model in the Reporting and Analysis module and as the itemset column header on the form in the InForm application. In addition, the clinical model is successfully generated.

A warning does not appear in the Central Designer application if a reserved word is used incorrectly.

English words	Japanese characters
SITECOUNTRY	施設の国名
SITENAME	施設名
SITEMNEMONIC	施設略称
SUBJECTINTIALS	症例イニシャル
SUBJECTNUMBERSTR	症例番号
VISITMNEMONIC	ビジット略称
VISITORDER	ビジットオーダー
VISITIDX	ビジットインデックス
FORMMNEMONIC	フォーム略称
FORMIDX	フォームインデックス
DELETED	削除フォーム
ITEMSETIDX	アイテムセットインデックス
DELETEDITEMSET	削除アイテムセット
DOV	ビジット日時

English words	Japanese characters
NOTDONE	未実施
CREATEDBYUSERID	基本データ
CREATEDDATETIME	その他のデータ
MODIFIEDBYUSERID	不完全データの理由
MODIFIEDDATETIME	内部ID
SITEID	施設
SUBJECTID	症例
VISITID	ビジット
VISITINDEX	フォーム
SUBJECTVISITID	セクション
FORMID	アイテム
FORMREV	ルール
FORMINDEX	クエリ
SECTIONID	コメント
ITEMSETID	定義済フィルタ
ITEMSETINDEX	すべての登録済症例
FORMASSOCID	脱落者を除いたすべての登録済症例
CD_COUNT	有効症例
Basic Data	クリニカルビジット
Additional Data	クリニカルフォーム
Incomplete Data Reasons	開始フォーム
Internal Id's	見込フォーム:予定ビジット
Sites	見込フォーム:予定ビジットまたは開始ビジット
Subjects	見込フォーム:現在までの予定ビジットまたは開始ビジット
Visits	見込フォーム:現在までの予定ビジットまたは開始フォーム
Forms	データ完了(クエリを含む)
Sections	データ完了(オープンクエリを含まない)
Items	クリーンデータ:完了かつSDV済み
Rules	

English words	Japanese characters
Queries	
Comments	
Predefined Filters	
All Enrolled Subjects	
All Enrolled Subjects except Dropped	
Active Subjects	
Clinical Visit	
Clinical Form	
Started Forms	
Expected Forms: Scheduled Visits	
Expected Forms: Scheduled or Started Visits	
Expected Forms: Scheduled or Started Visits to Date	
Expected Forms: Scheduled Visits or Started Forms to Date	
Data Complete (Queries Allowed)	
Data Complete, No Open Queries	
Data Clean: Complete and SV	
Site Country	
Site Name	
Site Mnemonic	
Subject Initials	
Subject Number	
Visit Mnemonic	
Visit Order	
Visit Index	
Form Mnemonic	
Form Index	
Deleted Form	
Itemset Index	
Deleted Itemset	
DOV	
Not Done	

UUIDs

Universal Unique Identifiers (UUIDs) ensure the unique identification of components across all servers, databases, and studies.

In most cases, assignment of a UUID attribute is optional. However, for certain purposes, the InForm application requires the use of specific, well-known UUIDs. For example, to allow users to define subject numbers, rather than allowing the InForm application to generate them automatically, you must use well-known UUIDs for the formset, form, section, item, and text control definitions that make up the specification of the subject number data-entry field.

To change a UUID, you must install the change using the MedML Installer in nonstrict mode. For more information, see *About the MedML Installer window* (on page 29).

Note: The InForm application and the MedML Installer utility convert alphabetic characters in UUIDs to uppercase.

The following cases require a specific well-known UUID:

- Common forms
- Date Of Visit
- Enrollment forms
- Group types
- Subject ID form
- Subject Initials item
- Subject Number item
- Randomization
- RegDocs
- Repeating visits
- Screening forms
- Sequences
- Study Completion
- Visit Report

Common CRFs

Common CRFs can appear in multiple visits and have cumulative data that appears in each visit where the CRF is included.

Note: After you define a common CRF and add data for any subject to the CRF, you must not change the form type to a regular CRF during a study version change. Similarly, you must not change a regular CRF to a common CRF if the form includes subject data. Doing so will cause data loss.

Do not use a common form as the first form of a visit. When the form is reused in other visits, the Date of Visit control cannot capture the specific date of each visit.

If you need to create a regular CRF that captures the same data as an existing common CRF, create it as a separate FORM definition with a different RefName from the common CRF, and add it to the appropriate VISIT formsets in the StudyVersion definition.

To define a common CRF, use the following UUID in the definition of the COMMONCRF formset in which the CRF is included:

Component	UUID
Common Form	9d6bbc5d-5811-11d2-8065- 00a0c9af7674

Date Of Visit

The Date of Visit item appears on the first form of every visit. To set up a scheduled or unscheduled visit, you must use the following UUIDs to create a section, item, and datetime control in which to capture the date and time of the visit.

Form component	UUID
DateTimeControl	BD991BC0-B0A4-11D2-80E3-00A0C9AF7674
Item	BD991BBF-B0A4-11D2-80E3-00A0C9AF7674
Section	BD991BBE-B0A4-11D2-80E3-00A0C9AF7674

Note: Do not use a common CRF as the first form of a visit. When the form is reused in other visits, the Date of Visit control cannot capture the specific date of each visit.

Enrollment forms

Component	UUID
Enrollment formset	d882ce3a-0f42-11d2-a419- 00a0c963e0ac
Enrollment form	d882ce3b-0f42-11d2-a419- 00a0c963e0ac
Enrollment form section that contains the Subject Number data entry item	abcfa388-223a-11d2-a426- 00a0c963e0ac

Additionally, if the Enrollment form includes an editable Subject Number item, that item requires a well-known UUID. For more information, see ***Subject Number item*** (on page 16).

Group types

The GROUPTYPE element defines each of the four types of administrative groups used in the InForm application: query, rights, signature, and site. The following UUIDs are required in the definitions of GroupTypes:

Component	UUID
Query	AC44A6E1-112E-11d2-8BED-0060082DE9D5
Rights	FA3C6201-112E-11d2-8BED-0060082DE9D5
Signature	002E58C1-112F-11d2-8BED-0060082DE9D5
Site	A4D7B9A1-112E-11d2-8BED-0060082DE9D5

Subject ID form

Use the Subject ID form to allow users to change the Subject ID after a subject is fully enrolled. The Subject ID form requires the following UUIDs:

Component	UUID
Visit containing Subject ID form	03B0D5D8-7F2C-11D2-A728-00A0C977C64B
Subject ID form	06702B62-7ED6-11D2-A728-00A0C977C64B
Subject ID section	3D25EE4B-7F1B-11D2-A728-00A0C977C64B
Either or both of the following items: • The Subject Number item used on the Enrollment form • A Change Initials item	Subject Number item: 3D25EE4C-7F1B- 11D2-A728-00A0C977C64B Change Initials item: D959FE72-7F1C- 11D2-A728-00A0C977C64B

Component	UUID
Either or both of the following text box controls: • Subject Number text box control, if the section contains the Subject Number item used on the Enrollment form • Change Initials text box control, if the section contains a Change Initials item	Subject Number control: 433DAFF6- 7F1C-11D2-A728-00A0C977C64B Change Initials control: 433DAFF7- 7F1C-11D2-A728-00A0C977C64B

Subject Initials item

The Subject Initials item definition is required on the Screening form. Use the following UUIDs to define the Subject Initials item:

Form component	UUID
Subject Initials item	aeb64f16-127c-11d2-a41c-00a0c963e0ac
Subject Initials text control	aeb64f17-127c-11d2-a41c-00a0c963e0ac
Date of Birth item	96cae359-126c-11d2-a41c- 00a0c963e0ac
Date of Birth control	40aee712-217c-11d2-a425- 00a0c963e0ac
Date Screened item	96cae356-126c-11d2-a41c- 00a0c963e0ac
Date Screened control	96cae357-126c-11d2-a41c- 00a0c963e0ac

Subject Number item

The Subject Number item definition is an optional item on the Enrollment form. If you allow users to edit the Subject Number, use the following required UUIDs to define the Subject Number item:

Form component	UUID
Subject Number item	3d25ee4c-7f1b-11d2-a728- 00a0c977c64b
Subject Number text control	433daff6-7f1c-11d2-a728-00a0c977c64b
Enrollment form section that contains the Subject Number data-entry item	abcfa388-223a-11d2-a426- 00a0c963e0ac

Randomization

When setting up the InForm software to generate randomization numbers, you must include a well-known control, item, and section on the CRF where the randomization number appears. In these component definitions, use the following UUIDs:

Form component	UUID
RANDOMIZATION calculated control	DC2EB0BF-4F12-11d2-9319- 00A0C9769A13
DRUGKIT item	52AF1207-4F13-11d2-9319- 00A0C9769A13
DRUGKITSECTION section	C0482B37-4F13-11d2-9319- 00A0C9769A13

Reg Docs

The Reg Docs item is required on the Regulatory Document CRF. The definition of Reg Docs requires the following UUIDs:

Component	UUID
Regulatory Document formset	PF_UUID_REGDOCS_FORMSET
Regulatory Document form	PF_UUID_REGDOCS_FORM
Regulatory Document section	PF_UUID_REGDOCS_SECTION

Repeating visits

A FORMSET definition with the **REPEATING** attribute set to true allows you to define an unscheduled visit with its own set of CRFs. To set up a visit, you must use the following UUIDs to create a section, item, and datetime control in which to capture the date and time of the visit:

Form component	UUID
DateTimeControl	BD991BC0-B0A4-11D2-80E3-00A0C9AF7674
Item	BD991BBF-B0A4-11D2-80E3-00A0C9AF7674
Section	BD991BBE-B0A4-11D2-80E3-00A0C9AF7674

Screening CRFs

The definition of a Screening CRF requires the following well-known UUIDs:

Form component	UUID
Screening formset	d882ce38-0f42-11d2-a419- 00a0c963e0ac
Screening form	d882ce39-0f42-11d2-a419- 00a0c963e0ac
Screening form section that contains the Subject Initials, Date of Birth, Date Screened	96cae354-126c-11d2-a41c- 00a0c963e0ac

Sequences

The InForm software automatically generates enrollment, query, randomization, and screening numbers as a study progresses. These numbers are generated according to a sequence established in a Sequence definition. The required UUIDs define the enrollment, query, randomization, and screening number sequences:

Form component	UUID
Enrollment	eb75b898-078b-11d2-a417-00a0c963e0ac
Randomization	4F4A0246-5009-11d2-931C-00A0C9769A13
Screening	f7f1b3b8-0b5c-11d2-a418-00a0c963e0ac

Study Completion

The Study Completion CRF records the last time a subject was seen and whether the subject completed the study. If the subject did not complete the study, the Study Completion CRF records the reason for not completing. Use the following UUIDs in the definition of a Study Completion CRF.

Note: The UUIDs in the Study Completion CRF are independent of study versions and apply to all subjects in a study; therefore, you cannot change study completion metadata by creating a new study version.

Component	UUID
Visit in which the Study Completion CRF is included	F4699051-69E2-11D2-8FB5-00A0C977C66A
Study Completion CRF	7314A6A5-316E-11d2-8F9A-00A0C977C66A
Control to indicate completion status	PF_SC_COMPLETECTL

Component	UUID
Element objects defining the values and display text for study completion or noncompletion	Both of the following element objects must be present in the study definition: • PF_SC_STUDY_COMPLETE — Indicates that the subject completed the study. • PF_SC_STUDY_INCOMPLETE — Indicates that the subject did not complete the study.
Control to indicate the reason the subject dropped out of the study	PF_SC_REASONCTL
Text resources mapping the internal values of noncompletion reasons to the text of column headings in the Reporting and Analysis module	PF_SC_REASONCTL_*internal_reason_value* *Internal_reason_value* is one of the following: • The Value property defined for an element object in a simple or pull-down control included in the control specifying noncompletion reason. • The Selection Value property defined for any other type of subordinate control included in the control specifying noncompletion reason. • The default value for a subordinate control for which no Selection Value property has been defined. The default value is assigned by the InForm application and is in the format !pf! *control_DBUID_path*.

Visit Report

The Visit Report item is required on the Visit Report form. The definition of Visit Report requires the following UUID:

Component	UUID
Visit Report formset	PF_UUID_VISITREPORT_FORMSET
Visit Report form	PF_UUID_VISITREPORT_FORM

Global tags

<!– –> (Comment)

The comment element allows you to insert a comment in an XML file by enclosing it with angle brackets. To begin a comment, insert the characters <!-- before the comment text; to conclude a comment, follow the comment text with the characters -->.

You cannot enter a comment within another element.

HTML formatting tags

The InForm application supports the following formatting tags, which you can use in any text-based study component definitions; for example, study protocols, CRF Help, CRF questions and notes, and rule help.

Because angle brackets (greater than and less than symbols) are disallowed by the MedML Installer utility, you must use the HTML escape character equivalents when installing data from a MedML file to the InForm application.

For more information, see ***HTML special characters*** (on page 21) and ***Disallowed characters*** (on page 23).

For this formatting	Use these equivalent tags	As substitutes for these HTML tags
Bold text	<B>	<B>
	</B>	</B>
Line break	 	
Centering text an equal distance from the left and right edges of the document	<CENTER>	<CENTER>
	</CENTER>	</CENTER>
Italic text	<I>	<I>
	</I>	</I>
List items	<LI>	<LI>
	</LI>	</LI>
Ordered (numbered) lists	<OL>	<OL>
	</OL>	</OL>
Paragraphs	<P>	<P>
	</P>	</P>
Preformatted plain text; for example, computer output	<PRE>	<PRE>
	</PRE>	</PRE>
Strikethrough text	<S>	<S>
	</S>	</S>
Strikethrough text	<STRIKE>	<STRIKE>
	</STRIKE>	</STRIKE>
Subscript text	<SUB>	<SUB>
	</SUB>	</SUB>
Superscript text	<SUP>	<SUP>
	</SUP>	</SUP>

For this formatting	Use these equivalent tags	As substitutes for these HTML tags
Monospace font	<TT>	<TT>
	</TT>	</TT>
Underlined text	<U>	<U>
	</U>	</U>
Unordered (bulleted) lists	<UL>	<UL>
	</UL>	</UL>

HTML special characters

The HTML specification includes character sequences for specifying special characters. When you include HTML special characters in a study component definition file, the MedML Installer utility passes the characters along to the database, and they are retrieved and processed by the forms rendering component of the InForm application.

The following special character definitions are supported by the PDF output format, in which the InForm Data Export utility exports printable CRFs.

Character	Entity Name	Numeric Code	Descriptive
"	quotation mark	&#34;	&quot;
&	ampersand	&#38;	&amp;
<	less-than sign	&#60;	&lt;
>	greater-than sign	&#62;	&gt;
	non-breaking space		
¡	inverted exclamation	&#161;	&iexcl;
¢	cent sign	&#162;	&cent;
£	pound sterling	&#163;	&pound;
¤	general currency sign	&#164;	&curren;
¥	yen sign	&#165;	&yen;
§	section sign	&#167;	&sect;
¨	umlaut(dieresis)	&#168;	&uml;
©	copyright	&#169;	&copy;
ª	feminineordinal	&#170;	&ordf;
«	left angle quote, guillemotleft	&#171;	&laquo;
¬	not sign	&#172;	&not;
-	soft hyphen	&#173;	&shy;
®	registered trademark	&#174;	&reg;

Character	Entity Name	Numeric Code	Descriptive
¯	macron accent	¯	¯
²	superscript two	²	²
¶	paragraph sign	¶	¶
¸	cedilla	¸	¸
º	masculine ordinal	º	º
»	right angle quote, guillemotright	»	»
¼	fraction one-fourth	¼	¼
¾	fraction three-fourths	¾	¾
¿	inverted question mark	¿	¿
Â	capital A, circumflex accent	Â	Â
Ã	capital A, tilde	Ã	Ã
Ä	capital A, dieresis or umlaut mark	Ä	Ä
Å	capital A, ring (Angstrom)	Å	Å
Æ	capital AE diphthong (ligature)	Æ	Æ
Ç	capital C, cedilla	Ç	Ç
Ê	capital E, circumflex accent	Ê	Ê
Ë	capital E, dieresis or umlaut mark	Ë	Ë
Ì	capital I, grave accent	Ì	Ì
Í	capital I, acute accent	Í	Í
Ï	capital I, dieresis or umlaut mark	Ï	Ï
Ð	capital Eth, Icelandic	Ð	Ð
Ñ	capital N, tilde	Ñ	Ñ
Ø	capital O, slash	Ø	Ø
Ù	capital U, grave accent	Ù	Ù
Ú	capital U, acute accent	Ú	Ú
Ü	capital U, dieresis or umlaut mark	Ü	Ü
Ý	capital Y, acute accent	Ý	Ý
Þ	capital THORN, Icelandic	Þ	Þ
ß	small sharp s, German (sz ligature)	ß	ß
ã	small a, tilde	ã	ã
ä	small a, dieresis or umlaut mark	ä	ä
å	small a, ring	å	å
æ	small ae diphthong (ligature)	æ	æ

Character	Entity Name	Numeric Code	Descriptive
ç	small c, cedilla	ç	ç
ð	small eth, Icelandic	ð	ð
ñ	small n, tilde	ñ	ñ
ô	small o, circumflex accent	ô	ô
õ	small o, tilde	õ	õ
ö	small o, dieresis or umlaut mark	ö	ö
ø	small o, slash	ø	ø
ù	small u, grave accent	ù	ù
ú	small u, acute accent	ú	ú
û	small u, circumflex accent	û	û
ü	small u, dieresis or umlaut mark	ü	ü
ý	small y, acute accent	ý	ý
þ	small thorn, Icelandic	þ	þ
ÿ	small y, dieresis or umlaut mark	ÿ	ÿ

Disallowed characters

To prevent rendering and other formatting problems, do not use the following special characters:

Character	Where not to use
Double quotes	• Data entry text box • Form title • Question text • Query answer text • Query text
Single quote	• Question text • Query answer text • Query text • Site name
Apostrophe (')	• Form title • Question text
\ or \\	Do not use these characters.

Character	Where not to use
> or <	Anywhere; use HTML escape character equivalents: • > -- > or > • < -- < or <
Superscript and subscript formatting specifications	Elements that will be used in pulldown controls

Additionally, avoid copying and pasting from the Microsoft Word application into text boxes and query text, as the Microsoft Word application can change characters to Unicode format.

Locale codes

The following table lists the locale codes that you can use to specify translated text in the definition of **TRANSLATION** attributes. Note that the codes used in the Central Designer application are culture-specific. In the table, culture-specific codes have a hyphen between the language and culture.

For more information, see *Translation* (on page 103).

Culture and language code	Culture
af-ZA	Afrikaans (South Africa)
sq-AL	Albanian (Albania)
ar-DZ	Arabic (Algeria)
ar-BH	Arabic (Bahrain)
ar-EG	Arabic (Egypt)
ar-IQ	Arabic (Iraq)
ar-JO	Arabic (Jordan)
ar-KW	Arabic (Kuwait)
ar-LB	Arabic (Lebanon)
ar-LY	Arabic (Libya)
ar-MA	Arabic (Morocco)
ar-OM	Arabic (Oman)
ar-QA	Arabic (Qatar)
ar-SA	Arabic (Saudi Arabia)
ar-SY	Arabic (Syria)
ar-TN	Arabic (Tunisia)
ar-AE	Arabic (U.A.E.)
ar-YE	Arabic (Yemen)
hy-AM	Armenian (Armenia)

Culture and language code	Culture
az-Cyrl-AZ	Azeri (Azerbaijan, Cyrillic)
az-Latn-AZ	Azeri (Azerbaijan, Latin)
eu-ES	Basque (Basque)
be-BY	Belarusian (Belarus)
bg-BG	Bulgarian (Bulgaria)
ca-ES	Catalan (Catalan)
zh-HK	Chinese (Hong Kong SAR, PRC)
zh-MO	Chinese (Macao SAR)
zh-CN	Chinese (PRC)
zh-Hans	Chinese (Simplified)
zh-SG	Chinese (Singapore)
zh-TW	Chinese (Taiwan)
zh-Hant	Chinese (Traditional)
hr-HR	Croatian (Croatia)
cs-CZ	Czech (Czech Republic)
da-DK	Danish (Denmark)
dv-MV	Divehi (Maldives)
nl-BE	Dutch (Belgium)
nl-NL	Dutch (Netherlands)
en-AU	English (Australia)
en-BZ	English (Belize)
en-CA	English (Canada)
en-029	English (Caribbean)
en-IE	English (Ireland)
en-JM	English (Jamaica)
en-NZ	English (New Zealand)
en-PH	English (Philippines)
en-ZA	English (South Africa
en-TT	English (Trinidad and Tobago)
en-GB	English (United Kingdom)
en-US	English (United States)
en-ZW	English (Zimbabwe)
et-EE	Estonian (Estonia)

Culture and language code	Culture
fo-FO	Faroese (Faroe Islands)
fa-IR	Farsi (Iran)
fi-FI	Finnish (Finland)
fr-BE	French (Belgium)
fr-CA	French (Canada)
fr-FR	French (France)
fr-LU	French (Luxembourg)
fr-MC	French (Monaco)
fr-CH	French (Switzerland)
gl-ES	Galician (Spain)
ka-GE	Georgian (Georgia)
de-AT	German (Austria)
de-DE	German (Germany)
de-LI	German (Liechtenstein)
de-LU	German (Luxembourg)
de-CH	German (Switzerland)
el-GR	Greek (Greece)
gu-IN	Gujarati (India)
he-IL	Hebrew (Israel)
hi-IN	Hindi (India)
hu-HU	Hungarian (Hungary)
is-IS	Icelandic (Iceland)
id-ID	Indonesian (Indonesia)
it-IT	Italian (Italy)
it-CH	Italian (Switzerland)
ja-JP	Japanese (Japan)
kn-IN	Kannada (India)
kk-KZ	Kazakh (Kazakhstan)
kok-IN	Konkani (India)
ko-KR	Korean (Korea)
ky-KG	Kyrgyz (Kyrgyzstan)
lv-LV	Latvian (Latvia)
lt-LT	Lithuanian (Lithuania)

Culture and language code	Culture
mk-MK	Macedonian (Macedonia, FYROM)
ms-BN	Malay (Brunei Darussalam)
ms-MY	Malay (Malaysia)
mr-IN	Marathi (India)
mn-MN	Mongolian (Mongolia)
nb-NO	Norwegian (Bokmål, Norway)
nn-NO	Norwegian (Nynorsk, Norway)
pl-PL	Polish (Poland)
pt-BR	Portuguese (Brazil)
pt-PT	Portuguese (Portugal)
pa-IN	Punjabi (India)
ro-RO	Romanian (Romania)
ru-RU	Russian (Russia)
sa-IN	Sanskrit (India)
sr-Cyrl-CS	Serbian (Serbia, Cyrillic)
sr-Latn-CS	Serbian (Serbia, Latin)
sk-SK	Slovak (Slovakia)
sl-SI	Slovenian (Slovenia)
es-AR	Spanish (Argentina)
es-BO	Spanish (Bolivia)
es-CL	Spanish (Chile)
es-CO	Spanish (Colombia)
es-CR	Spanish (Costa Rica)
es-DO	Spanish (Dominican Republic)
es-EC	Spanish (Ecuador)
es-SV	Spanish (El Salvador)
es-GT	Spanish (Guatemala)
es-HN	Spanish (Honduras)
es-MX	Spanish (Mexico)
es-NI	Spanish (Nicaragua)
es-PA	Spanish (Panama)
es-PY	Spanish (Paraguay)
es-PE	Spanish (Peru)

Culture and language code	Culture
es-PR	Spanish (Puerto Rico)
es-ES	Spanish (Spain)
es-ES_tradnl	Spanish (Spain, Traditional Sort)
es-UY	Spanish (Uruguay)
es-VE	Spanish (Venezuela)
sw-KE	Swahili (Kenya)
sv-FI	Swedish (Finland)
sv-SE	Swedish (Sweden)
syr-SY	Syriac (Syria)
ta-IN	Tamil (India)
tt-RU	Tatar (Russia)
te-IN	Telugu (India)
th-TH	Thai (Thailand)
tr-TR	Turkish (Turkey)
uk-UA	Ukrainian (Ukraine)
ur-PK	Urdu (Pakistan)
uz-Cyrl-UZ	Uzbek (Uzbekistan, Cyrillic)
uz-Latn-UZ	Uzbek (Uzbekistan, Latin)
vi-VN	Vietnamese (Vietnam)

Versions

About study versioning

After you create and deploy a study to the InForm application, you may need to update a study object such as a form or a visit. To make a study version change, or an in-place revision change, you must use the Central Designer application. For more information, see the Central Designer *InForm Design Guide*.

The MedML Installer utility

Overview of the MedML Installer utility

The MedML Installer utility:

- Parses the XML files that you generate with MedML elements.
- Validates the files against the following:
  - The MedML schema file (MedMLSchema.xsd).
  - The constraints enforced by the database schema.
- Loads the study components that the files define into the database.

You can run the MedML Installer utility:

- From the command line.

  For more information, see *Running the MedML Installer utility from the command line* (on page 32).
- From the Start menu.

Launching the MedML Installer utility

- Select **Start** > **Programs** > **Oracle** > **InForm 6.1** > **InForm MedML Installer**.

  OR
- Run the utility from the command line.

  For more information, see *Running the MedML Installer tool from the command line* (on page 32).

About the MedML Installer window

Definitions are processed from the bottom up. Make sure that your component definitions and XML files are organized in the correct order. The elemental components, such as controls, must be defined, imported, and read first, followed by items that reference them.

Field	Description
XML File	The name of the file to add to (or modify in) the install.
XML Files to be Processed	A list of all of the files to be included in the next install.
Trial Name	The name of the study on which you are working.
Additional Path	If you are using a response (RSP) file to process a group of XML files, type the name of the root directory where the XML files that are referenced in the RSP file are located.
Verbose	Creates a detailed description of what happened during the build.

Field	Description
Parse Only	Checks the input file for XML errors and compatibility with the InForm database without loading data.
Strict Mode	Accepts only complete component definitions. By default, the checkbox is not selected, indicating nonstrict mode. In nonstrict mode, the MedML Installer utility accepts incomplete component definitions in which not all dependent components are present. For example, it accepts a radio control definition in which not all of the element definitions have been loaded previously into the database. **Note:** This option is available only if you start the utility from the command line with the -notstrict option. Nonstrict mode is intended only for a study development environment. Do not use it in production. Similarly, once a connection is defined, you can only load study definition data with strict MedML processing.
Online	Starts the study in online mode. By default, the checkbox is selected. In online mode, if a study is not started when you start the MedML Installer utility, the utility starts the study and does not shut it down when the installation is complete. This mode allows users to view the results of installing study definition data immediately. If you deselect this checkbox, the MedML Installer utility starts the study in offline mode. In offline mode, the MedML Installer utility stops the study when installation is complete. Users cannot connect to the study until it is restarted. **Note:** This option is available only if you start the utility from the command line with the -online option.
Output file	Location of a file that contains processing messages.

Working with the MedML Installer utility

Running the MedML Installer utility for the first time

To start testing your XML files in the study, use the MedML Installer utility to build the files into the study database.

To run the MedML Installer utility:

1 Launch the MedML Installer utility. For more information, see *Launching the MedML Installer utility* (on page 29).

The MedML Installer window appears.

2 In the **XML File** field, select the XML file to process, or browse to locate the file.

Note: You can also list all the files to process in a response (RSP) file and type the response file name in the XML File field.

3 Click **Add**.

4 Repeat steps 1 to 3 for all your XML files.

5 From the **Trial Name** drop-down list, select or type the name of the study.

6 Optionally, select additional checkboxes in the MedML Installer window to specify the way that the MedML Installer utility will process the files.

For more information, see *About the MedML Installer window* (on page 29).

7 Click **Process**.

The MedML Installer utility processes the XML files and loads information into the study database. When all the XML files have been processed, the message **Completed Successfully** appears in the MedML Installer Output window.

For more information on error messages, see *MedML Installer output messages* (on page 34).

Updating a study that is already in progress

When you change a file in a study that is already in progress, you must process the new version of the file, and process a new study version to capture a new version of all of the objects that the changed file is referenced by. For example, if you make a change to a section, you must reprocess the file for the section and all files that contain references to that section.

When you use the MedML Installer utility to update a study that is already in progress, you must stop and restart the study definition with pfadmin commands for the changes to take effect.

To update a study that is already in progress:

1 Limit access to the study by changing the Directory Security properties of the study's virtual directory in **Internet Information Server** through **Microsoft Windows Components**.

For more information, see the *Study and Reporting Setup Guide*.

2 Launch the MedML Installer utility.

The MedML Installer window appears.

3 In the **XML File** field, select the file to which you have made changes.

4 Click **Add**.

The XML file appears in the list of XML files to be processed.

5 From the **Trial Name** drop-down list, select or type the name of the study.

6 Optionally, select additional checkboxes in the MedML Installer window to specify the way that the MedML Installer utility will process the files.

For more information, see *About the MedML Installer window* (on page 29).

7 Click **Process**.

The MedML Installer utility processes the XML files and loads information into the study database. When all the XML files have been processed, the message **Completed Successfully** appears in the MedML Installer Output window.

For more information on error messages, see *MedML Installer output messages* (on page 34).

8 Stop and restart the study.

Removing XML files from the build

Regulatory authorities have strict regulations against removing data from a study. XML files that are part of a study cannot be removed. You can instead remove all references to any unnecessary files in the other XML files, and reprocess all the files using the MedML Installer utility. The original file definitions for the removed files remain in the database, but are not visible in the study.

Running the MedML Installer utility from the command line

Note: Oracle recommends that you run the MedML Installer utility through the PFConsole utility. For more information, see ***Running the PFConsole utility from the command line*** (on page 3).

To run the MedML Installer utility from the command line, use the following syntax:

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

Note: All command line applications, such as the MedML Installer utility, the InForm Data Import utility, and the InForm Data Export utility use the default product locale specified during the InForm installation, or through PFAdmin commands.

Command line parameters

Parameter	Variable	Description
PFConsole utility		Starts the PFConsole utility.
PFMMinst		Starts the MedML Installer utility.
-trial	trialname	The name of the study you are building.
-verbose		Specifies whether to generate details of the build and display them during the build.
-?		Displays the syntax of the command line for running the MedML Installer utility.
-help		Opens the online Help for the MedML Installer utility.
-autorun		Runs the MedML Installer utility in a command window. Oracle recommends that you use the PFConsole utility to run the MedML Installer utility from the command line. For more information, see ***PFConsole utility*** (on page 1).

Parameter	Variable	Description
-notstrict		Opens the MedML Installer dialog box, which has a checkbox for specifying strict mode. By default the checkbox is deselected, indicating nonstrict mode. In nonstrict mode, the MedML Installer utility accepts incomplete component definitions in which not all dependent components are present; for example, it accepts a radio control definition in which not all of the element definitions have been previously loaded into the database. **Note:** Use nonstrict mode only in a study development environment. Never load production study definitions in nonstrict mode.
-outfile	output filename	Specifies whether to save the details of the build in an output file, and indicates the path and file name in which to save it.
-parse		Instructs the utility not to commit the build to the database. This mode is useful for testing for errors.
-xml	XMLFile	Specifies the name of the XML file to include in the build.
@<response>		Specifies the name of a response file with the list of XML files to be processed.

MedML Installer output messages

The MedML Installer utility generates three types of output messages during the installation process.

Message	Description	Example
Information	Conditions that you should be aware of, but that probably will not cause serious problems with the file installation.	Information: Inserting PFELEMENT 'mestrUnk'
Warning	Events that might be detrimental and should be fixed, but that do not stop the file installation.	Warning: File "D:\PF408\FLTELT.xml' does not contain any known tags
Failure (Error)	An event that stops the file installation.	Error: Could not obtain child section 'sctIEInclusion'. Error: Item not found

If the MedML Installer utility does not complete successfully, to determine where the error occurred:

- Check the output messages.
- Make sure that you started the InForm application before you ran the MedML Installer utility.
- Make sure that the build is in the correct directory.
- Make sure that all the parameters are accurately fulfilled.
- Check for correct paths to input files.
- Make sure components are processed in the correct order and that all necessary components are present in the XML file or database.

Note: The MedML Installer utility shows both errors and warnings in the Output Window and in the log file. You must correct errors before you can install the XML file.

External Data Mapping

Defining mappings to external data formats

In conjunction with defining a study, you may want to create mappings from the study components in an InForm study database to external data formats. After creating mapping definitions, you can use the InForm Data Export utility to export study data in the format you select. The MedML schema elements support mappings to:

- Clintrial database.
- Customer-Defined Database (CDD).

To load mapping definitions into the study database:

1 Generate mapping definitions as XML files.

2 Process the XML files with the MedML Installer utility.

External mapping targets

The external entities for which you can generate mappings are:

- **Central Coding**—Mappings specify information that enables codes to be applied in the Central Coding application and returned to the InForm application:
  - Coding dictionary.
  - Controls in the InForm application to be coded (verbatims).
  - Controls in the InForm application to be used as code targets and context items.
- **Clintrial**—Mappings specify target items within panels in a Clintrial study definition.
- **Customer-Defined Database (CDD)**—Mappings specify target table columns within an external Oracle database.

External Data Mapping Reference

AssocCDD

Purpose

Specifies the mapping in a customer-defined database of an association between two forms. An association definition is specified as a formset with **TYPE**=RELATION.

Syntax

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

Attributes

REFNAME="*name*"

RefName of the CDD to which to map the association. Required.

DESIGNNOTE="*text*"

Free-form text, with a maximum of 255 characters, containing any information you want to capture about the design of the component. This information is for documentation only and is not displayed. Optional.

ACTIVE="*true*|*false*"

Indicates whether the component is active. The options are true or false. True is the default. Optional.

TARGETTABLE="*name*"

Name of the target CDD table in which the components of the association will be updated. An association table consists of four columns containing the RefNames of the two forms that make up the association, along with the visit in which each form occurs. Required.

ASSOCREFNAME="*name*"

RefName of the formset that defines the association. Required.

LABEL="*name*"

Text label for the association. The label can have a maximum of 255 characters. Optional.

Example

The following example illustrates the mapping of an association between the AE and CM forms in the COMMONCRF formset.

```
<EXTERNALMAP>

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

  <ASSOCCDD REFNAME="CDD_WITH_ASSOC"
    TARGETTABLE="t_assoc1"
    ASSOCREFNAME="AECM_ASSOC"/>

</EXTERNALMAP>
```

CDD

Purpose

Specifies the target of one mapped control in a CDD.

Syntax

```
<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|PATIENTTOSEC
TION|
    PATIENTTOITEMSET|PATIENTTOITEM|PATIENTTOCONTROL|
    PIVOTPATIENT|PIVOTVISIT|PIVOTFORM|PIVOTSECTION"
  [PIVOTCOLUMN="true|false"]
  [LABEL="string"]/>
```

Attributes

REFNAME="*name*"

RefName of the CDD. Required.

DESIGNNOTE="*text*"

Free-form text, with a maximum of 255 characters, containing any information you want to capture about the design of the component. This information is for documentation only and is not displayed. Optional.

ACTIVE="*true*|*false*"

Indicates whether the component is active. The options are true or false. True is the default. Optional.

TARGETTABLE="*name*"

Name of the target customer data table in which the mapped form item will be updated. The name must be 30 characters or fewer. Required.

TARGETCOLUMN="*name*"

Name of the column in the target customer data table in which the mapped form item will be added or updated. The name must be 25 characters or fewer. Required.

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

Type of data contained in the target column. Required.

The options are:

- **NUMERIC**
- **FLOAT**
- **DATE**— Data for a complete date and time field. Note that when you map a date and time field, the InForm software creates four columns:
  - **DATE**—For a complete date and time.
  - **_DT suffix**—For a date and time field with only date components.
  - **_TM suffix**—For a date and time field with only time components.
  - **_STR suffix**—For an incomplete date and time field.
- **SPLITDATE**—Data from a DateTime control, mapped to six columns, each containing one of the date or time components of the control. Each column has the generated or specified column name and the appropriate one of the following suffixes: _Day, _Mon, _Year, _Hour, _Min, or _Sec.
- **STRING**—Fewer than 255 characters.
- **TEXT**—Long varchar data.

Note: The **TARGETCOLUMNTYPE** attribute of a column that receives data from a multiple-selection control such as a checkbox control, must be set to STRING.

TARGETCOLUMNMAXLENGTH="*length*"

Maximum length of a target column with a **TARGETCOLUMNTYPE**=STRING. The column must be fewer than 255 characters. Optional.

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

For the following key types, this specifies the composition of the primary key columns of the target table. Each time a component of the primary key changes from the previous submitted primary key, the InForm software inserts a new row in the target table. Primary keys consist of the following DBUIDs and indexes:

- **PATIENT**—PatientID, ItemsetIndex.
- **PATIENTVISIT (default)**—PatientID, VisitID, ItemsetIndex, and VisitIndex.
- **PATIENTTOFORM**—PatientID, VisitID, ItemsetIndex, VisitIndex, and FormID.
- **PATIENTTOSECTION**—PatientID, VisitID, ItemsetIndex, VisitIndex, FormID, and SectionID.
- **PATIENTTOITEMSET**—PatientID, VisitID, ItemsetIndex, VisitIndex, FormID, SectionID, and ItemsetID.

- **PATIENTTOITEM**—PatientID, VisitID, ItemsetIndex, VisitIndex, FormID, SectionID, ItemsetID, and ItemID.
- **PATIENTTOCONTROL**—PatientID, VisitID, ItemsetIndex, VisitIndex, FormID, SectionID, ItemsetID, ItemID and five ControlIDs. A target table with this key type also contains a data label that can be used for data selection.

For the following key types, the primary key columns are PatientID, VisitID, ItemsetIndex, VisitIndex, FormID, SectionID, ItemsetID, ItemID and five ControlIDs. The key type selection determines the composition of a pivot set (a group of columns in which one column is defined as the pivot column); within a pivot set, data elements mapped to non-pivot columns are repeated in each row. Target tables with these key types also contain a data label, specified as the value of the **LABEL** attribute, which can be used for data selection. Pivot set keys consist of the following DBUIDs and indexes:

- **PIVOTPATIENT**—PatientID and VisitIndex
- **PIVOTVISIT**—PatientID, VisitID, and VisitIndex
- **PIVOTFORM**—PatientID, VisitID, FormID, and VisitIndex
- **PIVOTSECTION**—PatientID, VisitID, FormID, SectionID, and VisitIndex

Note: A table that has mappings for controls within an itemset cannot have any of the pivot key types.

PIVOTCOLUMN="*true*|*false*"

Indicates whether the column specified in the **TARGETCOLUMN** attribute is a pivot column. The options are true or false. When the **TARGETKEYTYPE** is PIVOTPATIENT, PIVOTVISIT, PIVOTFORM, or PIVOTSECTION, data elements mapped to non-pivot columns are repeated in each row within a pivot set.

Note: The pivot column must be the first column in the table.

LABEL="*string*"

Text label for the data item, allowing access to an item in a target table with the PATIENTTOCONTROL or any of the PIVOT key types. The label must be 255 characters or fewer. Optional.

Example

The following example illustrates the use of the CDD element to map the DESCRIBETEXT control to a column in the CDD1 CDD.

```
<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

Purpose

Identifies the RefName of a formset as the location of a mapped control in the Path element of a mapping definition. Include one Chapterref element in a RefName path defined by a Path element.

Syntax

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

Attributes

REFNAME="*name*"

RefName of the formset in which the mapped source control occurs. To specify that the data should be mapped to the target specified in the mapping definition from every visit in which the specified form/section/itemset/item combination occurs, enter the special RefName "PF_ALL_VISITS" as the RefName. Required.

Example

The following example illustrates the use of the PF_ALL_VISITS RefName in the Chapterref element of a Path definition.

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

CodeTarget (mappings)

Purpose

Specifies the mapping of a control holding coded data in an InForm study to a specific type of code target in the dictionary used to code the data. The CodeTarget element is a child element in the CodingMap structure for a verbatim specified by the Path element in a set of Central Coding data mappings.

Syntax

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

Attributes

NAME="*name*"

Name of the dictionary code target used to code the data in the verbatim specified by the Path element and its child components. Required.

PATH="*path*"

RefName path of the control that holds data after it is coded. Required.

Restrictions

The MedML Installer utility and the InForm application enforce the following restrictions on Path elements defining code target control paths:

- The code target control path must be unique within a mapping.
- The code target control must be different from the verbatim and from any code target control paths within a mapping.
- The code target control path must point to the top-level text box control or calculated control.

- Verbatim, code target, and context item controls must be:
  - In the same visit if the visit is repeating.
  - On the same form if the form is repeating.
  - In the same itemset if the coded data appears in an itemset.
  - Different from each other.

Example

In the following example, the ExternalMap element identifies the mappings for one verbatim in an InForm study, specified with the Path element, to be coded with the Central Coding application.

```
<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

Purpose

Specifies the mappings for one verbatim item to be coded.

Syntax

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

Attributes

REFNAME="*name*"

RefName of the coding mapping component containing the verbatim item to be coded. Required.

VERBATIMTYPE="*type*"

Type of verbatim to be coded. Optional. This value corresponds to the item type in the Central Coding application:

- **AE**—Adverse event
- **DISEASE**—Disease
- **LABDATA**—Lab data
- **MEDPROD**—Medical product

AE, DISEASE, and LABDATA are valid verbatim types for the MedDRA dictionary. MEDPROD is a valid verbatim type for the WHO-DD dictionary.

Children

A CodingMap element has the following child elements:

- One Dictionary element specifying the coding dictionary to use.
- At least one CodeTarget element specifying the mapping between the InForm control holding the data after it is coded and the dictionary code target used to code the data.
- Optionally, a ContextInformation element containing one or more ContextItem elements. A ContextItem element specifies the mapping between an InForm control providing additional context information and the dictionary context item used to code the data.

Example

```
<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>
```

ContextInformation

Purpose

Encloses the mapping definitions for context items. Controls identified as context items in the InForm application provide data to context items in a coding dictionary.

Syntax

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

Children

The ContextInformation element has one or more ContextItem child elements, each specifying the mapping between a control in the InForm application and a dictionary context item.

Example

```
<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>
```

ContextItem (mappings)

Purpose

Specifies the mapping of a control holding coded data in an InForm study to a specific type of context item in the dictionary used to code the data. The ContextItem element is a child element in the ContextInformation structure for a verbatim specified by the Path element in a set of Central Coding data mappings.

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

Attributes

NAME="*name*"

Name of the dictionary context item used to code the data in the verbatim specified by the Path element and its child elements. Required.

PATH="*path*"

RefName path of the control that holds the data for the dictionary context item. Required.

Restrictions

The MedML Installer utility and the InForm application enforce the following restrictions on Path elements defining code target control paths:

- The code target control path must be unique within a mapping.
- The code target control must be different from the verbatim and from any code target control paths within a mapping.
- The code target control path must point to the top-level text box control or calculated control.
- Verbatim, code target, and context item controls must be:
    - In the same visit if the visit is repeating.
    - On the same form if the form is repeating.
    - In the same itemset if the coded data appears in an itemset.
    - Different from each other.

Example

```
<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

Purpose

Defines a Clintrial panel in a set of mappings to be exported from the InForm database to the CIS application. A panel definition consists of a ContextPanel element, which is associated with CTItem elements defining the items common to all panels and one or more CTPanel elements, each associated with a single CTItem element defining a panel item.

Syntax

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

Attributes

REFNAME="*name*"

RefName of the set of Clintrial mappings. Required.

ACTIVE="*true*|*false*"

Indicates whether the component is active. The options are true or false. True is the default. Optional.

DESIGNNOTE="*text*"

Free-form text, with a maximum of 255 characters, containing any information you want to capture about the design of the component. This information is for documentation only and is not displayed. Optional.

Children

The definition of a ContextPanel element can include one or more CTItem elements, each of which defines a panel item that appears in all panels.

Example

This example illustrates a ContextPanel definition included in a mapping set called MEDIKA_MAP. The ContextPanel definition includes four CTItem elements, each of which defines a panel item common to all panels.

```
<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

Purpose

Includes the definition of one type of component in the definition of another component. A ControlRef appears only as the child of a component in which it is included; it is not submitted as a stand-alone component.

Types of components in which you can include ControlRefs:

- CheckBoxControl
- GroupControl
- Item
- RadioControl
- Path

Types of components you can include by using ControlRefs:

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

Syntax

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

Attributes

REFNAME="*name*"

RefName of the component that the ControlRef is including in the definition of a compound control. Required.

ORDER="*n*"

Sequence in which each ControlRef appears in the compound control definition. Optional. If you do not specify an order, the MedML Installer utility orders the components referred to by the ControlRefs in the order in which you enter them.

SELECTIONVALUE="*text*"

Value of the compound control included by the Controlref. Optional. If you do not specify a value, when a user selects one of the controls in the compound control, the value stored in the database is a string consisting of the characters !pf! and the DBUID path of the selected control. For example, this attribute is useful for providing a value to store for a checkbox or radio control component labeled *Other* that is associated with a text box control.

Note: When defining compound controls with Controlref or Unitref definitions, ensure that all subordinate controls return the same data type, by defining them with the same TYPE attribute.

When defining compound controls, note that the InForm application supports a maximum of five levels of nesting. Although five levels are supported, as a design practice, you should attempt to minimize the number of nested levels to help performance.

When including a Controlref definition in a Path element, only the REFNAME attribute is valid.

Examples

This example illustrates how to create a list of radio buttons in which the first button is a drop-down list and the second is a text box. The definition of the radio button list includes the definition of each list item as a Controlref. This list will be used to capture a medication regimen.

1 Define the drop-down list as a set of PFElements, and include them in a PullDownControl definition by using Elementrefs.

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 Define the text box and its caption as a TextControl.

4
```
<TEXTCONTROL REFNAME="MEDREGTEXT"
   NAME="MEDREGTEXT"
   HEIGHT="1"
   LENGTH="20"
   MAXLENGTH="20"
   DATATYPE="STRING"
   CAPTION="Other (specify): "
   CAPTIONALIGN="TOP"/>
```

5 Include the pull-down control and text control in a RadioControl by using Controlrefs. Note the use of the **SELECTIONVALUE** attribute to assign the value "OTHER" to the Other (Specify): radio button.

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

This example illustrates the use of Controlref definitions in the Path element to identify the source control used in creating a mapping of components between the InForm database and a CDD.

```
<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>
```

CTItem

Purpose

Defines the mapping of an InForm form item to a Clintrial panel item. The attributes of CTItem describe the attributes of the item in the Clintrial software.

Syntax

```
<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*"]/>

Attributes

REFNAME="*name*"

RefName of the item. Required.

DESCRIPTION="*text*"

Description of the item. Optional.

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

Data type for the value of the item. The options are:

- Text
- Fixed
- Float
- Date
- DateTime

Required.

SUBSETVALUE="*text*"

Value the item takes if it is the subset key for subset page sections based on the panel. Optional.

BLOCKKEYVALUE="*text*"

Value of the Clintrial block key. If you specify this value, it overrides the visit RefName as the block key. Optional.

PAGEKEYVALUE="*text*"

Value of the Clintrial page key. If you specify this value, it overrides the form RefName as the page key. Optional.

DATEPART="*n*"

Part of a date time control to be mapped to a Clintrial item. Use this attribute if you are mapping parts of a date time control to separate Clintrial items. The options are:

- 0—Undefined
- 1—Year
- 2—Month
- 3—Day
- 4—Hour
- 5—Minute

- 6—Second

Optional.

ISDERIVED="*true*|*false*"

Indicates whether the value of the item is determined from a derivation associated with the panel. The options are true or false. False is the default. Optional.

ISREQUIRED="*true*|*false*"

Indicates whether the item is required. The options are true or false. Required.

DBFORMAT="*format*"

Format in which the Clintrial software stores values for the item in the Oracle database. The Oracle database formats are:

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

Required.

SASNAME="*name*"

Name of the item when data is sent to SAS through the Clintrial SAS interface. The name must be eight characters or fewer and conform to SAS naming requirements. Optional.

CONTEXTTYPE="*n*"

Context type of the item. Context items are associated with each record in a clinical data table defined by a panel of Types, 1, 2, 3, 4, or 5 and are included in the ContextPanel definition for a protocol. Types are:

- 0—Not a context item.
- 1—Subject-related context item.
- 2—Visit-related context item.
- 3—Page-related context item.
- 4—Other context item.

Required.

ISREPEAT="*true*|*false*"

Indicates whether an item is one for which multiple values can be entered within a page section.

CODELIST="*text*"

Name of a codelist associated with the item. **CODELIST** and **CHECKLIST** are mutually exclusive. A codelist encodes entered values. Only codes or values in the codelist can be entered as values of the item. Optional.

CHECKLIST="*text*"

Name of a checklist associated with the item. **CODELIST** and **CHECKLIST** are mutually exclusive. A checklist is a type of codelist that is used to view suggested entries for a field. Optional.

RANGELB="*n*"

Minimum value that can be entered for the value of the item. Optional.

RANGEUD="*n*"

Maximum value that can be entered for the value of the item. Optional.

KEYORDER="*n*"

The order in which the item appears in the concatenation of key items, if the item is part of the panel's key. 0 (not a key item) is the default. Optional.

COPYWITHPANEL="*true*|*false*"

Indicates whether the item should be included with the panel if the panel is copied. Options are true or false. True is the default. Optional.

LOCKSTATUS="*n*"

Indicates whether the protocol in which the item is included is locked:

- 0—item is modifiable
- 1—item is not modifiable, but it can be reset to modifiable
- 2—item is not modifiable and cannot be made modifiable.

The default is 0. Optional.

ISKEY="*true*|*false*"

Indicates whether the item is part of the panel's key. False is the default. Optional.

Example

This example specifies the mapping of three items in the CLIN1 panel.

```
<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

Purpose

Defines a Clintrial panel in a set of mappings to be exported from the InForm database to the CIS application. A panel definition consists of one or more CTPanel elements, each associated with a single CTItem element defining a panel item, and a ContextPanel element, which is associated with CTItem elements defining the items common to all panels.

Syntax

```
<CTPANEL
  REFNAME="name"
  PANELNAME="name"
  [DESCRIPTION="text"]
  PANELTYPE="n"
  [SUBSETITEM="name"]
  [ISPROTECTED="true|false"]
  [ISVERIFIABLE="true|false"]
  ISDETAILPANEL="true|false"
  [DETAILCTITEM="name"]
  [MASTERPANEL="name"]
  [MASTERCTITEM="name"]
  [SASNAME="name"]
  [LOCKSTATUS="n"]
  [ACTIVE="true|false"]
  [DESIGNNOTE="text"]>
  <CTITEM attributes/>
</CONTEXTPANEL>
```

Attributes

REFNAME="*name*"

RefName of the set of Clintrial mappings. Required.

PANELNAME="*name*"

Name of the Clintrial panel. Required.

DESCRIPTION="*text*"

Description of the Clintrial panel. Optional.

PANELTYPE="*n*"

Clintrial panel type:

- 0—Non-subject data; that is, data is not related to a specific subject.
- 1—One record per subject; one record can be collected only one time during the study for each subject.
- 2—More than one record per subject; multiple records can be collected one time in the study for each subject.
- 3—One record per subject visit; one record can be collected for each subject visit.

- 4—More than one record per subject visit; multiple records can be collected for each subject visit.
- 5—One record for each enrolled subject.

Required.

SUBSETITEM="*name*"

Name of the item specified as the subset key for subset page sections based on the panel. A subset page section can occur multiple times on a study page in a Type 0, Type 2, or Type 4 panel, with each value of the subset key item representing distinct rows (subsets) of data. Optional.

ISPROTECTED="*true*|*false*"

Indicates whether access rights to the panel are limited in Clintrial. The options are true or false. False is the default. Optional.

ISVERIFIABLE="*true*|*false*"

Indicates whether double-entry of data in panel items is required for verification. False is the default. Optional.

ISDETAILPANEL="*true*|*false*"

Indicates whether the CTPanel definition participates in a detail page section in a master-detail relationship: true or false. A master-detail relationship is a relationship between two page sections on a study page, in which each record in one page section (the master page section) can have one or more associated records in the other section (the detail page section). During data entry the displayed records in the detail page section are associated with the selected record in the master page section. Required.

DETAILCTITEM="*name*"

Name of the item identified as the detail key item, if the CTPanel definition is part of a detail page section. Optional.

MASTERPANEL="*name*"

PANELNAME of the master panel with which this CTPanel definition participates in a master-detail relationship. This attribute is valid only if the value of the **ISDETAILPANEL** attribute is true. Optional.

MASTERCTITEM="*name*"

Name of the item on the associated master panel that corresponds to the detail key item specified in the **DETAILCTITEM** attribute. This attribute is valid only if the value of the **ISDETAILPANEL** attribute is true. Optional.

SASNAME="*name*"

Name of the panel when data is sent to SAS through the Clintrial SAS interface. The name must be eight characters or fewer and conform to SAS naming requirements. Optional.

LOCKSTATUS="*n*"

Indicates whether the protocol in which the panel is included is locked:

- 0—panel is modifiable

- 1—panel is not modifiable, but it can be reset to modifiable
- 2—panel is not modifiable and cannot be made modifiable.

The default is 0. Optional.

ACTIVE="*true*|*false*"

Indicates whether the component is active. The options are true or false. True is the default. Optional.

DESIGNNOTE="*text*"

Free-form text, with a maximum of 255 characters, containing any information you want to capture about the design of the component. This information is for documentation only and is not displayed. Optional.

Children

The definition of a CTPanel element includes one CTItem element, which defines an item in the panel.

Example

The following example illustrates the definition of the **INCLUS** panel in the MEDIKA_MAP mapping, along with the definition of the INCLU1 item. The Path element maps this panel to a RefName path in an InForm study definition.

```
<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 (mappings)

Purpose

Identifies a coding dictionary within a CodingMap structure that specifies the mappings for one verbatim item to be coded.

Syntax

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

Attributes

TYPE="*text*"

Type of dictionary, for example, MedDRA or WHO-DD. Required.

VERSION="*text*"

Dictionary version, for example, 8.1, 05Q4. Required.

CULTURE="*text*"

Language and culture, for example, en-JP. Required.

Note: The combination of **TYPE**, **VERSION**, and **CULTURE** values uniquely identifies a dictionary.

Example

```
<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

Purpose

A wrapper for the mappings for one control in an InForm study. The ExternalMap element can include:

- One Path element, which identifies the source control in the InForm study or signals the presence of mapping targets that have no corresponding control in the InForm study. For example:
  - Columns to be created in a CDD that have no associated control on a CRF.
  - Items that are common to all panels in a Clintrial protocol.
  - Multiple elements that specify the target of each mapping.

Syntax

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

Children

The ExternalMap element can include the following mapping definition elements:

- Path
- CDD
- ContextPanel
- CTPanel
- CodingMap

Example

The following example illustrates the use of the ExternalMap element along with an unqualified Path element to define a set of mappings for items that are common to all panels in the CLIN1 mapping. Because the items are common, they do not correspond to a specific RefName path in the InForm study.

```
<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>
```

In the following example, the ExternalMap element defines the mapping from a control in an InForm study, specified by the Path element, to a table column in a CDD, specified by the CDD element.

```
<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>
```

In the following example, the ExternalMap element identifies the mappings for one verbatim in an InForm study, specified with the Path element, to be coded with the Central Coding application.

```
<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

Purpose

Identifies a set of external mappings with one RefName. Optional.

Syntax

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

Attributes

REFNAME="*name*"

RefName of the ExternalMapSet. Required.

DESIGNNOTE="*text*"

Free-form text, with a maximum of 255 characters, containing any information you want to capture about the design of the component. This information is for documentation only and is not displayed. Optional.

ACTIVE="*true*|*false*"

Indicates whether the component is active. The options are true or false. True is the default. Optional.

TYPE="CDD|CLINFORM|CODINGMAP"

Indicates the target entities for which mappings are generated. Values are:

- **CDD**—Table columns within an external Oracle database.
- **CLINFORM**—Items within panels in a Clintrial study definition.
- **CODINGMAP**—Central Coding application.

Required.

Example

The following example illustrates the ExternalMapSet element in the MedML that defines the MEDIKA-CLINICAL sample study.

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

Itemref

Purpose

- Includes previously defined items in the definition of a section, itemset, or item group. Include one Itemref element for each item in the section, itemset, or item group definition.
- Specifies the RefName of an item as the location of a mapped control in the Path element of a mapping definition. Include one Itemref element in a RefName path defined by a Path element.

An itemref appears only as the child of a section or path in which it is included; it is not submitted as a stand-alone component.

Syntax

```
<ITEMREF
  REFNAME="name"
   [KEYITEM="true|false"]
   [UNIQUEKEY="true|false"]
   [ORDER="n"]/>
```

Attributes

REFNAME="*name*"

RefName of the item that the itemref is including in the definition of a section or path. Required.

KEYITEM="*true*|*false*"

Indicates whether the item that the itemref is including is a key item in an itemset definition. The values of key items are displayed in a drop-down list in the InForm application to allow you to navigate to a specific instance of the itemset. Only items with text, numeric, or date/time values can be key items. Items with compound controls, including checkbox controls, group controls, radio controls, or drop-down lists cannot be key items. Options are true or false. False is the default. Optional.

Note: You can specify key items for repeating forms by using the KeyItemref element. Items that are defined as key items in an itemset cannot also be key items in a repeating form.

UNIQUEKEY="*true*|*false*"

Indicates whether the key item must be unique within the formset, form, and itemset. true or false. If a key item is defined as unique, and two rows of an itemset are submitted in the same visit and form with the same key item value, the InForm application rejects the input of the second instance. Options are true or false. False is the default. Optional.

ORDER="*n*"

Sequence in which each itemref appears in the Section definition. If you do not specify an order, the MedML Installer utility orders the items referred to by the itemrefs in the order in which you enter them. Optional.

Note: When including an Itemref definition in a Path element, only the **REFNAME** attribute is valid.

Examples

This example illustrates the use of itemrefs in the definition of a section called Duration of Hypertension, which contains two items.

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

The following example shows the use of an itemref definition in a Path element. The CLIN Item is the location where the mapped TESTTEXT control occurs.

```
<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>
```

The following example shows how three key items designated as a unique key combination are included in an itemset definition with Itemref elements.

```
<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

Purpose

Identifies the RefName of an itemset as the location of a mapped control in the Path element of a mapping definition. Include one ItemSet element in a RefName path defined by a Path element.

Syntax

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

Attributes

REFNAME="*name*"

RefName of the itemset in which the mapped source control occurs. Required.

Example

The following example shows the LAB itemset as the location where the mapped TESTTEXT control occurs.

```
<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

Purpose

Identifies the RefName of a form as the location of a mapped control in the Path element of a mapping definition. Include one Pageref element in a RefName path defined by a Path element.

Syntax

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

Attributes

REFNAME="*name*"

RefName of the form in which the mapped source control occurs. Required.

Example

The following example shows the ECG form as the location where the mapped COMMONDATE control occurs.

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

Path

Purpose

- Used alone, with no child components, a Path element signals the presence of mapping targets that have no corresponding control in the InForm study. For example:
  - Columns to be created in a CDD that have no associated control on a CRF.
  - Items common to all panels in a Clintrial protocol.
- Used in conjunction with one or more child components, each of which provides the RefName of a portion of a RefName path, a Path element specifies a control in an InForm study. This control identifies the source data for one data mapping.

The Path element is a required component of a mapping definition created by the ExternalMap element and must be the first child component element in the ExternalMap definition.

Syntax

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

Children

Each optional child component of the Path element specifies the RefName of one portion of the RefName path of a source control in an InForm study:

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

Example

This example illustrates the use of the Path element without child components in a mapping definition for items common to all Clintrial panels in the mapping.

```
<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>
```

Sectionref

Purpose

- Includes previously defined sections in the definition of a form.
- Includes one Sectionref element for each section in the form.
- Specifies the RefName of a section as the location of a mapped control in the Path element of a mapping definition.
- Includes one Sectionref element in a RefName path defined by a Path element.

A sectionref appears only as the child of a Form or Path element in which it is included; it is not submitted as a stand-alone component.

Syntax

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

Attributes

REFNAME="*name*"

RefName of the section that the Sectionref is including in the definition of a form or path. Required.

ORDER="*n*"

Sequence in which each sectionref appears in the form definition. If you do not specify an order, the MedML Installer utility orders the items referred to by the sectionrefs in the order in which you enter them. Optional.

Note: When including a Sectionref definition in a Path element, only the **REFNAME** attribute is valid.

Example

This example illustrates the use of Sectionrefs in the definition of the Demographics form, which contains two sections: Demographics (DEM) and Smoking History (SH).

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

The following example shows the LEADECG section as the location where the mapped COMMONDATE control occurs.

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

MedML Schema Reference

QueryGroup

Purpose

Specifies a set of users who can close queries initiated by other members of the same group.

Note: You cannot create query groups in the user interface. You must use MedML to define query groups.

Note: The only way to remove a user from a group is through the InForm application. You cannot remove a user by running the MedML Installer utility.

Syntax

```
<QUERYGROUP
   GROUPNAME="name"
   [GROUPDESCRIPTION="text"]
   [UUID="id"]>
   <USERREF* attributes/>
</QUERYGROUP>
```

Attributes

GROUPNAME="*name*"

Name of the QueryGroup. Required.

GROUPDESCRIPTION="text"

Text describing the QueryGroup. Optional.

UUID="*id*"

Universally Unique Identifier; a string that identifies the component uniquely across all studies, study databases, and machines. Optional.

Children

A QueryGroup definition can include zero or more Userref definitions. Each Userref refers to a previously created User definition that identifies one InForm user.

Example

In the following QueryGroup, users named Kevin and George can each close queries initiated by the other user.

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

ReportingGroup

Purpose

Note: You cannot create reporting groups in the user interface. You must use MedML to define reporting groups.

Defines the reporting functionality and type of access available to users with reporting rights. Some reporting groups allow members to access only standard reports; others allow members access to the InForm Ad Hoc Reporting workspace

Note: The only way to remove a user from a group is through the Admin function of the InForm application. You cannot remove a user by running the MedML Installer utility.

Syntax

```
<REPORTINGGROUP
   GROUPNAME="name"
   GROUPDESCRIPTION="name">
   <USERREF USERNAME="name"/>
</REPORTINGGROUP>
```

Attributes

GROUPNAME="*name*"

Name of the reporting group Required.

GROUPDESCRIPTION="*name*"

Text describing the reporting group Optional.

Children

A ReportingGroup definition can include zero or more Userref definitions. Each Userref refers to a previously created User definition that identifies one InForm user.

Example

The example below shows the syntax for creating a reporting group named Ad Hoc Users.

```
<REPORTINGGROUP
   GROUPNAME="Ad Hoc Users"
   GROUPDESCRIPTION="Ad Hoc Users">
   <USERREF USERNAME="cra"/>
</REPORTINGGROUP>
```

Resource

Purpose

Identifies a single resource to load.

Syntax

```
<RESOURCE
   [UUID="id"]
   [FILENAME="file"]
   [DESCRIPTION="text"]
   DATATYPE="TEXT|GIF|JPEG"
   [LANGUAGE="name"]
   [TEXT="text"]
   [DATA="data"]
   [CLIENTID="id"]/>
```

Attributes

UUID="*id*"

Universally Unique Identifier; a string that identifies the component uniquely across all studies, study databases, and machines. Either a **UUID** or a **FILENAME** attribute is required unless you include explicit text or a data stream in the **TEXT** or **DATA** attribute.

FILENAME="*file*"

Name of the file in which the resource is enclosed. Either a **UUID** or a **FILENAME** attribute is required.

DESCRIPTION="*text*"

Description of the resource. Optional.

DATATYPE="TEXT|GIF|JPEG"

Type of data contained in the file: TEXT, GIF, or JPEG. Required.

LANGUAGE="*name*"

Code for the culture and language used to display text values. If you do not include a **LANGUAGE** attribute, the default is the installed default product locale. Required if text strings are translated (if a **TRANSLATIONS** element is included as a child of the study component definition).

> **Note:** Relying on the default product locale is not recommended, because the default product locale can be changed.

TEXT="*text*"

Actual text of a resource for which the **DATATYPE** is TEXT. If you include actual text, do not specify a **UUID** or **FILENAME** attribute.

DATA="*data*"

Data stream that defines a resource for which the **DATATYPE** is GIF or JPEG. When the **DATATYPE** is GIF or JPEG, a **UUID**, a **FILENAME**, or a DATA definition is required. If you include a data stream, do not specify a UUID or FILENAME attribute.

CLIENTID="*id*"

String identifier, reserved for use by the Central Designer application.

Example 1

The following example illustrates two text resources.

```
<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"/>
```

Example 2

The following example illustrates three graphical resources.

```
<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"/>
```

Right

Purpose

Loads definitions of rights that identify specific InForm activities that users can perform. The InForm software is delivered with predefined rights. For a list of the predefined rights in the InForm application, see *Predefined rights in the InForm application* (on page 74).

Syntax

<RIGHT
  RIGHT="*name*"/>

Attributes

RIGHT="name"/>

Name of the right.

Example

The following example illustrates the definitions of some of the rights that are predefined for the InForm application.

<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"/>

Predefined rights in the InForm application

Rights—User Manager RG

- Create User
- Modify User Information
- Manage Rights Groups
- Manage Other Groups
- Manage Sites

Rights—User Activator RG

- Activate Site User
- Deactivate Site User
- Activate Sponsor User
- Deactivate Sponsor User
- Un-terminate User (change to Inactive Status)
- Terminate User

Rights—Auto Query RG

- Change Query State from Candidate to Open
- Change Query State from Candidate to Deleted
- Change Query State from Answered to Closed
- Answer Query
- Enter Query in Candidate State
- Enter Query in Open State
- Re-Issue Query in Candidate State
- Re-Issue Query in Open State

Rights—System Creator Group

- Modify System Configuration
- Create User
- Modify User Information
- Manage Rights Groups
- Manage Other Groups
- Activate Site User
- Deactivate Site User
- Activate Sponsor User
- Deactivate Sponsor User
- Un-Terminate User (change to Inactive Status)
- Terminate User
- Manage Sites
- Print
- Monitor
- Reports
- Enroll Subjects
- View CRF
- Access Data Viewer / Data Export Listings
- Assign Review State 1
- Assign Review State 2
- Assign Review State 3
- Assign Review State 4
- Assign Review State 5
- View Review States
- Enter Data into a CRF
- Edit Data on a CRF
- Enter Comments into a CRF
- Freeze a CRF
- Unfreeze a CRF
- Lock a CRF
- Unlock a CRF
- Mark and Unmark a CRF as SVed

- Mark a CRF as Ready for SV
- Sign a CRF
- Sign a Case Book
- Mark and Unmark a Case Book as Ready for SV
- Freeze a Case Book
- Unfreeze a Case Book
- Lock a Case Book
- Unlock a Case Book
- Change Query State from Candidate to Open
- Change Query State from Candidate to Deleted
- Change Query State from Answered to Closed
- Answer Query
- Enter Query in Candidate State
- Enter Query in Open State
- Re-Issue Query in Candidate State
- Re-Issue Query in Open State
- Deactivate a Rule
- Run a Rule
- Modify a Rule

Rights—InForm Server Group

- Reports
- Enroll Subjects
- View CRF
- Access Data Viewer / Data Export Listings
- Assign Review State 1
- Assign Review State 2
- Assign Review State 3
- Assign Review State 4
- Assign Review State 5
- View Review States
- Enter Data into a CRF
- Edit Data on a CRF
- Enter Comments into a CRF
- Freeze a CRF
- Unfreeze a CRF
- Lock a CRF
- Unlock a CRF
- Mark and Unmark a CRF as SVed
- Mark a CRF as Ready for SV
- Sign a CRF
- Sign a Case Book
- Mark and unmark a Case Book as Ready for SV
- Freeze a Case Book
- Unfreeze a Case Book
- Lock a Case Book
- Unlock a Case Book
- Deactivate a Rule
- Run a Rule
- Reordering Of Subjects

Rights—PFArchUser Rights Group

- Create User
- Modify User Information
- Manage Rights Groups
- Manage Other Groups
- Activate Site User
- Deactivate Site User
- Activate Sponsor User
- Deactivate Sponsor User
- Un-Terminate User (change to Inactive Status)
- Terminate User
- Manage Sites

Rights—Reports Only Rights Group

- Reports

Rights—Integration RG

- ODM Submit

Rights—Deployment RG

- Study Deployment

Rightref

Purpose

Allows you to include previously defined rights in the definition of a rights group. A Rightref appears only as the child of a rights group in which it is included; it is not submitted as a stand-alone component.

Syntax

<RIGHTREF
  RIGHT="*name*"/>

Attributes

RIGHT="name"/>

Name of the right referenced by the Rightref. This is the same name as specified in the Right attribute of the Right definition.

Example

The following definition of the PF Admin Rights Group, which identifies the activities that a study administrator can perform, illustrates how to use Rightrefs to include Right definitions in a rights group.

```
<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

Purpose

Allows you to create a set of rights and to assign the set of rights to users. The InForm software is delivered with a set of predefined rights and rights groups.

Syntax

```
<RIGHTSGROUP
  GROUPNAME="name"
   [GROUPDESCRIPTION="text"]
   [UUID="id"]>
   <RIGHTREF* attributes/>
   <USERREF* attributes/>
   <ITEMGROUPREF* attributes/>
</RIGHTSGROUP/>
```

Attributes

GROUPNAME="*name*"

Name of the rights group. Required.

Note: A rights group name cannot be the same as an existing group name.

GROUPDESCRIPTION="*text*"

Text describing the rights group. Optional.

UUID="*id*"

Universally Unique Identifier; a string that identifies the component uniquely across all studies, study databases, and machines. Optional.

Children

A RightsGroup definition can include zero or more:

- Rightref definitions. Each Rightref refers to a previously created right definition that identifies an InForm activity that users can perform.
- Userref definitions. Each Userref refers to a previously created user definition that identifies one InForm user.
- ItemGroupref definitions. Each ItemGroupref refers to a previously created item group definition that identifies a group of items for which a user can set a rights group-specific display override.

Example

The following example illustrates the definition of the PF Admin Rights Group, which identifies the activities that a study administrator can perform. This definition also assigns the administrator rights set to the user named admin.

```
<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>
```

In the following example, the CRC RG rights group includes the definitions of the CRC user and the CRC_Hidden ItemGroupref.

```
<RIGHTSGROUP GROUPNAME="CRC RG">
  <RIGHTREF RIGHT="Print" />
  <RIGHTREF RIGHT="Reports" />
  <RIGHTREF RIGHT="Enroll Subjects" />
  <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>
```

Sequence

Purpose

Specifies a numbering sequence. Sequences allow sponsors to indicate how numbers are assigned to screened subjects, enrolled subjects, queries, and other types of numbered data. Each sequence is associated with a sequence type, defined with the SequenceType element.

Syntax

```
<SEQUENCE
  UUID="id"
   SEQUENCENAME="name"
   SEQUENCETYPENAME="name"/>
```

Attributes

UUID="*id*"

Universally Unique Identifier; a string that identifies the component uniquely across all studies, study databases, and machines. Required.

Note: The InForm application and the MedML Installer utility convert alphabetic characters in UUIDs to uppercase.

SEQUENCENAME="*name*"

Descriptive name of the type of number for which you are creating a sequence definition. Required.

SEQUENCETYPENAME="*name*"

Name of the sequence type to which the sequence number belongs. Required.

Example

The following example shows the XML elements used to define query, enrollment, screening, and randomization number sequences.

```
<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

Purpose

Defines the types of entities for which the InForm application generates sequential numbers. When you create a definition of the numerical sequence by using the Sequence element, you specify a type to which the sequence belongs by including a SequenceType definition. The InForm software defines the following types of sequence numbers:

- Screening
- Enrollment
- Query
- Randomization

Syntax

```
<SEQUENCETYPE
  [UUID="id"]
   SEQUENCETYPENAME="name"/>
```

Attributes

UUID="*id*"

Universally Unique Identifier; a string that identifies the component uniquely across all studies, study databases, and machines. Optional.

SEQUENCETYPENAME="*name*"

Name of the sequence type. Required.

Example

The following SequenceType definitions create the base sequence types used in the InForm software:

```
<SEQUENCETYPE SEQUENCETYPENAME="Enrollment"/>
<SEQUENCETYPE SEQUENCETYPENAME="Screening"/>
<SEQUENCETYPE SEQUENCETYPENAME="Query"/>
<SEQUENCETYPE SEQUENCETYPENAME="Randomization"/>
```

SignatureGroup

Purpose

Specifies a set of users who can sign case books or CRFs.

Notes:

You cannot create signature groups in the user interface. You must use the Central Designer application or MedML to define signature groups.

The only way to remove a user from a group is through the Admin function of the InForm application. You cannot remove a user by running the MedML Installer utility.

Syntax

```
<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>
```

Attributes

GROUPNAME="*name*"

Name of the signature group. Required.

GROUPDESCRIPTION="*text*"

Text describing the signature group. Optional.

UUID="*id*"

Universally Unique Identifier; a string that identifies the component uniquely across all studies, study databases, and machines. Optional.

LANGUAGE="*name*"

Code for the culture and language used to display text values. If you do not include a **LANGUAGE** attribute, the default is the installed default product locale. Required if text strings are translated (if a **TRANSLATIONS** element is included as a child of the study component definition).

Note: Relying on the default product locale is not recommended, because the default product locale can be changed.

CRFTEXT="*text*"

Text of the Electronic Signature Affidavit displayed to members of the signature group when signing a CRF associated with the signature group. Specify either the **CRFTEXT** or **CRFFILE** attribute. If you do not specify either attribute, the MedML Installer utility uses the text provided in a default CRF signing text resource. The text must include:

- What the signature means. For example, attestation to the accuracy of the data provided in the CRF.
- Stated intention of the signer for the electronic signature to be the legal equivalent of a handwritten signature.

Optionally, the text can include string substitution characters (%s) to represent the user's first and last names. See the Example section. You can use the **TRANSLATIONS** attribute to provide translations of the attribute value. Optional.

CRFFILE="*file*"

Path name of an HTML or text file containing the text of the Electronic Signature Affidavit displayed to members of the signature group when signing a CRF associated with the signature group. The path name must be relative to the directory from which you run the MedML Installer utility. Specify either the **CRFTEXT** or **CRFFILE** attribute. If you do not specify either attribute, the MedML Installer utility uses the text provided in a default CRF signing text resource. Optional.

CRFMEANING="*text*"

Text that summarizes the meaning of the signature on the CRF. This text is displayed on the Signature Details page and in the list of completed and required signatures on the CRF. You can use the **TRANSLATIONS** attribute to provide translations of the attribute value.

Note: The text for the meaning of a signature is stored in the language of the site where the CRF was signed. The language does not change if a subject is transferred to another site with a different site study locale.

CRBTEXT="*text*"

Text of the Electronic Signature Affidavit displayed to members of the signature group when signing a case book associated with the signature group. Specify either the **CRBTEXT** or **CRBFILE** attribute. If you do not specify either attribute, the MedML Installer utility uses the text provided in a default case book signing text resource. The text must include:

- What the signature means. For example, attestation to the accuracy of the data provided in the form.
- Stated intention of the signer for the electronic signature to be the legal equivalent of a handwritten signature.

Optionally, the text can include string substitution characters (%s) to represent the user's first and last names. See the Example section. You can use the **TRANSLATIONS** attribute to provide translations of the attribute value. Optional.

CRBFILE="*file*"

Path name of an HTML or text file containing the text of the Electronic Signature Affidavit displayed to members of the signature group when signing a case book associated with the signature group. The path name must be relative to the directory from which you run the MedML Installer utility. Specify either the **CRBTEXT** or **CRBFILE** attribute. If you do not specify either attribute, the MedML Installer utility uses the text provided in a default case book signing text resource. Optional.

CRBMEANING="*text*"

Text that summarizes the meaning of the signature on the case book. This text is displayed on the Signature Details page and in the list of completed and required signatures on the CRF used for signing a case book. You can use the **TRANSLATIONS** attribute to provide translations of the attribute value.

UPDATE="*true*|*false*"

Perform an incremental update for the specified signature group by modifying only the properties that you specify.

Children

- Zero or more Userref definitions. Each Userref refers to a previously created User definition that identifies one InForm user.
- Zero or one Translations definition that specifies a set of translated strings for the translatable attributes of the study component. A Translations element must include one or more Translation definitions. Each Translation element represents the translation of one attribute of a study component into the language of one locale. The Translations element must be the last child element in the study component definition. In a SignatureGroup definition, you can translate the **CRFTEXT**, **CRBTEXT**, **CRFMEANING**, and **CRBMEANING** attributes. If you translate any attributes, include translation definitions for all translatable attributes that are part of the study component definition.

> **Note:** Translations are not supported for the **CRFFILE** and **CRBFILE** attributes.

Example

In the following SignatureGroup, users dobrien and lhill can sign.

```
<SIGNATUREGROUP GROUPNAME="CRA Signature">
   <USERREF USERNAME="dobrien"/>
   <USERREF USERNAME="lhill"/>
</SIGNATUREGROUP>
```

The following SignatureGroup definition illustrates the inclusion of the signing text for a CRF. The two %s characters specify that the displayed Electronic Signature Affidavit should include the user's first and last names. Note that the <font> tag attributes must be in single quotes because they appear within the double quotes of the CRFTEXT attribute. Additionally, the SignatureGroup definition includes the definition of a translation string for the CRFTEXT attribute.

```
<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="dobrien"/>
      <USERREF USERNAME="lhill"/>
      <TRANSLATIONS>
         <TRANSLATION NAME="CRFTEXT" LOCALE="fr-FR"
         DISPLAYTEXT="&lt;font name='Arial' size='2'&gt;Par ma signature datée
         ci-dessous,
         Moi, %s %s, je confirme que ce cas formulaire de rapport affiche avec
         précision
         les résultats des examens, des tests, des évaluations et des
         constatées dans les traitements. &lt;br&gt; &lt;br&gt;
         Conformément à la Section de 11,100 Titre 21 du Code of Federal
         Règlement, il s'agit de certifier que j'ai l'intention que ce
         signature électronique doit être juridiquement contraignant équivalent
         de ma signature manuscrite. &lt;br&gt; &lt;br&gt;
         Pour cela, je ne témoignent 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

Purpose

Identifies forms that require signature and specifies which signature group a user must belong to in order to be able to sign the CRF.

Syntax

```
<SIGNCRF
   SIGNATUREGROUPNAME="name"
   FORMREFNAME="name"
   [RESETFORMSTATE="true|false"]
   [INVALIDATIONLEVEL="USER|GROUP"]
   [FINALCRF="true|false"]/>
```

Attributes

SIGNATUREGROUPNAME="*name*"

RefName of a signature group that contains users who are authorized to sign the form specified in the FORMREFNAME attribute. Required.

FORMREFNAME="*name*"

RefName of a CRF that must be signed by a user who is a member of the SignatureGroup specified in the SIGNATUREGROUPNAME attribute. Required.

Note: The Screening and Enrollment CRFs are not intended to be signed. Do not include their RefNames in a SignCRF definition.

RESETFORMSTATE="*true*|*false*"

Indicates whether associating a new signature group with a CRF that is fully signed resets the state of the CRF to not fully signed. When a CRF is reset, the original signatures remain valid. However, the CRF must now be signed by a member of the newly-associated signature group as well. False is the default. Optional.

INVALIDATIONLEVEL="USER|GROUP"

Specifies whether a signature should be invalidated when a data item is imported after the CRF has been signed, for example, when a coded value is imported with the Central Coding application. The options are:

- **USER**—The CRF or case book signature is invalidated if the user who signed can view the item being imported.
- **GROUP**—The CRF or case book signature is invalidated if at least one user in the signature group can view the item being imported.

If you do not specify the **INVALIDATIONLEVEL** attribute, the InForm application invalidates the signature whenever the form is edited, either directly or by import. Optional.

FINALCRF="*true*|*false*"

Indicates whether signing the form specified in the **FORMREFNAME** attribute signs entire the case book. False is the default. Optional.

Example

The following example illustrates the use of the SignCRF element to designate the DEM, VS, and SC forms as requiring signature by a member of the CRA Signature group. The FINALCRF attribute indicates that signing the SC form signs the case report book.

```
<SIGNCRF SIGNATUREGROUPNAME="CRA Signature" FORMREFNAME="DEM"/>
<SIGNCRF SIGNATUREGROUPNAME="CRA Signature" FORMREFNAME="VS"
   INVALIDATIONLEVEL="USER"/>
<SIGNCRF SIGNATUREGROUPNAME="CRA Signature" FORMREFNAME="SC"
   FINALCRF="true"/>
```

Site

Purpose

Defines a study location.

Note: You cannot define sites in the user interface. You must use MedML to define site groups.

Syntax

```
<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"]
   [UPDATE="TRUE|FALSE"]
   [APPLYLATESTSTUDYVERSION="TRUE|FALSE"]>
</SITE>
```

Attributes

NAME="*name*"

Name of the site. Required.

MNEMONIC="*name*"

Abbreviated name with which to refer to the site. Required.

ADDRESS="*addr1*"

First line of the site address. Optional.

ADDRESS2="*addr2*"

Second line of the site address. Optional.

CITY="*name*"

City in which the site address is located. Optional.

STATE="*name*"

State in which the site address is located. Optional.

PROVINCE="*name*"

Province in which the site address is located. Optional.

ZIPCODE="*code*"

Site Zip code. Optional.

POSTCODE="*code*"

Site postal code. Optional.

COUNTRY="*name*"

Country in which the site address is located. Optional.

PHONE="*num*"

Site telephone number. Optional.

ALTPHONE="*num*"

Site alternate telephone number. Optional.

FAX="*num*"

Site fax number. Optional.

EMAIL="*addr*"

E-mail address used for contacting the site. Optional.

TIMEZONE="*name*"

Time zone in which the site is located, used to convert from internal universal system time to local time. The value for this attribute must be one of the following:

- One of the sub-key names listed in the Windows registry key HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Time Zones.

  This option ensures that the SITE MedML is processed for any operating system locale.

- The Display name of the InForm server operating system locale.

  This option allows the SITE MedML to be processed only on an operating system locale that matches the Display name.

Note: Occasionally, time zones can change during standard Windows updates. This can cause time zones associated with existing sites to become invalid and invalid time zone alerts to appear in the user interface. Use this attribute to clear the alerts.

Required.

STARTDATE="*date*"

Date that the site came online. Users cannot add data for a site before the specified date. Note the following considerations for specifying date information:

- Year values must be between 100 and 9999, inclusively. Always enter the full year, even in abbreviated date formats.
- Many date and time formats are valid. The following table gives examples:

Format	Example
"dd month yyyy"	"25 January 1996"
"hh:mm:ss" (12- hour clock)	"8:30:00"
"hh:mm:ss" (24- hour clock)	"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"

Required.

ENDDATE="*date*"

Date that the site came offline. For example, the date that the last subject was signed off and locked. Optional.

SITESERVER="*server name*"

Name of the server designated as the site server. The site server is dedicated for specific activities such as randomization, screening and enrollment, and generating subject numbers.

SITEDATEFORMAT="MONTH_DAY_YEAR"

The format of the date as you want it to be displayed for the site, if a format isn't specified at the user level. Optional.

STUDYLOCALE="*text*"

Code for the preferred study locale of the site. The study locale is the locale in which the study metadata is defined, including visit names, CRF names, section labels, questions, and control labels. The value must match the study locale for a study version in the study. Required.

Note: After you add a subject to a site, you cannot change the site study locale from the user interface or through MedML.

SVAUTOSELECTRATE="n"

Number that corresponds to the percent of forms marked SV Required, which you must source verify.

SVFIRSTNSUBJECTS="n"

The number of subjects to source verify in order, based on the time when the subject is assigned the status Enrolled. For example, to indicate that you must source verify the first seven subjects enrolled in a study, specify SVFIRSTNSUBJECTS=7.

USERNAMEORDER="F,L|L,F"

Order in which the user's given and surnames are presented in a signature affidavit. The default is the user name order specified for the study on the System Configuration page. The order specified for the site overrides the order specified for the study.

- **F,L**—Given name followed by surname.
- **L,F**—Surname followed by given name.

Optional.

UPDATE="TRUE|FALSE

Perform an incremental update for the specified site by modifying only the properties that you specify.

APPLYLATESTSTUDYVERSION="TRUE|FALSE"

When you use the MedML Installer utility to add a new site, specify whether to associate the latest study version with the site:

- TRUE—The latest study version is associated with the site.
- FALSE—No study version is associated with the site.

  If the APPLYLATESTSTUDYVERSION attribute is not included, then no study version is associated with the site.

If the latest study version is not applied to the site, the association must be created through the InForm user interface or by using the STUDYVERSIONSITE MedML element.

Example

The following example defines a site for Clínica Ortopédica in Valencia, Spain, including a user whose Username is Dr Cortina.

```
<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"
   APPLYLATESTSTUDYVERSION="TRUE">
</SITE>
```

SiteGroup

Purpose

Specifies a group of users who have access to a named site.

Note: The only way to remove a user from a group is through the Admin function of the InForm application. You cannot remove a user by running the MedML Installer utility.

Syntax

<SITEGROUP
  SITENAME="*name*">
  <USERREF* *attributes*/>
</SITEGROUP>

Attributes

SITENAME="*name*"

Name of the site to which the SiteGroup gives access. This name corresponds to the **NAME** attribute in the Site definition for the site. Required.

Children

A SiteGroup definition can include zero or more Userref definitions. Each Userref refers to a previously created User definition that identifies one InForm user.

Example

In the following example, users Marge and Jonah are assigned to a SiteGroup for the Beth Israel site.

```
<SITEGROUP SITENAME="Beth Israel">
  <USERREF USERNAME="Marge"/>
  <USERREF USERNAME="Jonah"/>
</SITEGROUP>
```

StudyVersionSite

Purpose

Records the study version that a site is currently using. This component needs to be updated each time a study version changes in a way that does not require Institutional Review Board (IRB) approval, and whenever IRB approval is received for a study version change that does require approval.

Note: Before you can use the StudyVersionSite component to specify the study version that a site is using, you must run the XML file that loads site definitions.

Syntax

<STUDYVERSIONSITE
VERSIONDESCRIPTION="*n*"
[SITENAME="*name*"]
[SITEMNEMONIC="*name*"]
[ACCEPTDATE="*date*"]/>

Attributes

VERSIONDESCRIPTION="*n*"

Number of the study version currently in use at the site. This number must match the **VERSIONDESCRIPTION** attribute with which the study version is defined. Required.

SITENAME="*name*"

Name of the site. Either the **SITENAME** or the **SITEMNEMONIC** attribute is required. This name must match the **NAME** attribute with which the Site is defined.

SITEMNEMONIC="*name*"

Abbreviated name of the site. This name must match the **MNEMONIC** attribute with which the site is defined. Either the **SITENAME** or the **SITEMNEMONIC** attribute is required.

ACCEPTDATE="*date*"

Date of IRB approval of a study version change.

Note: The **ACCEPTDATE** attribute is for documentation purposes and to differentiate study versions for repeating form display. The actual date that a site moves to a new study version is the date that the StudyVersionSite component for the site is revised in the database.

Example

This example illustrates a set of StudyVersionSite components, one for each of five sites.

```
<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" />
```

SVCriticalForm

Purpose

Adds a form to the Critical Forms list.

Syntax

```
<SVCRITICALFORM
    FORMREFNAME="name"
    SITEMNEMONIC="name"
    CRITICAL="true|false"
</SVCRITICALFORM>
```

Attributes

FORMREFNAME="*name*"

RefName of the form to add to the Critical Forms list. Required.

SITEMNEMONIC="*name*"

Mnemonic for the site with which the form is associated. Required.

CRITICAL="*true*|*false*|*studydefault*"

Indicates whether to mark the form as critical for source verification. The options are study default, true, or false. Required.

The study default option allows you to reset a previously-defined override for the form. If you specify **study default** for the CRITICAL attribute, the SDVCRITICAL attribute of the FORM element determines the critical setting for the form.

SVCriticalItem

Purpose

Overrides the critical setting for an item.

Syntax

```
<SVCRITICALITEM
ITEMREFNAME="name"
CRITICAL="study default|true|false"
FORMREFNAME="name"
SITEMNEMONIC="name"
</SVCRITICALITEM>
```

Attributes

ITEMREFNAME="*name*"

RefName of the item to mark as critical. Required.

CRITICAL="s*tudy default*|*true*|*false*"

Indicates whether to mark the item as critical for source verification. The options are study default, true, or false. Required.

The study default option allows you to reset a previously-defined override for the item. If you specify **study default** for the CRITICAL attribute, the critical setting for the item is determined by the SDVCRITICAL attribute of the Item element.

FORMREFNAME="*name*"

RefName of the form on which the item to mark as critical exists. Optional.

SITEMNEMONIC="*name*"

Mnemonic of the site with which the item is associated. Optional.

- To mark an item as critical across all sites, use the SVCRITICALITEM element, and do not specify a SITEMNEMONIC.
- To mark an item as critical for one site, do one of the following:
  - Use the SVCRITICAL element, and specify a SITEMNEMONIC.
  - Use the SVCRITICAL setting for the ITEM.
  - Use the InForm user interface.

SysConfig

Purpose

Specifies the setting of an InForm configuration variable.

Syntax

```
<SYSCONFIG
    CONFIGNAME="name"
    TYPE="0"
    VALUE="text" />brian
```

Attributes

CONFIGNAME="*name*"

Name of the configuration variable. Required.

The InForm application has the following configuration variables:

- **AllowPasswordReuse**—1 (yes) or 0 (no), indicating whether users can change to a previously used password when performing password updates. The default is 1.
- **AllowedRuleObjects**—Specifies the scripting objects that can be called by a user-defined rule or execution plan. The default objects are CDO.Message and CDONTS.NewMail.

  Note: You must change the HKEY_LOCAL_MACHINE\SOFTWARE\OracleHS\InForm\WhiteListScriptObjects registry key to 0 (zero) before you can update the allowed rules object list. Before modifying the list of allowed rule objects, contact Oracle Global Support for assistance.

- **AutoAnswerManualQueries**—1 (on) or 0 (off), indicating whether the InForm application automatically answers a manual query when a data item change satisfies the rules on the data item. The default is 1.
- **CookServer**—Name of the server used for installing MedML metadata definitions.
- **DaysPasswordExpiration**—Number of days that can pass before the InForm software requires users to change their passwords. The default is 30.
- **EnableForgotPassword**—1 (yes) or 0 (no), indicating whether to enable the feature that lets users request a password reset if they have forgotten their password. The default is 1.
- **EmailForForgotPasswordNotification**—The email address of an administrator who receives notification when a user requests a password reset.
- **EmailForNewSiteAndUserNotification**—The email address of an administrator who receives notification when a new site or new user is added.
- **EnforceVisitDate**—1 (yes) or 0 (no), indicating whether to require the use of Date of Visit on the first form of every visit. The default is 0. Read only.
- **EnrollWithIncompleteForms**—1 (yes) or 0 (no), indicating whether the InForm application permits a subject to be enrolled with incomplete screening or enrollment information, after override authorization. The default is 0.

- **ExePlanServer**—The name of the server(s) defined as the server(s) on which execution plans run.
- **FORMSAVEMODE**—The location and format of the "Form submitted successfully" message.
  - Inline: message displays in the header of the form.
  - Popup: message displays as a pop-up. User must click OK in order to proceed. Default.
- **InactivateRetryCount**—Number of failed login attempts to allow before inactivating the user account. The default is 3.
- **INLINEDURATION**—1-9, Specifies the number of seconds that the 'Form submitted successfully" message remains visible in the header of the form before it fades. Applies when FORMSAVEMODE is set to Inline.
- **MaxNumOfResubmissions**—Maximum number of times to retry submission of a failed execution plan before it is logged as an error in the event log and removed from the queue of execution plans to be run. The default is 2.
- **MinPasswordLength**—Minimum number of characters required for passwords. The default is 6.
- **MinutesReauthenticate**—Number of minutes of inactivity that can pass before the InForm software requires a user to log in again. The default is 5.
- **MinutesReIdentification**—Number of minutes that a session can be active before the InForm software requires a user to log in again. The default is 120.
- **NavigationMode**—1 (enable) or 0 (disable), indicating whether to allow use of the subject ordering feature. When enabled, users who have the Reordering Of Subjects right can change the order of subjects as displayed in the InForm user interface. The default is 0.
- **NumOfExePlanListenThreads**—Number of threads running in the background to process pending execution plans. The default is 4; at least 1 is required for any execution plans to run.
- **OneNonAlphaNumericCharacter**—1 (yes) or 0 (no), indicating whether passwords must include at least one special character. The default is 0.
- **OneNumericalCharacter**—1 (yes) or 0 (no), indicating whether passwords must include at least one numeric character. The default is 0.
- **OneUppercaseCharacter**—1 (yes) or 0 (no), indicating whether passwords must include at least one uppercase character. The default is 0.
- **PatientSequence**—Format for assigning subject numbers. Read only.
- **PostQueryForConflictResolution**—1 (yes) or 0 (no), indicating whether to create a query when during synchronization data is found to be entered into a data item by two different servers. The default is 1.
- **QueryMaxLength**—Maximum number of characters of query text displayed below an item on a CRF. The default is 350.
- **QUERYSELECTION**—Specifies whether queries are created in Opened or Candidate state.
- **RandCentralStratified**—Sequence number format for Central Stratified randomization schemes (multiple drug kit lists, patient assigned a drug kit based on stratification criteria).

- **RandomizationSrc**—Name of the randomization source manager (COM object) that accesses the default randomization source database. The default name is "Inform.PFRandomization.1." Read only.
- **RandSimpleCentral**—Sequence number format for Simple Central randomization schemes (one central drug kit list from which numbers are assigned sequentially). Read only.
- **RandSimpleCentralSRC**—Name of the randomization source manager (COM object) that accesses the randomization source database for Simple Central randomization schemes. Read only.
- **RandSimpleSite**—Sequence number format for Simple Site randomization schemes (drug kit list for each site, patient assigned sequentially to drug kit on list for their site). Read only.
- **RandSimpleSiteSRC**—Name of the randomization source manager (COM object) that accesses the Simple Site randomization source database. Read only.
- **RandStratifiedBySite**—Sequence number format for Site Stratification randomization schemes (multiple drug kit lists for each site, patient assigned a drug kit based on site and stratification criteria). Read only.
- **RandStratifiedBySiteSRC**—Name of the randomization source manager (COM object) that accesses the Site Stratification randomization source database. Read only.
- **REPORTINGINTERNALURI**—Cognos parameter that is set when running the CRNConfig installer. It is an internal URI that the InForm server uses to communicate to the Cognos server. The information can be found in cogstartup.xml. Example: http://appsru02.north.pf.com:9300/p2pd/servlet/dispatch.
- **REPORTINGSERVER**—The URL for the Cognos Web service.
- **REPORTINGAUTHENTICATIONNAMESPACE**—The LDAP namespace that is used to authenticate InForm users on the reporting server.
- **REPORTINGUSERROOT**—The top-level reporting folder for the company. Use this field only if you are hosting several companies on one reporting server and have set up reporting folders for each company. Leave this field blank if you are not hosting studies for different companies, or if you have not set up separate reporting folders for each company.
- **RequireCommentForNA**—1 (yes) or 0 (no), indicating whether the InForm system requires a user to enter a comment when entering N/A, Unknown, or Not Done in response to a question on a form. The default is 0.
- **ScreeningSequence**—Sequence number format for assigning screening numbers. Read only.
- **SponsorEditFrozen**—1 (yes) or 0 (no), indicating whether sponsors can edit a form after it has been marked as frozen. The default is 0.
- **SSLFlag**—1 (on) or 0 (off), indicating whether Secure Socket Layer is enabled to provide encryption of data. The default is 1.

> **Note:** Before this option can take effect, you must stop and restart the study.

- **TrialDateFormat**—"Month_Day_Year", "Day_Month_Year" or "Year_Month_Day", indicating the format in which you want the date to appear in the trial. The default is Month_Day_Year.
- **TrialOrgID**—Company identifier for the associated with the study.

  Note: Reserved for Oracle-hosted studies in a single sign-on environment.

- **UNC_DownloadDirectory**—Full path name of the physical InForm server directory used to download data listings with the Listings button. Read only.
- **UniqueIntlDOBSwtch**—Study, site or none, indicating whether the InForm application requires a unique combination of subject initials and date of birth for a study, a site, or not at all:
  - 0 (default)—Initials and DOB combination is not required to be unique.
  - 1—Initials and DOB combination must be unique within a site.
  - 2—Initials and DOB combination must be unique within a study.

  Note: If you specify that unique initials and date of birth is not required and subjects with duplicate initials and date of birth are entered and then you specify that unique IDs are required, the previously entered duplicate information will not be reported. If you plan to allow the transfer of subjects from one site to another, be aware that if a user transfers a subject to a site where another subject exists with the same initials and date of birth, and the study does not require unique initials and date of birth or only requires site uniqueness, the subject transfer fails. The user must change the subject initials to make the combination unique.

  To prevent this situation, set the UniqueIntlDOBSwitch attribute to require unique initials and DOB across the study.

- **UniquePatIDSwtch**—Study, site, or none, indicating whether the InForm application requires a unique subject ID for a study, a site or not at all:
  - 0—Subject ID is not required to be unique.
  - 1—Subject ID must be unique within a site.
  - 2 (default)—Subject ID must be unique within a study.

  Notes:

  If you are using the Data Viewer in the InForm application, you must configure your system to require unique subject IDs within each site or each study.

  If you specify that unique initials and date of birth is not required and subjects with duplicate initials and date of birth are entered and then you specify that unique IDs are required, the previously entered duplicate information will not be reported. If you plan to allow the transfer of subjects from one site to another, be aware that if a user transfers a subject to a site where another subject exists with the same initials and date of birth, and the study does not require unique initials and date of birth or only requires site uniqueness, the subject transfer fails. The user must change the subject initials to make the combination unique.

  To prevent this situation, set the UniqueIntIDOBSwitch attribute to require unique initials and DOB across the study.

- **UserNameOrder**—F,L or L,F, indicating the order in which the user's given and surnames are presented in a signature affidavit. The default is the user name order specified for the study on the System Configuration page. The order specified for the site overrides the order specified for the study.
- **ViewCRFSignList**—1 (yes) or 0 (no), indicating whether a list of required signatures should appear on each CRF for which a signature is required. The default is 1. Read only.
- **Virtual_DownloadDirectory**—Full path name of the InForm server virtual directory used to download data listings with the Listings button. Read only.
- **VISITCALCULATORENABLED**—1 (yes) or 0 (no), indicating whether to enable the visit calculator for a subject after the subject is enrolled. The default is 0.

Note: You can set the Server Friendly Name parameter, which provides a user-friendly server name to be displayed on Query Details and Signing Details screens, only through the Admin user interface of the InForm application. There is no equivalent MedML variable.

TYPE="0"

0 is the only value currently accepted. Required.

VALUE="*n*"

Value to assign to the configuration variable. Required.

Example

```
<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="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="FORMSAVEMODE" VALUE="1"/>
<SYSCONFIG CONFIGNAME="INLINEDURATION" VALUE="3"/>
<SYSCONFIG CONFIGNAME="ViewCRFSignList" VALUE="1"/>
<SYSCONFIG CONFIGNAME="QUERYSELECTION" VALUE="QUERYACTION_CREATEOPEN"/>
```

Translation

Purpose

Specifies a translated text string that is associated with an attribute of a study component, along with the locale in which the translated text is used. Use one Translation element for each combination of attribute and locale that is needed in the study version. If you translate any attributes, include translation definitions for all translatable attributes that are part of the study component definition.

The Translation element is a child of the Translations element.

Syntax

```
<TRANSLATION
   NAME="attributename"
   LOCALE="text"
   DISPLAYTEXT="text"
   />
```

Attributes

NAME="*attributename*"

Name of the study component attribute for which translated text is being supplied. Required.

LOCALE="*text*"

Code for the language and culture of the translated text. Required.

DISPLAYTEXT="text"

Translated text string. Required.

Examples

The following example illustrates the translation of an item question and label into Spanish and German.

```
<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

Purpose

Defines a set of translated strings for the translatable attributes of a study component. A Translations definition is a child of the study component for which it provides translated strings. The Translations element must be the last child element in the study component definition. If you translate any attributes, include translation definitions for all translatable attributes that are part of the study component definition.

Note: When you use a Translations element, the **LANGUAGE** attribute of the associated study component is required.

Syntax

```
<TRANSLATIONS>
   <TRANSLATION* attributes/>
</TRANSLATIONS>
```

Children

A Translations element must include one or more Translation definitions. Each Translation element represents the translation of one attribute of a study component into the language of one locale.

Examples

The following example illustrates the translation of an item question and label into Spanish and German.

```
<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/>
```

User

Purpose

Defines a person who has access to an InForm study database. User access to specific aspects of a study are determined by the user's:

- Site group—Gives the user access to a specific study site.
- Rights group—Gives the user access to a set of rights to perform specific activities.
- Query group—Gives a user who has been assigned the right to close queries the additional right to close queries initiated by another member of the same query group.
- Signature group—Gives a user the right to sign documents requiring signature. To sign a site's documents, a user must be in a signature group and the appropriate site group.

A user can also be included in the definition of a Sponsor or Site. Inclusion in either of those definitions indicates that the user is a contact at the sponsor or site location.

If you attempt to install information about an existing, active user, including the user's password, the InForm application makes the user inactive as a security precaution.

Note: If you use the **IMAGEFILE** attribute, you must load study definition files into the database after you load the resource XML files that define user images.

Syntax

```
<USER
   USERNAME="name"
   USERTYPE="SYSTEM|SITE|SPONSOR|INTEGRATION|SUPPORT"
   [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"/>
```

Attributes

USERNAME="*name*"

Name that identifies the user in the database. Required.

USERTYPE="SYSTEM|SITE|SPONSOR|INTEGRATION|SUPPORT"

Type of user.

- **SYSTEM**—User with specialized system capabilities. For example, when the InForm application generates a query automatically, the user name assigned as the query originator is "autoquery." The autoquery user is a system user.
- **SITE**—User associated with a site.
- **SPONSOR**—User associated with a sponsor.

- **INTEGRATION**—User for integrations with external applications, such as the Clinical Data API.

  An Integration user:

  - Is restricted so that it can be assigned rights associated only with integration activities. This user type cannot have any other rights.

    For example, in this release, an Integration user can have only the ODM Submit right, which allows communication between the Clinical Data API and the InForm application.

  - Has no access to any clinical data; this user cannot see or change any of the study clinical data through the user interface.
  - Cannot be assigned to a site.
  - Cannot have any administration rights.
  - Can log in to the InForm user interface but cannot do anything other than change his user information.
  - Can be created in the InForm Admin user interface or with the MedML Installer utility.

    After an Integration user type is created, the user type becomes read-only in the user interface. It can only be changed to another user type with the MedML Installer utility.

- **SUPPORT**—User typically responsible for support or troubleshooting tasks. Support users behave like all other InForm sponsor users, except they cannot edit their user name or user type.

Required.

FIRSTNAME="*name*"

Given name of the user. Optional.

LASTNAME="*name*"

Surname of the user. Optional.

DISPLAYNAME="*name*"

User name as displayed in the navigation toolbar in the InForm application. Maximum length is 63 characters; shorter strings are recommended. Optional.

DESCRIPTION="*text*"

User description; for example, user's role in the study. Optional.

TITLE="*name*"

User title. Optional.

ADDRESS="*addr1*"

First line of the user address. Optional.

ADDRESS2="*addr2*"

Second line of the user address. Optional.

CITY="*name*"

City of the user address. Optional.

STATE="*name*"

State of the user address. (Not for use with Province.) Optional.

PROVINCE="*name*"

Province of the user address. (Not for use with State.) Optional.

ZIPCODE="*code*"

User zip code. (Not for use with Postcode.) Optional.

POSTCODE="*code*"

User postal code. (Not for use with Zipcode.) Optional.

COUNTRY="*name*"

Country of the user address. Optional.

PHONE="*num*"

User telephone number. Optional.

ALTPHONE="*num*"

User alternate telephone number. Optional.

FAX="*num*"

User fax number. Optional.

EMAIL="*addr*"

User email address. Optional.

HOMESCREENURL="*url*"

Local or external URL identifying the initial screen that appears when a user logs in to the InForm application. The address must include the http:// prefix and must identify the server on which the file is located by name or by IP address. Optional.

IMAGEFILE="*file*"

Name of the image file that appears on the navigation toolbar in the InForm application. Optional.

IMAGETYPE="GIF|JPEG|TEXT"

Type of image file: GIF, JPEG, or TEXT. Required if you specified a filename for Imagefile.

LANGUAGE="*name*"

User preferred language. English is the default. Optional.

ACTIVESTATE="*true*|*false*"

Indicates whether the user is active. The options are true or false. True is the default. Optional.

DELETESTATE="*true*|*false*"

Indicates whether the user has been terminated. The options are true or false. False is the default. Terminated users remain in the database. Optional.

PASSWORD="*name*"

User password. Optional.

USERDATEFORMAT="MONTH_DAY_YEAR|DAY_MONTH_YEAR|YEAR_MONTH_DAY"

Desired date format for viewable InForm application pages for this particular user.

PRODUCTLOCALE="*en-US*|*ja-JP*"

Code for the user product locale. The product locale is the locale in which the system data in the InForm user interface is displayed. The value must be either en-US (English) or ja-JP (Japanese). Required.

STUDYLOCALE="*text*"

Code for the study locale of the user. The study locale is the locale in which the study metadata is defined, including visit names, CRF names, section labels, questions, and control labels. The value must match the study locale for a study version in the study. Required.

UPDATE="*true*|*false*"

Perform an incremental update for the specified user by modifying only the properties that you specify.

Example 1

The following example adds a user named CRC to the database.

```
<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="gcp123"
   USERDATEFORMAT="MMM/DD/YYYY"/>
```

Example 2

The following example adds a user who is located in Switzerland to the database.

```
<USER USERNAME="jcrecy"
   USERTYPE="SITE"
   FIRSTNAME="Jeanne"
   LASTNAME="Crecy"\
   DISPLAYNAME="J Crecy"
   HOMESCREENURL="/inform1/custom/Defaut.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

Purpose

Allows you to update an image for an existing user. When you use the UserImage element, all other existing user information remains the same.

Note: To update other individual user attributes, use the InForm Admin user interface or use the MedML Installer utility. When you update a User definition with the MedML Installer utility, you must repeat the entire user information set; you cannot update single attributes other than the user image individually.

Syntax

```
<USERIMAGE
   USERNAME="name"
   [IMAGEFILE="file"]
   [IMAGETYPE="GIF|JPEG|TEXT"]
   [LANGUAGE="name"]/>
```

Attributes

USERNAME="*name*"

Name that identifies the user in the database. Required.

IMAGEFILE="*file*"

Name of the image file that appears on the InForm application navigation toolbar. If you do not specify a filename, the MedML Installer utility removes the current image file if one is defined for the user. Optional.

IMAGETYPE="GIF|JPEG|TEXT"

Type of image file: GIF, JPEG, or TEXT. Required if you specified a filename for Imagefile.

LANGUAGE="*name*"

Language of the user image file. English is the default. Optional.

Example

The following example illustrates the use of the UserImage element to add the image in the xena.jpg file to the user with username "sm."

```
<USERIMAGE USERNAME="sm"
   IMAGEFILE="..\Resources\UserPics\xena.jpg"
   IMAGETYPE="GIF"/>
```

Userref

Purpose

Includes previously defined users in the definition of any of the following components:

- QueryGroup
- ReportingGroup
- RightsGroup
- SignatureGroup
- Site
- SiteGroup
- Sponsor

A Userref appears only as the child of an component in which it is included; it is not submitted as a stand-alone component.

Note: The only way to remove a user from a group is through the Admin function of the InForm application. You cannot remove a user by running the MedML Installer utility.

Syntax

```
<USERREF
   USERNAME="name"/>
```

Attributes

USERNAME="*name*"

Name of the user to include in an component, as specified in the **USERNAME** attribute of the User definition.

Example

The following example illustrates the inclusion of the users Homer and Marge in the definition of a query group called SiteXQueries.

```
<QUERYGROUP GROUPNAME="SiteXQueries">
   <USERREF USERNAME="Homer"/>
   <USERREF USERNAME="Marge"/>
</QUERYGROUP>
```

Scripting object reference

Conversion object

The InForm application allows you to define CRF components that you specify as units. You can define unit CRF components with reference to other units, so that data can be displayed online as one unit and stored in the database, after conversion, as the other. For example, you can design a CRF so that a CRC can choose whether to enter a weight in ounces or pounds, and specify that the weight is always stored in the database in ounces. The unit in which data is stored in the database after conversion is called the base unit, and the data value, after conversion to the base unit, is called the normalized value.

To perform conversions between the entered and base unit values, you create rules with a rule type of Conversion. The scripts for these rules can use the Data object.

Data object

The Data object allows you to specify how to perform a conversion between the item entered value and the normalized value stored in the database after conversion. A conversion rule is attached in an XML file to a unit definition, which includes the RefName of the rule.

The Data object has the following properties:

Property	Type	Purpose
Result	FLOAT	Holds the result of the conversion.
BaseValue	NUMERIC	Contains the value to be converted.

Example

The following conversion rule script converts from centimeters to millimeters:

```
Data.Result=Data.BaseValue*10
```

Execution plan objects and methods

Execution plans are associated with events, and an execution plan runs when its event fires. The InForm application supports execution plans that do either of the following:

- Send email.
- Log a message to the Windows log.

A script that is part of the definition of an execution plan can use the objects and methods described in this topic.

Note: You can define execution plans only by using MedML and the MedML Installer utility.

Global object

Allows you to retrieve information about the current site, the user whose action caused the execution plan to run, or the user designated as the primary contact at the current site. Additionally, the Global object allows you to pass a message generated by a rule to an email message or to the Windows log. The Global object has the following properties, which are Get only.

Property	Purpose
Site	Holds the current properties of the Site object, which contains site data.
SiteUser	Holds the current properties of the User object, which contains user data, for the user who is specified as the primary contact for the current site.
User	Holds the current properties of the User object, which contains user data, for the user whose action caused the execution plan to fire.
ContextString	Holds the message received from the Message property of the Patient or Result object. ContextString is a string property.

Global methods

Method	Purpose
EMailToGroup	Sends email to the members of a specified query group, signature group, or user manager group defined in the study database.
EMailToInFormUser	Sends email to a specified InForm user.
EMailToInternetUser	Sends email to any internet email alias.
EMailToUser	For internal use only.
GetNamedValue	Returns the data value associated with a name specified in the AddNamedValue method on the Patient or Result object.
LogMessage	Logs the specified message to the Windows log.
OutputDebug	Not supported.

Site object

Allows you to retrieve information about the current site. The Site object has the following properties, which are Get only:

Property	Type	What the property holds
ContactID	DWORD	Database ID of the InForm user indicated as the primary contact for the site.
SiteName	string	Site name.
Address	string	Site address.
Phone	string	Site phone number.
AltPhone	string	Alternate site phone number.
Beeper	string	Site beeper number.
Email	string	Site email address.
Fax	string	Site fax number.
TimeZone	string	Site time zone.

Site method

Method	Purpose
SetSite	Loads the Site object from cache with data from the site with the specified site ID. For Oracle use only.

User object

Allows you to retrieve information about the current user. The User object has the following properties, which are Get only:

Property	Type	What the property holds
FirstName	string	User given name.
LastName	string	User surname.
Address	string	User address.
Phone	string	User phone number.
AltPhone	string	Alternate user phone number.
Beeper	string	User beeper number.
Email	string	User email address.
Fax	string	User fax number.

Property	Type	What the property holds
UserName	string	User account name used to log in to the InForm application.

User method

Method	Purpose
SetUser	Loads the User object from cache with data from the user with the specified user ID. For Oracle use only.

EMailToGroup(GroupID,Subject,Msg)

EMailToGroup is for internal Oracle use only.

Sends email to the members of a specified query group, signature group, or user manager group defined in the study database.

Arguments

GroupID

Internally assigned database ID of the query group, signature group, or user manager group to receive the email.

Subject

String containing the subject of the email.

Msg

String containing the text of the email message.

Note: The GroupID argument identifies the recipient of the email. The following registry entry specifies the sender (the From address): HKLM/SOFTWARE/OracleHS/InForm/PFMngrExecutionPlan. The entry is a string value named "FromAddress". The default is "nobody@Oracle.com".

EMailToInFormUser "InFormUser,szSubject,szMsg"

Sends email to a specified InForm user.

Arguments

InFormUser

Internally assigned database ID of the InForm user to receive the email.

Subject

String containing the subject of the email.

Msg

String containing the text of the email message.

Note: The InFormUser argument identifies the recipient of the email. The following registry entry specifies the sender (the From address): HKLM/SOFTWARE/OracleHS/InForm/PFMngrExecutionPlan. The entry is a string value named "FromAddress". The default is "nobody@Oracle.com".

Example

Global.EMailToInformUser "sm", "CRFUnLock", "Sending InForm User email when Unlock-CRF"

EMailToInternetUser "EmailAddress,szSubject,szMsg"

Sends email to any email alias.

Arguments

EmailAddress

The email address of the InForm user to receive the email.

Subject

String containing the subject of the email.

Msg

String containing the text of the email message.

Note: The EmailAddress argument identifies the recipient of the email. The following registry entry specifies the sender (the From address): HKLM/SOFTWARE/OracleHS/InForm/PFMngrExecutionPlan. The entry is a string value named "FromAddress". The default is "nobody@Oracle.com".

Example

Global.EMailToInternetUser "user@Oracle.com", "CRFUnLock", "Sending External User email when Unlock-CRF"

GetNamedValue(Name)

Returns the data value associated with a name specified in the AddNamedValue method on the Patient or Result object.

Argument

Name

String containing the name passed by the AddNamedValue method.

Example

```
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"

Logs the specified message to the Windows log.

Argument

Message

String containing the text of the message to log to the Windows log.

Example

```
Global.LogMessage "Query generated"
```

SetSite(SiteID)

SetSite(SiteID) is for internal use only.

Loads the Site object from cache with data from the site with the specified site ID.

Argument

SiteID

Database ID for the site you want to access by using the Site object.

SetUser(UserID)

SetUser(UserID) is for internal use only.

Loads the User object from cache with data from the user with the specified user ID.

Argument

UserID

Database IF for the user you want to access by using the User object.

Randomization objects and methods

When you implement randomization in a study, you must perform several configuration steps. One of these is to create a script that generates a drug kit number to assign to each subject as the subject is enrolled. This randomization script can use the objects and methods that are available for rules and calculations, and can also use the Randomization object and method described in this topic.

Use MedML and the MedML Installer utility to attach the randomization script to a calculated control on the CRF where you want to generate the randomization sequence number and assign a drug kit.

Randomization object

Allows you to assign the subject, site, randomization, and study revision information that the GetNextKit method uses to determine the next drug kit number. The Randomization object has the following properties:

Property	Purpose
PatientID	Holds the ID of the current subject.
SiteID	Holds the ID of the current site. The site ID is an internally assigned value by which the site is known in the database.
Type	Holds the randomization type: • 1 (Simple Central)—The study uses one list of drug kits. Each new subject is assigned the next sequential drug kit number on the list. • 2 (Central Stratified)—The study has multiple lists of drug kits. Each new subject is assigned to a drug kit list based on entered subject data. Then, the subject is assigned the next sequential drug kit number on that list. • 3 (Simple Site)—Each site has a different drug kit list. Each new subject is assigned the next sequential drug kit number on the list for the subject's site. • 4 (Stratified by Site)—Each site has multiple lists of drug kits. Each new subject is first assigned to the set of list for the subject's site. Then, the subject is assigned to one of the site's drug kit lists based on entered subject data. Finally, the subject is assigned the next sequential drug kit number on that list.
Source	Randomization source list name, or stratification code.
Revision	Revision number of the study version.

GetNextKit method (KitInfo)

Gets the next sequence number from the randomization source specified in the Randomization.Source property. If the next number in the sequence has additional kit information associated with it, GetNextKit also returns the corresponding kit information.

Argument

Kit information associated with the returned sequence number, if provided.

Examples

The following example illustrates a randomization rule for a Simple Central (Type 1) randomization scheme.

```
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
```

The following example illustrates a randomization rule for a Central Stratified (Type 2) randomization scheme. In this example, the stratification is based on the subject's weight, as entered in the Demographics form in Visit 1. Based on the subject's weight, the InForm software gets the next sequence number from either the CS_WT150 or the CS_WT275 randomization source list.

```
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
```

The following example illustrates a randomization rule for a Simple Site (Type 3) randomization scheme. In this example, the subject's site determines the randomization source list from which the InForm application gets the next sequence number.

```
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
```

The following example illustrates a randomization rule for a Stratified by Site (Type 4) randomization scheme. In this example, the subject's site and height determine the randomization source list from which the InForm application gets the next sequence number. This example uses the Oracle randomization source lists SR_PF_HT45 and SR_PF_HT75 and the Beth Israel randomization source lists SR_BID_HT45 and SR_BID_HT75.

```
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 method (KitInfo)

Gets the next sequence number from the randomization source specified in the Randomization.Source property. This method differs from the GetNextKitNumber method in that the additional kit information associated with the sequence number is not returned.

Argument

Kit information associated with the returned sequence number, if provided.

CHAPTER 3

InForm Data Import utility

In this chapter

Overview of the InForm Data Import utility

Parts of the import utility

Part	Description
Data file	A pipe-delimited file that contains the following: • Line feed characters and carriage returns between lines. • Data to load into the study database.
Map file	A file that contains all the mapping information that is necessary to import the data file to the InForm database.
XML file	A file that contains information that can be directly imported into the study database. For more information, see *Creating a data and map file* (on page 149) and *Adding new clinical data* (on page 133).
Rules	Information against which the XML file is compared to ensure that it complies with the study standards. You use rules to: • Raise queries about data. • Validate data. **Note:** Running rules might cause the import to run slowly.
InForm application server	The server on which the InForm application is running.
Study database	The database in which the data for the study resides.

Import methods

The following data loading methods are available in the InForm Data Import utility:

- **Importing a data file**—Uses a map file to define mappings between the imported data and the InForm database tables. The InForm Data Import utility then loads the data from a pipe-delimited (|) file into the InForm database. To run the import for this input file format, select the **InForm Data and Map files** option in the InForm Data Import main window.

  When you use this method, the data is processed by the InForm application server, which runs edit checks to validate the data before writing to the database.

- **Importing an XML file**—Loads data from an XML file into the InForm database or transfers selected subject records from one site to another. To run the import for this input file format, select the **MedML file** option in the InForm Data Import main window.

  When you use this method, you can run rules during the import. If you run rules, the application processes the data as if you were entering it online; it runs edit checks and generates queries on data that fails the checks.

Note: The InForm Data Import utility does not support importing Regulatory Documents or Visit Reports data.

Special considerations

- **Importing new subject data**—You must use the MedML file option to import data for a subject who has not gone through the screening and enrollment process.

  You cannot use the data and map file option to import screening and enrollment data.

Note: You cannot import new data to a form with a Frozen or Locked status in the InForm application.

- **Importing comments**—To import form-level comments, you must use the MedML file option.

  To import item-level comments, you must use either the data and map file option or the MedML file option.

- **Importing calculated controls**—When you import calculated controls, the data type definition of the import field must be Text control. If Integer or Floating number is used, the match is not recognized and data is not updated. Instead, the data is added as a new row.

- **Importing units**—When you submit unit data, the units must be associated with the previous field in the data and map file.

- **Importing unscheduled visit data**—To import data to unscheduled visits, use either the MedML file or data and map file option.

- **Importing on multiple servers**—If your study is running on multiple servers, run the InForm Data Import utility on only one server at a time to eliminate the possibility of importing duplicate data.

- **Editing repeating forms**—To create or edit repeating form instances, you must create the map file using a text editor, not the InForm Data Import utility user interface. Follow these guidelines for editing a map file:

  - Use NOFORMNEW in the first line of the map file to instruct the InForm Data Import utility not to create a new repeating form instance.

  - Use the new !formmatch! element in the map file to specify an item (not within an itemset) that will be compared to other repeating form instances.

    You can use multiple !formmatch! elements. If all such elements match some existing repeating form instance, then that form instance will be updated. If there is no match (or no !formmatch! element) then a new repeating form instance will be created, as illustrated in the following sample code:

    ```
    FORMNOVISITNEWNOFORMNEW|
    !cd!Site|
    !cd!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!|
    ```

  - When you create a repeating form, include regular items instead of itemsets to uniquely identify the form. If you cannot uniquely identify the form, then you can enter data only one time to the form, and the next data entry will go to a new form instance.

- **Importing files that contain non-alphanumeric characters**—All import files that include non-alphanumeric characters, such as Japanese characters, must be in Unicode format. All other formats will be read as ASCII and may yield misleading error messages.

- **Importing data to an Add Entry itemset**—You can add rows to an Add Entry itemset, and you can add or modify data in an existing Add Entry itemset row.
  - To add a row to an Add Entry itemset, use the PATIENTDATA tag. You can add one row at a time to the itemset.

    For more information, see *Adding new clinical data* (on page 133).
  - To add or modify data for an item in an Add Entry itemset, use the EDITPATIENTDATA tag.
    - You must use the EDITPATIENTDATA tag to add or modify data for an item in an Add Entry itemset, regardless of whether the item previously contained data.
    - When you use the EDITPATIENTDATA tag, you cannot use the DATA sub-tag with the ITEMSETINDEX tag to modify itemset data. You must use the ITEMSETINDEX attribute with the EDITPATIENTDATA tag.

    For more information, see *Updating existing clinical data* (on page 138).
- **Importing data to a Repeating Data itemset**—You can create the rows in the database for a Repeating Data itemset, and you can add or modify data in an existing Repeating Data itemset row.
  - To create rows in the database for a Repeating Data itemset, use the PATIENTDATA tag. You can create one or more rows at a time for a Repeating Data itemset.

    Note: To create rows in a Repeating Data itemset using the PATIENTDATA tag, you must use the ITEMSETINDEX attribute with the DATA sub-tag.

    For more information, see *Adding new clinical data* (on page 133).
  - To add or modify data for an item in a Repeating Data itemset, use the EDITPATIENTDATA tag.
    - You must use the EDITPATIENTDATA tag to add or modify data for an item in a Repeating Data itemset, regardless of whether the item previously contained data.
    - When you use the EDITPATIENTDATA tag, you cannot use the DATA sub-tag with the ITEMSETINDEX tag to modify itemset data. You must use the ITEMSETINDEX attribute with the EDITPATIENTDATA tag.

    For more information, see *Updating existing clinical data* (on page 138).
  - Do not use the itemset **!match!** field in a map file, which is used to match an itemset row, when using the **!cd!ItemsetIndex** field. The **!match!** field is generally added manually to a Map file, and does not appear on the Field Definition page of the InForm Data Import utility.
- **Deleting and undeleting Add Entry itemsets**—To delete or undelete Add Entry itemset data, you must use the MedML file option.

  Note: You cannot delete or undelete a Repeating Data itemset.

Importing an XML file

Overview of importing an XML file

To import an XML file:

1 Create an XML file that contains the subject data to add or update.

For more information, see *Creating an XML import file* (on page 128).

2 Run the import with the MedML file option.

For more information, see *Running the import with the MedML file option* (on page 147).

Creating an XML import file

The import file for the MedML file option is an XML file that contains elements that specify the type of processing to perform during the import and the destinations and values of the import data. The file can contain tags for the following types of import actions:

- Screening and enrolling a subject.
- Adding new subject data.
- Updating existing subject data.
- Transferring a subject from one site to another.

To create the import file, use any text editor that creates plain text files. For more information, see *Appendix A: Sample data import XML* (on page 203).

Creating an XML submission

1 Create a first line that contains the xml version number. This string must be lower case:

<?xml version="1.0"?>

2 Add an opening and closing element that tells the InForm Data Import utility what type of processing to perform:

```
<CLINICALDATA>
</CLINICALDATA>
```

3 Between the opening and closing elements, add opening and closing elements for each activity for the InForm Data Import utility to perform. Use one set of activity elements for each subject for whom to import data. For example, to import screening data for a new subject, use the following elements:

```
<SCREEN>
</SCREEN>
```

4 In the opening element for the import activity, add the attributes required for that activity type, and any optional attributes. For example, the **SCREEN** element requires a **SITEMNEMONIC** or **SITENAME** attribute to specify the subject site by mnemonic or by name. If the subject site mnemonic is PF, you would insert the **SITEMNEMONIC** attribute as follows:

```
<SCREEN SITEMNEMONIC="PF">
```

Note: If you are using the InForm Data Import utility to transfer subject records between sites, go to the final step. A subject record transfer import file does not use the DATA element.

5 Between the opening and closing elements that specify the import activity, insert a DATA element for each form control:

```
<DATA/>
```

6 The DATA element has the following required attributes:

- **TAG**—A database path that identifies the target data item control, and that is made up of RefNames in the following order:

  Section.Itemset.Item[.control[.control…]]

- **Section**—RefName of the section, as specified in the XML file that contains the section definition.

- **Itemset**—RefName of the itemset, as specified in the XML file that contains the itemset definition. If the target data item is a regular CRF item, not an itemset, type 0.

- **Item**—RefName of the item, as specified in the XML file that contains the item definition. Create a separate DATA element for each item in an itemset.

- **Control**—RefName of the control, as specified in the XML file that contains the control definition. To access an element of a group control, refer to each parent control in which the child element is nested. For example, to address one of two text controls within a group control, type the RefName of the group control followed by the RefName of the text control, and separate the names with periods, as follows:

  GroupControlRefName.TextControlRefName.

- One of the following:
  - **VALUE**—The value of the data to import into the control. Enclose the value in double quotes.

    Note: Because double quotes are used to delimit the value of an attribute, you cannot include double quotes as part of the value text. If you need to include double quotes as part of the value text, use the XML entity reference **"**.
  - **CHILDSELECTED**—The RefName of the selected child control, if the child control is nested within a compound control. For example, use the **CHILDSELECTED** attribute to indicate which radio control to select if the radio control includes two drop-down lists.
  - **MONTH**, **DAY**, **YEAR**, **HOUR**, **MINUTE**, **SECOND**—The value of each applicable part of a datetime control.
  - **UNIT**—The RefName of the selected unit, when a unit definition is part of the target control.
  - **COMMENT**—The text of an item-level comment.
  - **REASONINCOMPLETE**—The reason the item is incomplete. When you specify this attribute, do not include a VALUE or any datetime control attributes in the DATA element.
  - **NOMULTIVALUE**—Indicates that a data value containing a comma (,) is a single value. Without this attribute, only the data preceding a comma is stored as a value. Data that occurs after a comma is assumed to be a separate value in a multi-value DATA element.

  For existing subject data:

  - **CLEARVALUE**—TRUE, to clear the existing value of the control.

7 Save the file.

Examples

The following example shows a DATA element that is used to import the initials of subject AAA in the PF site to the Subject initials field in the screening form:

```
<SCREEN SITEMNEMONIC="PF">
   <DATA TAG="screen.0.patientinitials.patientinitials"
   VALUE="AAA"/>
</SCREEN>
```

The following example shows the use of the CHILDSELECTED attribute of the DATA element to indicate that, in the RACE radio control, the RACEPULLDOWN radio button is being selected. The second DATA element gives the selected value within the drop-down list.

```
<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>
```

The following example shows a DATA element used to specify a date to be imported into an enrollment form:

```
<ENROLL PATIENTINITIALS=”AAA” SITEMNEMONIC=”PF”
PATIENTNUMBER=”BK1” ENROLL=”TRUE”>
   <DATA TAG=”consent.0.consentdate.date”
   MONTH=”1” DAY=”6” YEAR=”1999”/>
</ENROLL>
```

The following example shows the use of the UNIT attribute to specify that the unit in which height is being measured is inches.

```
<PATIENTDATA PATIENTINITIALS=”AAA” SITEMNEMONIC=”PF”
FORMSETREFNAME=”Visit1” FORMREFNAME=”DEM”>
   <DATA TAG=”DEM.0.HEIGHT.HEIGHTTEXT”
   VALUE=”67” UNIT=”Inches”/>
</PATIENTDATA>
```

The following example shows the use of the NOMULTIVALUE attribute to specify that a data value containing a comma (,) is a single value.

```
<PATIENTDATA PATIENTINITIALS=”AAA” SITEMNEMONIC=”PF”
FORMSETREFNAME=”Visit1” FORMREFNAME=”DEM”>
   <DATA TAG=”DEM.0.HEIGHT.ID”
   VALUE=”67,11” NOMULTIVALUE=””/>
</PATIENTDATA>
```

Screening

To import screening data for a subject at a site, use the SCREEN element with the following required attribute:

Attribute	Definition
SITEMNEMONIC	Mnemonic of the site at which the subject is being screened.

Example

The following sample file illustrates the tags used to screen subject AAA at site PF.

```
<?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>
```

Enrolling

To import enrollment data for a subject at a site, use the ENROLL element with the following required attributes:

Attribute	Definition
PATIENTINITIALS	Initials of the subject that is being enrolled.
SITEMNEMONIC	Mnemonic of the site at which the subject is being screened.
DUPLICATEORDER	Number that specifies the order in which subjects who have the same subject initials, and were enrolled in the same site were screened.
PATIENTNUMBER	Subject number of the subject that is being enrolled.
ENROLL	TRUE or FALSE, indicating whether to enroll the subject.

Example 1

The following example illustrates the tags used to enroll subject XYZ at site PF. This example assumes that subject XYZ has previously been screened.

```
<?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>
```

Example 2

The following example illustrates the use of the DUPLICATEORDER attribute to specify the order in which subjects with duplicate subject initials should be enrolled. Subjects John R. Doe and Jane R. Doe (JRD) are screened at site PF; first John, then Jane.

```
<!-- 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>
```

Adding new clinical data

To add new clinical data to an existing subject at a site, use the PATIENTDATA element with the following attributes:

Attribute	Definition
PATIENTINITIALS	Initials of the subject that is being enrolled. Either **PATIENTINITIALS** or **PATIENTNUMBER** is required.
PATIENTNUMBER	Subject number of the subject that is being enrolled. Either **PATIENTINITIALS** or **PATIENTNUMBER** is required.
SITEMNEMONIC	Mnemonic of the site at which the subject is being screened. Either **SITEMNEMONIC** or **SITENAME** is required.
SITENAME	Name of the site at which the subject is enrolled. Either **SITEMNEMONIC** or **SITENAME** is required.
FORMSETREFNAME	RefName of the visit to which you are importing data.
FORMSETINDEX	Indicates to which visit instance to add the data.
FORMREFNAME	Specifies the RefName of the CRF to which you are importing data.
REASONINCOMPLETE	This attribute may apply to either the form or item level. The value is one of the values in the radio group control. This attribute will be ignored if the form or item is complete. To add an incompletion reason at the form level, do not include a DATA element in the PATIENTDATA group.
FORMINDEX	(Optional) If not present, then a new repeating form instance will be created (if **FORMREFNAME** is a repeating form). If present, the value indicates to which form instance the new data will be added.
ASSOCIATION	Indicates that an association exists.

Attribute	Definition
ITEMSETINDEX	Row number for the row to add to an itemset. • For an Add Entry itemset: ▪ Specify one row number at a time. ▪ Adds a single row to an Add Entry itemset. ▪ You can use the ITEMSETINDEX attribute with the PATIENTDATA tag or the DATA sub-tag. • For a Repeating Data itemset: ▪ Specify one or more rows at a time. ▪ Creates all rows for a Repeating Data itemset. ▪ For a Repeating Data itemset, you must use the ITEMSETINDEX attribute with the DATA sub-tag. **Notes:** You cannot specify the ITEMSETINDEX attribute on both the PATIENTDATA and DATA tags. ITEMSETINDEX is an optional attribute of the PATIENTDATA tag and the DATA sub-tag.

Adding a row to an Add Entry itemset

To add a row to an Add Entry itemset on a form, use the following additional attributes:

Attribute	Definition
SECTIONNAME	Specifies the RefName of the section in which the itemset occurs.
ITEMSETNAME	Specifies the RefName of the itemset definition.
ITEMSETINDEX	Number of the Add Entry itemset row to update. If the ITEMSETINDEX attribute is blank or has a value of 0, the InForm Data Import utility adds a new itemset row. If ITEMSETINDEX is a number other than 0, the InForm Data Import utility updates the specified row. Note that existing itemset data is not changed, but missing data is filled in. **Note:** To add or modify data in a row in an Add Entry or a Repeating Data itemset, you must use the EDITPATIENTDATA tag. This is true regardless of whether the row contains data.

Creating rows for a Repeating Data itemset

To create rows for a Repeating Data itemset on a form, use the following attributes:

Attribute	Definition
PATIENTDATA attribute	Definition
SECTIONNAME	Specifies the RefName of the section in which the itemset occurs.
ITEMSETNAME	Specifies the RefName of the itemset definition.
DATA attribute	Definition
ITEMSETINDEX	Specifies one or more itemset row numbers for the rows to create for the Repeating Data itemset. Valid values are 1-*n*, where n=the value of the INITIALROWCOUNT of the Repeating Data itemset. **Note:** To add or modify data in a row in an Add Entry or a Repeating Data itemset, you must use the EDITPATIENTDATA tag. This is true regardless of whether the row contains data.

Adding data to an unscheduled visit

To add data to an unscheduled visit, use the following additional attributes:

Attribute	Definition
NEWUNSCHEDVISIT	Indicates whether the data is being added to a new unscheduled visit. Values are TRUE or FALSE (the default). Use this attribute on the Visit Date form, a predefined form with the FORMREFNAME of DOV. For more information, see *Example 3: Adding data to an unscheduled visit* (on page 137).
FORMSETINDEX	Number that specifies the unscheduled visit to which to add the data. The number corresponds to the order in which the unscheduled visit was added to the study.

Adding a comment

You can add a comment to an item, an item within an itemset, or a form by using the **COMMENT** attribute:

- To add a comment to an item, include the **COMMENT** attribute along with the text of the comment in the **DATA** attribute.
- To add a comment to a form, include the **COMMENT** attribute along with the text of the comment in a **PATIENTDATA** attribute that does not include **SECTIONNAME** and **ITEMSETNAME** attributes.

Example: Adding data to a form

The following XML fragment shows elements used to add data to the DEM form for subject AAA at site PF.

```
<?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>
```

Example: Adding a row to an Add Entry itemset

The following XML fragment shows the elements used to create a new row in an Add Entry itemset on the Hypertension History form.

```
<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>
```

The following XML fragment shows the elements used to finish entering items for an existing Add Entry itemset on the Hypertension History form.

```
<PATIENTDATA
   PATIENTINITIALS="AAA" SITEMNEMONIC="PF"
   FORMSETREFNAME="Visit1" FORMREFNAME="HH"
   SECTIONNAME="PT" ITEMSETNAME="PT" ITEMSETINDEX="1">
   <DATA TAG="PT.PT.DOSEDATE.DOSEDATE" MONTH="12" DAY="23"
      YEAR="1998"/>
   <DATA TAG="PT.PT.DISCPULLDOWN.DISCPULLDOWN"
      VALUE="Not Effective"/>
</PATIENTDATA>
```

Example: Creating rows for a Repeating Data itemset

The following XML fragment shows the elements used to create the rows in the Sample Collection itemset on the Dose VS2 form.

The Repeating Data itemset contains two rows with predefined values:

- Hour 10, Minute 0
- Hour 10, Minute 2

The value of 50.000 is entered for the Hour 10, Minute 0 data point.

```
<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>
```

Example: Adding data to an unscheduled visit

The following XML fragment shows the elements used to add data to the DOV form and Vital Signs form in the first and second unscheduled visits containing those forms.

```
<?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>
```

Example: Specifying REASONINCOMPLETE

The following XML fragment shows an addition to the Pulse Rhythm item on the Vital Signs (VS) form.

```
<PATIENTDATA
   PATIENTINITIALS="A3" SITEMNEMONIC="PF"
   FORMSETREFNAME="DV1" FORMREFNAME="VS"
   REASONOTHER="test reason">
   <DATA TAG="VS.0.PULSERHYTHM.PULSERHYTHMRADIO"
      COMMENT=irregular REASONINCOMPLETE="NAElement"/>
</PATIENTDATA>
```

Example: Creating a new association instance

The following XML fragment shows the creation of a new association instance.

```
<PATIENTDATA
   PATIENTINITIALS="pjb" SITEMNEMONIC="PF"
   FORMSETREFNAME="Visit6" FORMREFNAME="VS"
   FORMINDEX=”2”
   REASONPULLDOWN=”3”
   <ASSOCIATION FORMSETREFNAME=”Visit1” FORMREFNAME=”DEM”
      FORMINDEX=”4”/>
</PATIENTDATA>
```

Updating existing clinical data

To modify existing clinical data for a subject at a site, use the EDITPATIENTDATA element with the following attributes:

Attribute	Definition
PATIENTINITIALS	Initials of the subject that is being enrolled. Either **PATIENTINITIALS** or **PATIENTNUMBER** is required.
PATIENTNUMBER	Subject number of the subject that is being enrolled. Either **PATIENTINITIALS** or **PATIENTNUMBER** is required.
SITEMNEMONIC	Mnemonic of the site at which the subject is being screened. Either **SITEMNEMONIC** or **SITENAME** is required.
SITENAME	Name of the site at which the subject is enrolled. Either **SITEMNEMONIC** or **SITENAME** is required.
FORMSETREFNAME	RefName of the visit to which you are importing data.

Attribute	Definition
FORMREFNAME	Specifies the RefName of the CRF to which you are importing data.
REASONPULLDOWN	Specifies the value of a predefined reason for change, as listed in the Reason for Change drop-down list on the Data Value(s) form.
REASONOTHER	Specifies the text of a reason for change, as entered in the Other Reason for Change field on the Data Value(s) form.
REASONINCOMPLETE	This attribute may apply to either the form or item level. The value is one of the values in the radio group control. This attribute will be ignored if the form or item is complete.
CLEARCRF	When true, indicates that all following <Data Tags> are ignored.
FORMINDEX	Required for repeating forms. Indicates a repeating form to update. Value corresponds to the form instance to update.
ASSOCIATION	Indicates that an association exists.
ACTION	ADD or REMOVE; specifies whether to add or remove an association.
ITEMSETINDEX	Row number for the item in an Add Entry or Repeating Data itemset to modify or add data to. **Note:** You cannot use the ITEMSETINDEX with the DATA sub-tag of the EDITPATIENTDATA tag.

Editing data in an Add Entry itemset

To edit data in an Add Entry itemset on a form, use the following additional attributes:

Attribute	Definition
SECTIONNAME	Specifies the RefName of the section in which the itemset occurs.
ITEMSETNAME	Specifies the RefName of the itemset definition.
ITEMSETINDEX	Number of the Add Entry itemset row to update. If the ITEMSETINDEX attribute is blank or has a value of 0, the InForm Data Import utility adds a new itemset row. If ITEMSETINDEX is a number other than 0, the InForm Data Import utility updates the specified row. Note that existing itemset data is not changed, but missing data is filled in. **Note:** To add or modify data in a row in an Add Entry or a Repeating Data itemset, you must use the EDITPATIENTDATA tag. This is true regardless of whether the row contains data.

Editing data in a Repeating Data itemset

To edit data in a Repeating Data itemset on a form, use the following additional attributes:

Attribute	Definition
SECTIONNAME	Specifies the RefName of the section in which the itemset occurs.
ITEMSETNAME	Specifies the RefName of the itemset definition.
ITEMSETINDEX	Adds or modifies data for an item in a Repeating Data itemset. Specifies the row number for the item in a Repeating Data itemset to modify or add data to. Valid values are 1-*n*, where n=the value of the INITIALROWCOUNT of the Repeating Data itemset. **Note:** To add or modify data in a row in an Add Entry or a Repeating Data itemset, you must use the EDITPATIENTDATA tag. This is true regardless of whether the row contains data.

Editing a comment

You can edit a comment on an item, an item within an itemset, or a form by using the **COMMENT** attribute:

- To edit a comment on an item, include the **COMMENT** attribute along with the text of the comment in the **DATA** element.
- To edit a comment on an itemset, include the **COMMENT** attribute along with the text of the comment in an **EDITPATIENTDATA** element that includes the **SECTIONNAME** and **ITEMSETNAME** attributes, which signal that the **EDITPATIENTDATA** element is updating an itemset.
- To edit a comment on a form, include the **COMMENT** attribute along with the text of the comment in an **EDITPATIENTDATA** element that does not include **SECTIONNAME** and **ITEMSETNAME** attributes.

Example: Changing a data value

The following sample file illustrates a change to the value of the Height item on the DEM form. The Reason for Change is a string other than the predefined Reason for Change strings on the Data Value(s) form.

Note: You must enter the UNIT value even if you are not modifying it, or it will be removed.

```
<?xml version="1.0"?>
<CLINICALDATA>
<!-- Demographics form -->
<EDITPATIENTDATA
   PATIENTINITIALS="XYZ"
   SITEMNEMONIC="PF"
   FORMSETREFNAME="Visit1"
   FORMREFNAME="DEM"
   REASONOTHER="received new data">
   <DATA TAG="DEM.0.HEIGHT.PFHT_TC" VALUE="75"
      UNIT="Inches" />
</EDITPATIENTDATA>
</CLINICALDATA>
```

Example: Clearing a data value

The following XML file fragment shows the use of the **CLEARVALUE** attribute of the **DATA** element. When you change the selection of a compound radio control, you must clear the value of the original child control.

```
<EDITPATIENTDATA
   PATIENTINITIALS="PE1" SITEMNEMONIC="PF"
   FORMSETREFNAME="Visit1" FORMREFNAME="DEM"
   REASONPULLDOWN="New Information">
   <DATA TAG="RACEGROUP.RACETEXT" CLEARVALUE="TRUE"/>
</EDITPATIENTDATA>
```

Example: Modifying an Add Entry itemset row

The following XML file fragment modifies an Add Entry itemset on the Adverse Events (AE) form.

```
<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>
```

Example: Modifying a Repeating Data itemset

The following XML file fragment modifies a Repeating Data itemset on the Dose VS2 form.

```
<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>
```

Example: Specifying REASONINCOMPLETE

The following XML file fragment shows a modification to the Pulse Rhythm item on the Vital Signs (VS) form.

```
<EDITPATIENTDATA
   PATIENTINITIALS="A3" SITEMNEMONIC="PF"
   FORMSETREFNAME="DV1" FORMREFNAME="VS"
   REASONOTHER="test reason">
   <DATA TAG="VS.0.PULSERYTHM.PULSERHYTHMRADIO"
      COMMENT=me too REASONINCOMPLETE="NAElement"/>
</EDITPATIENTDATA>
```

Example: Clearing a CRF

The following XML file fragment shows the clearing of a CFR.

```
<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>
```

Example: Deleting an association

The following XML file fragment shows a command to delete an association.

```
<EDITPATIENTDATA
   PATIENTINITIALS="DD" SITEMNEMONIC="PF"
   FORMSETREFNAME="VISIT7" FORMREFNAME="PREIM"
   FORMINDEX="2"
   REASONPULLDOWN="Transcription Error">
   <ASSOCIATION FORMSETREFNAME="Visit6" FORMREFNAME="PE" FORMINDEX="3"
   ACTION="REMOVE"
</EDITPATIENTDATA>
```

Example: Deleting a repeating form

The following XML file fragment shows a command to delete a repeating form.

```
<?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>
```

Example: Restoring a repeating form

The following XML file fragment shows a command to restore (undelete) a repeating form.

```
<?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>
```

Transferring a subject record

You can use the InForm user interface to transfer subject information one subject at a time. You can also transfer subjects in bulk using the InForm Data Import utility.

You can transfer subjects who:

- Change permanent address before completing the study.
- Have multiple residences throughout the course of the study.
- Were initially assigned to the wrong sites or to an investigator who is no longer with the study.

Whether you transfer subjects individually, or transfer them in bulk, keep in mind that:

- The InForm application only allows you to transfer subjects from one site to another if the study version at the destination site is the same or greater than the study version at the current subject site.

  You can view the study version for a site in the Admin interface in the InForm user interface.

  You can transfer only subjects who are fully enrolled; you cannot transfer a subject who is screened but not enrolled or a subject who has failed enrollment.

To transfer a subject record from one site to another, use the following elements:

- PATIENTSITECHANGE
- NEWSITE
- CURRENTSITE

PATIENTSITECHANGE

The **PATIENTSITECHANGE** element identifies each subject to transfer. Use one pair of opening and closing **PATIENTSITECHANGE** elements for each subject to transfer. The **PATIENTSITECHANGE** element surrounds all information about a single subject, the current site, and the destination site.

The **PATIENTSITECHANGE** element has one required attribute:

Attribute	Definition
REASON	A textual description of the reason for the subject transfer.

Within the **PATIENTSITECHANGE** element, include these elements:

- NEWSITE
- CURRENTSITE

NEWSITE

The **NEWSITE** element provides information about the destination site for the subject. Use the **NEWSITE** element within the opening and closing **PATIENTSITECHANGE** elements.

The **NEWSITE** element requires one attribute to identify the destination site. You can use any one of these attributes:

Attribute	Definition
SITEMNEMONIC	Mnemonic for the destination site.
SITENAME	Site name for the destination site.

The **NEWSITE** element has one optional attribute:

Attribute	Definition
PATIENTNUMBER	The subject number for the subject at the destination site. Use this attribute only if you need to change the subject number from the one that exists at the current site because a duplicate exists at the destination site.

CURRENTSITE

The **CURRENTSITE** element provides information about the current, or originating, site for the subject. Use the **CURRENTSITE** element within the opening and closing **PATIENTSITECHANGE** elements.

The **CURRENTSITE** element requires two attributes: one to identify the current site, and one to identify the subject who is to be transferred.

Use any one of the following attributes to identify the current site:

Attribute	Definition
SITEMNEMONIC	Mnemonic for the destination site.
SITENAME	Site name for the destination site.

Use any one of the following attributes to identify the subject to be transferred:

Attribute	Definition
PATIENTINITIALS	The initials of the subject being transferred. If you use this attribute, and you know that more than one subject at the current site has the specified initials, use the optional **DUPLICATEORDER** attribute to indicate which subject to transfer.
PATIENTNUMBER	The subject number of the subject being transferred, as it exists on the current site.

Use the following attribute with the **CURRENTSITE** element if more than one subject at the site has the initials specified in the **PATIENTINITIALS** element:

Attribute	Definition
DUPLICATEORDER	Specifies which subject should be transferred, if more than one subject at the site has the same initials. When multiple subjects have the same initials, the InForm application checks the screening numbers of those subjects and transfers the subject whose screening number is in the order specified by the **DUPLICATEORDER** tag. Use this attribute only if you have used **PATIENTINITIALS** to identify the subject.

Example

This example shows the elements in a MedML file that is used to transfer several subjects. Note that each pair of **PATIENTSITECHANGE** attributes defines current and destination site information for one subject.

```
<?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>
```

Additional InForm Data Import utility attributes

Attribute	Description	Values
CLEARCRF	Indicates whether to clear the specified form.	• TRUE • FALSE
COMMONFORM	Indicates whether a form is a common CRF.	• TRUE • FALSE
DELETEITEMSET	Indicates whether to delete an Add Entry itemset.	• TRUE • FALSE
FORMSTATUS	Modifies the status of a form.	• FREEZE • UNFREEZE • LOCK • UNLOCK • DELETE • UNDELETE
OVERRIDE	Specifies whether a subject screening or enrollment was overridden.	• TRUE • FALSE
UNDELETEITEMSET	Indicates whether to undelete a deleted Add Entry itemset.	• TRUE • FALSE

Running the import with the MedML file option

Note: To import data into the database, the InForm server must be running before you start the InForm Data Import utility.

To import an XML file into the database and submit it through the InForm application server:

1 Click **Start** > **All Programs** > **Oracle® Health Sciences** > **InForm 6.1** > **InForm Data Import**.

The InForm Data Import main window appears.

2 Select **MedML file**.

3 In the **Trial Name** field, enter the name of the study into which to import the file. The last 10 studies you accessed appear in the drop-down list.

4 Click **Next**.

5 In the **Select MedML file** field, type the full path name of the XML file to import, or click **Browse** and locate the file. Any of the following options are valid:

- The name of one XML file.
- The name of multiple XML files, each separated by a space.
- The name of a response file that contains multiple XML files, one per row, in the following format:

  @filename

6 Optionally, to run the import without actually importing data into the database, select **Parse Only**. Use this function to test the syntax of your MedML file before you import it into the database.

7 Click **Next**.

A dialog box appears and requests your InForm name and password.

Note: If you selected **Parse Only**, you do not need to specify a name and password.

8 In the **Name** field, type the name of an InForm user who has the appropriate rights for the data you are importing:

To import data that matches this InForm system activity	User needs these rights
Add subject clinical data	Enter Data into a CRF
Update subject clinical data	Edit Data on a CRF

9 In the **Password** field, type the user password, and click **Next**.

The Summary window appears.

10 Optionally, select any of the following:

- **Stop on Error**—To instruct the InForm Data Import utility to stop if it encounters an error.
- **Verbose**—To instruct the InForm Data Import utility to generate detailed messages as it processes the file.
- **Use output file**—Specify the filename to save the output file as a text file.

11 Click **Start**.

The InForm Data Import utility processes the import file, writes messages to the message area and the output file, if specified, and adds or updates data in the database.

12 Close the InForm Data Import utility.

Importing a data and map file

Specifying a data and map file

To import a data and map file:

1 Do one of the following:

- Create a map file to import.

  For more information, see *Creating a data and map file* (on page 149).

- Edit an existing map file to import.

  For more information, see *Editing an existing data and map file* (on page 152).

2 Run the import with the InForm Date and Map files option.

  For more information, see *Running the import with the InForm Data and Map files option* (on page 161).

Creating a data and map file

The data and map file that are used in the direct import method must be a text file. You can use any text editor that creates plain text files, or develop a conversion utility that automatically formats your raw data.

You can direct the data in the file to target controls on more than one CRF; however, you must use separate files for each itemset into which you are importing rows of data, and data that you import into CRF itemsets must be in a separate file from data that you import into regular CRF items.

The import file must have the following characteristics.

- Each import row must include either of the following:
  - A field that specifies the subject number and initials in the following format:

    subject_number (subject_initials)

    If it is possible that more than one subject could have the same subject number and initials, you must also include a field for the site mnemonic of the subject.
  - The database ID of the subject.
- All import fields must be separated by a pipe character (|).
- Do not include double quotation marks (") in the data file.

Additionally, some types of data must be presented in specific ways. The following sections describe how to set up the following fields and controls for import:

- *Date fields* (on page 150).
- *Time fields* (on page 150).
- *DateTime fields* (on page 150).
- *Nested Controls* (on page 151).
- *Checkboxes and multiple-selection drop-down lists* (on page 151).
- *Units* (on page 151).
- *Item comments* (on page 151).

Date fields

Dates must consist of three fields in month|day|year format, using a 4-digit year. Observe these formatting considerations:

- If a date is missing a component, include a null field, as in the following example:

```
month||year
```

- If a component of a date is unknown, use the keyword UNK, as in the following example:

```
month|UNK|year
```

Time fields

Times must consist of three fields in hour|minute|second format, using a 24-hour clock. Observe these formatting considerations:

- If a time is missing a component, include a null field, as in the following example:

```
hour|minute|
```

- If a component of a time is unknown, use the keyword UNK, as in the following example:

```
hour|UNK|UNK
```

DateTime fields

Dates must consist of three fields: month, day, and year. The year must use four digits, and times must use a 24-hour clock. Observe these formatting considerations:

- By default, the date order format must match the order specified in the study configuration. If data is entered in a date order format that is different from the study configuration setting, you can override the setting by using the -datetimeformat command line parameter during import. For more information, see *Running the InForm Data Import utility from the command line* (on page 165).
- If a datetime data item is missing a component, include a null field.
- If a component of a datetime item is unknown, use the keyword UNK.

Example:

```
PF|001 (CJB)|Sep|23|1975|1|1|215210|LABDATA|Clinical Chemistry:|AG
Ratio|XGR|Nov|UNK|1998|0.8-2||2.0|N|||
PF|001 (CJB)|Sep|23|1975|1|1|215210|LABDATA|Clinical
Chemistry:|ALAT(SGPT)|XGP|Nov|UNK|1998|0-48|U/L|42|N|||
PF|001 (CJB)|Sep|23|1975|1|1|215210|LABDATA|Clinical
Chemistry:|Albumin|XAL|Nov|UNK|1998|3.2-5|G/DL|4.3|N|||
PF|001 (CJB)|Sep|23|1975|1|1|215210|LABDATA|Clinical Chemistry:|Alkaline
Phosphatase|XLK|Nov|UNK|1998|20-125|U/L|63|N|||
```

Nested Controls

Nested controls must consist of a field for each control separated by a pipe character (|). Only one of these fields will contain information for each row of data.

Example:

```
PF|001 (CJB)|Sep|23|1975|1|1|215210|LABDATA|hematology|||
PF|001 (CJB)|Sep|23|1975|1|1|215210|LABDATA||urinalysis||
PF|001 (CJB)|Sep|23|1975|1|1|215210|LABDATA|||toxicology|
```

Checkboxes and multiple-selection drop-down lists

Checkboxes and multiple-selection drop-down lists must consist of a field with a list of zero or more controls separated by a comma.

Example:

```
PF|001 (CJB)|Sep|23|1975|LABDATA|Sinus Tachycardia, Premature Ectopic
Junctional Beats, Sinus Bradycardia, Premature Ectopic Atrial Beats|
```

Units

A field that contains units must immediately follow the value that it describes.

The example shows weight in pounds. The number value of the weight (177) is contained in one field, and the unit value (pounds) is contained in the following field.

```
PF|001 (CJB)|Sep|23|1975|70|in|177|lb|
```

Item comments

Item comments must consist of a field that contains the text comment for the item.

Example:

```
PF|001 (CJB)|Sep|23|1975|70|177|weight measured with shoes and socks
```

Editing an existing data and map file

To edit an existing import map file:

Note: If you change definitions in a map file and click **Cancel** (instead of **Finish**), an empty map file is saved with the path and filename you specified in the **Map** field.

1 Click **Start** > **All Programs** > **Oracle® Health Sciences** > **InForm 6.1** > **InForm Data Import**.

The InForm Data Import main window appears.

2 Select **InForm Data and Map files**.

3 In the **Trial Name** field, enter the name of the study into which to import the file. The last 10 studies you accessed appear in the drop-down list.

4 Click **Next**.

A dialog box appears where you specify the data and map files to import.

5 In the **Data** field, type the full path name of the data file you want to import, or click **Browse**.

6 In the **Map** field, type the full path name of the map file you want to import, or click **Browse**.

Note: After you enter a file name and open the map file editor, the file is created, regardless of whether you click Start or Stop. This file is stored in the same location as the pfimport.exe file.

7 Click **Edit Map File**.

The Field Definition dialog box appears.

8 To edit the map file:

a Specify whether the data is targeted for CRF items or CRF itemsets. For more information, see *Specifying a submission type* (on page 153).

b Specify the input field type. For more information, see *Specifying an input field type* (on page 153).

c Specify the definition of each map file, one field at a time. For more information, see *Building an item path* (on page 155).

d Specify the data type. For more information, see *Specifying a data type* (on page 157).

e Specify mappings between the values in your import file and the values defined for the target data fields in the InForm database. For more information, see *Mapping strings and child controls* (on page 157).

f Indicate whether the data contains multiple selection items. For more information, see *Indicating data contains multiple selection items* (on page 158).

g Check for duplicate information within itemsets. For more information, see *Checking for duplicate information within itemsets* (on page 159).

h Import information to an unscheduled visit. For more information, see *Importing information into an unscheduled visit* (on page 159).

i Insert or delete an import field into the import file. For more information, see *Inserting or deleting an import field* (on page 160).

j Check the map against the import file. For more information, see *Checking the map against the import file* (on page 160).

9 After you have saved the map file, exit the map file editor to return to the InForm Data Import utility window to import the data and map file. For more information, see *Running the import using the data and map import file* (on page 161).

Specifying a submission type

1 Select one of the following options:

- **Form**—If all of the data in the import file is targeted for regular CRF items.
- **Itemset**—If all of the data in the import file is targeted for CRF itemsets. Optionally, to specify that you want to import only data that updates existing itemsets, select **Disallow New Itemset Rows**.

2 If you want to import data that only updates existing unscheduled visits, select **Disallow New Repeating Visit Instances**.

3 In the **Reason String** field, type the text that should appear in the **Reason for Change** section on the Data Value(s) screen if data is updated in the data load. The default reason is Lab Import.

4 Click **Continue**.

The Field Definition dialog box appears.

5 To define map fields, follow the procedure in *Specifying an input field type* (on page 153).

Specifying an input field type

Use the Field Definition dialog box to define each map field one at a time. Each map field corresponds to a data item in the import file.

In the Field Definition dialog box, select one of the following field types:

- **InForm Item Path**—Contains data that is targeted to a data item on a CRF. When you select this option, you must provide additional information about the import field. For more information, see *Building an item path* (on page 155).
- **Patient Field – Number (Initials)**—Contains subject identification information in either patient_number (patient_initials) format or in the form of a subject database ID. If your import data identifies each subject by the subject database ID, select the **Field Contains Known Patient ID** checkbox.

  When you select this option, the definition of the map field is complete, and you can move to another map field definition or save the map file and exit the map file editor. For more information, see *Navigating the map file* (on page 160).
- **Site Mnemonic Field**—Contains the mnemonic of the site where the subject is enrolled. When you select this option, the definition of the map field is complete, and you can move to another map field definition or save the map file and exit the map file editor. For more information, see *Navigating the map file* (on page 160).

- **Itemset Index Field**—Contains the itemset index, a 1 based row number. Used to identify an existing row of an Add Entry or Repeating Data itemset. When you select this option, the definition of the map field is complete, and you can move to another map field definition or save the map file and exit the map file editor. For more information, see *Navigating the map file* (on page 160).
- **Ignore This Field—**Indicates that you do not want the field to be imported. When you select this option, the definition of the map field is complete, and you can move to another map field definition or save the map file and exit the map file editor. For more information, see *Navigating the map file* (on page 160).
- **Unit Symbol for previous field**—Contains the symbol for the units that apply to the previous field in the import and map files. The symbol of a unit is the text that identifies the unit on the CRF, as defined in the SYMBOL attribute of the UNIT definition in the appropriate XML file. When you select this option, the definition of the map field is complete, and you can move to another map field definition or save the map file and exit the map file editor. For more information, see *Navigating the map file* (on page 160).
- **Comment Field for previous field**—Contains the comment text that is associated with a form or itemset. When you select this option, the definition of the map field is complete, and you can move to another map field definition or save the map file and exit the map file editor. For more information, see *Navigating the map file* (on page 160).

Optionally, select one or more of the following checkboxes:

- **Comma separates multiple-values**—Indicates that the import file contains data which will populate multiple-selection controls such as radio button groups, checkbox groups, and drop-down lists.
- **Match Itemset instance with this field**—Indicates that you want the InForm Data Import utility to check for duplicate data in the import file before importing to the InForm application.
- **Match Repeating visit instance with this field**—Indicates that you want the InForm Data Import utility to check for duplicate dates of visit for the current field in the import file and a date of visit for a repeating visit already in the InForm database.

Building an item path

If you selected **InForm Item Path** as the item type, you must specify the RefName path for the target CRF data item for the field in the import file.

To build an item path, do one of the following:

- Type the path in the **InForm Item Path** text box.

  For more information, see *Entering an item path explicitly* (on page 155).

- Select each RefName from the drop-down lists in the Build Path From Database dialog box.

  For more information, see *Using the Build Path from Database dialog box* (on page 156).

Entering an item path explicitly

To specify an item path, use the following item path:

```
0.Visit.Form.Section.Itemset.Item[.control[.control…]]
```

Each component of the item path is the RefName used to define an element:

- **0**—Indicates the current subject.
- **Visit**—RefName of the visit, as specified in the XML file that contains the visit definition.
- **Form**—RefName of the CRF or other form, as specified in the XML file that contains the form definition.
- **Section**—RefName of the section, as specified in the XML file that contains the section definition.
- **Itemset**—RefName of the itemset, as specified in the XML file that contains the itemset definition. If the import data for which you are creating a map field definition is a regular CRF item, not an itemset, type 0.
- **Item**—RefName of the item, as specified in the XML file that contains the item definition. Create a separate map field for each item in an itemset.
- **Control**—RefName of the control, as specified in the XML file that contains the control definition. To access an element of a group control, refer to each parent control in which the child element is nested. For example, to address one of two text controls within a group control, type the RefName of the group control followed by the RefName of the text control, and separate the names with periods, as follows: GroupControlRefName.TextControlRefName.

Examples

- The item path for the TEMPTEXT field in the TEMPTEXT item in the VS section of the VSL form in a visit called VISIT1 appears as follows:

```
0.Visit1.VSL.VS.0.TEMPTEXT.TEMPTEXT
```

- The item path for an item in an itemset appears as follows:

```
0.Visit1.SS.SECTION2.SS2GROUP.ONSETDATE.ONSETDATE
```

  The control is a date control called ONSETDATE, in the ONSETDATE item of the SS2GROUP itemset in the SECTION2 section of the SS form in VISIT1.

- The item path for a text control nested within the VIEW radio control item in the CHESTXRAY section of the ECG form in VISIT1 appears as follows:

```
0.Visit1.ECG.CHESTXRAY.0.VIEW.VIEWRADIO.OTHERTEXT
```

  The item is identified by the VIEW RefName; the radio control is identified by VIEWRADIO, and the text control is identified by OTHERTEXT.

Using the Build Path From Database dialog box

1 In the Field Definition dialog box, select **InForm Build Path**.

2 Click **Build Path**.

The Build Path From Database dialog appears.

3 From the **Visit** drop-down list, select the RefName of the target visit.

4 From the **Form** drop-down list, select the RefName of the target CRF.

5 From the **Section** drop-down list, select the RefName of the target section.

6 Optionally, from the **Itemset** drop-down list, select the itemset RefName to load the import data into an itemset.

7 From the **Item** drop-down list, select the RefName of the target item.

> **Note:** You must create a separate map field definition for each item in an itemset.

8 From the **Control** and **Child Control** drop-down lists, select the RefName of the target group and child controls.

9 Click **OK**.

Specifying a data type

You must specify a data type for each import field definition that you create as an InForm item path field.

To specify the data type of an import field:

- Click the appropriate button in the **Data Type** section.

  The InForm Data Import utility issues an error if any of the following exist:

  - An invalid integer field. An invalid integer field cannot be fully converted to an integer value. For example, 123$ is an invalid integer field.
  - A string identified as a floating number that does not meet the specification for the CRF control for allowed number of digits before and after the decimal point.
  - A string identified as a text control that is not within the defined size range for the CRF text control.

Mapping strings and child controls

Use the InForm Data Import utility to map field values in your import file to the database definitions of CRF controls, which have predefined values. Additionally, the mapping feature generates mappings in compound controls between individual child controls and their database ID paths.

Mapping strings

When the target of an input data field is a control for which an online user selects a predefined value, the value of the import field must be the same as the value of the selected control as defined in the database. These values are case-sensitive.

If your import file does not match the defined database values, you can convert your file to match them.

Alternatively, you can use the string mapping feature of the InForm Data Import utility to specify mappings between the values in your import file and the values defined for the target data fields in the InForm database.

To use this feature while creating the definition of a map file field:

1 In the Field Definition dialog box, click **Map Strings**.

  The String Map dialog box appears.

2 In the top field, type a possible value of the control as it appears in the import file.

3 In the next field, type the value of the control as it is defined in the database. This definition is specified, generally with a VALUE attribute, in the XML file that is used to load form and data item definitions into the database.

4 Click **Map To**.

  The InForm Data Import utility transfers the pair of values to the **Currently Mapped Strings** field. For example, if you typed *Yes* as a value that appears in your file and *Y* as the defined control value, the **Currently Mapped Strings** field shows the mapping as *Yes maps to Y*.

5 Repeat the mapping definition for each possible combination of values that the field can have in your import file and in the database definition of the control.

6 Click **Update Map**.

Mapping child controls

When the target control is nested within another control (for example, a field within a list of radio buttons), you must create separate map fields for the group control and for each child control within the group. Similarly, your import file must contain fields for the group control and for each possible child control selection.

To assign a specific value to the group control selection, the InForm Data Import utility maps child control names to their database ID paths.

To generate child control mappings for a group control map field:

1 In the Field Definition dialog box, click **Map Strings**.

The String Map dialog box appears.

2 Click **Generate Child Control Mappings**.

The **Currently Mapped Strings** field shows the mappings between child control RefNames and their database IDs. In the import file field that corresponds to the map field that defines the group control, type the child control RefName for which you are providing data.

3 Click **Update Map**.

Indicating that data contains multiple selection items

To import data to a multiple-selection control, a checkbox group, or a multiple-selection drop-down list:

1 In the import file, include all applicable selections in a single field. Separate each selection with a pipe (|).

For example, if you want the cigarettes and cigars checkboxes to be selected in a list containing cigarettes, cigars, and Not Done, and the values defined for those selections are "cigarettes," "cigars," and "ND," the import file should contain a field with the following value:

```
|cigarettes,cigars|
```

2 Define a map field as an InForm Item Path field that references the parent control for the checkboxes or the drop-down list.

3 In the Field Definition dialog box, select **Comma separates multiple-values**.

Checking for duplicate information within itemsets

The InForm Data Import utility can determine whether data already exists in the database by comparing the data itemset in which you are mapping to existing itemsets in the database.

For example, if you are importing lab information and the subject name, the data, and the type of test match data are already in the database, this might indicate that the data is a duplicate. The InForm Data Import utility recognizes this as duplicate data and does not add a second instance of the data in the database.

To use this feature:

1 In the Submission Type window, select **Itemset**.

2 Click **Continue**.

The Field Definition dialog box appears.

3 Click **Next** to navigate to the data object.

4 Select **InForm Item Path** for the data object to match to an itemset in the database.

5 Select **Match Itemset instance with this field**.

6 Repeat these steps for any other data items you want to compare.

Importing information into an unscheduled visit

1 In the Submission Type window, select **Itemset**.

2 Click **Continue**.

The Field Definition window appears.

3 Select the **Item Path** for the date of visit to match to a repeating visit in the database.

4 Select **Match Repeating visit instance with this field**.

5 Repeat these steps for any other unscheduled visits to import.

Inserting or deleting an import field

1 In the Field Definition dialog box, click **Insert Field**.

The InForm Data Import utility clears the data entry fields on the Field Definition dialog.

2 Type the definition for the new field.

3 Click **Next**, **Back**, or **Finish**, as appropriate.

The InForm Data Import utility inserts the new field definition immediately before the field that was displayed when you clicked **Insert Field**.

4 To save the map definition, click **Finish**.

Note: You cannot navigate away from the field until you create or delete the new field definition by clicking Finish or Delete Field.

To delete an import field definition:

1 Click **Next** or **Back** to find the field definition to delete.

2 Click **Delete Field**.

3 To save the map definition, click **Finish**.

Note: If you change definitions in a map file and click **Cancel** (instead of **Finish**), an empty map file is saved with the path and filename you specified in the **Map** field.

Checking the map against the import file

Use the InForm Data Import utility map file editor to review the map field definitions against the actual import data that you will be processing.

To check the map file against the first line of the import file:

- In the Field Definition dialog, navigate through the field definitions and compare the data that appears in the **Sample Data** field with the data for the import file.

Navigating the map file

As you create the fields in a map file, the field definitions are strung together in a sequence in which you can move back and forth.

Use the control buttons at the bottom of the Field Definition dialog box to do the following:

- To view a field that occurs earlier in the map file, click **Back**.
- To view a field that occurs later in the map file, click **Next**.
- To create a new field definition, advance to the last field in the map file and click **Create Next**.

Note: You cannot navigate away from the field until you create or delete the new field definition by clicking **Finish** or **Delete**.

Running the import with the InForm Data and Map files option

Note: To import data into the database, the server must be running before you start the InForm Data Import utility. To see the effect of imported data on the patient status icons, you must stop and restart the server after importing.

To import InForm data and map files into the InForm database:

1 Click **Start** > **All Programs** > **Oracle® Health Sciences** > **InForm 6.1** > **InForm Data Import**.

The InForm Data Import main window appears.

2 Select **InForm Data and Map files**.

3 In the **Trial Name** field, enter the name of the study into which to import the file. The last 10 studies you accessed appear in the drop-down list.

4 Click **Next**.

A dialog box appears where you specify the data and map files to import.

5 In the **Data** field, type the full path name of the data file you want to import, or click **Browse**.

6 In the **Map** field, type the full path name of the map file you want to import, or click **Browse**.

7 Optionally, to edit the map file, click **Edit Map File**. For more information, see *Editing an existing data and map file* (on page 152).

8 Click **Next**.

A dialog box appears and requests your InForm name and password.

Note: If you selected **Parse Only**, you do not need to specify a name and password.

9 In the **Name** field, type the name of an InForm user who has the appropriate rights for the data you are importing:

To import data that matches this InForm system activity	User needs these rights
Add subject clinical data	Enter Data into a CRF
Update subject clinical data	Edit Data on a CRF

10 In the **Password** field, type the user password, and click **Next**.

The Summary window appears.

11 Optionally, select any of the following:

- **Stop on Error**—To instruct the InForm Data Import utility to stop if it encounters an error.
- **Verbose**—To instruct the InForm Data Import utility to generate detailed messages as it processes the file.
- **Use output file**—Specify the filename to save the output file as a text file.

12 Click **Start**.

The InForm Data Import utility processes the import file, writes messages to the message area and the output file, if specified, and adds or updates data in the database.

13 Close the InForm Data Import utility.

Checking the error file

- Open the output file with the filename that you specified, in the directory that you specified.

Note: If you did not specify a directory and filename for the output file, the error file is saved in the same directory in which the InForm Data Import utility executable, PFImport.exe, is stored. After the import is complete, check for the presence of an ERR file and review the file for errors.

Date and time validation

The InForm Data Import utility performs the following validation checks for datetime data:

Validation check	Description
DISPLAY attribute of CRF datetime control.	The MedML **DISPLAY** attribute must be set to True for the components of the CRF datetime control into which a date, time, or datetime item is being imported. The InForm Data Import utility ignores date and time components for which the **DISPLAY** attribute is False, and it does not import them.
Year value of import field.	If the imported year is outside the range allowed in the CRF datetime control, the InForm Data Import utility issues an error.
Day value of import field.	If the Year and Month values are valid, the InForm Data Import utility checks for a valid day value.
REQUIRED attribute of CRF datetime control.	If any datetime component is coded as **REQUIRED** in the CRF datetime control and is missing in the import file, the InForm Data Import utility issues an error.
UNKNOWN attribute of CRF datetime control.	If an imported datetime component is coded UNK, but the **UNKNOWN** attribute of the CRF datetime control is not enabled, the InForm Data Import utility issues an error.
Consistency checking.	If consistency checking is enabled in the CRF datetime control, the InForm Data Import utility issues an error if a datetime component is present in the import file without a higher component, or an UNK value, in the following sequence: `ss:mm:hh dd/mm/yy` For example, if the imported datetime includes a Day field without a Month, the InForm Data Import utility issues an error.
Hour and Minute values of import field.	The values for both Hour and Minute must be within valid ranges or coded as UNK.

Running the InForm Data Import utility from the command line

While you are creating or editing your map file, use the InForm Data Import utility in interactive mode. After your map file is stable, you can run the utility in the PFConsole utility, from the command line, or you can insert the import command in a batch file that is scheduled to run periodically.

Note: Oracle recommends that you run the InForm Data Import utility through the PFConsole utility. For more information, see ***Running the PFConsole utility from the command line*** (on page 3).

Use the following command line parameters. You must include a space between the parameter name and its value.

```
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>]
```

Note: All command line applications, such as the MedML Installer utility, the InForm Data Import utility, and the InForm Data Export utility use the default product locale specified during the InForm installation, or through PFAdmin commands.

Parameter	Variable	Description
PFImport		Starts the InForm Data Import utility.
-?		Displays command help.
-autorun		Runs the InForm Data Import utility in a command window.
-verbose		The InForm Data Import utility will write detailed messages as it processes the input file.
-parse		The import will run without actually importing data into the database. Use this function to test the syntax of your MedML file before it is imported into the database.
-trial	*trialname*	The study into which you are importing data. Use the full pathname of the study.
-errstop		The InForm Data Import utility will stop processing when it encounters an error. When errstop is not specified, the tag containing the error is skipped and the import continues with the next data tag in the file.
-norules		Rules will not run during the import. This parameter is required. This option is valid only if there are no synchronization connections defined.

Parameter	Variable	Description
/accountparams "*path_to_password_file*"		When specified, includes the path to a text file that contains the user name and passwords required to run the command. If the accountparams option is not specified, the command prompts for the required user names and passwords. The format of the parameter file is parameter=value. There is a new line for each parameter, and there are no spaces on a line. You must manually create the text file. **Note:** Oracle recommends that you do not use this option. The utility prompts you to enter a username and password when you run the utility from the command line. This is the most secure way to provide a password.
-validate		Checks to make sure that all required XML tags exist and the specified control paths can be found, without loading data. Optionally, use this parameter to validate an XML file before importing it.
@	*RSP_file*	You are submitting a response file (RSP extension) that contains the names of the XML files to process. Specify the full pathname of the response file. Use this option or the -xml option or the -template and -import options.
-xml	*xml_file*	You are submitting one or more XML files to process. If you are submitting multiple XML files, separate them with a space. Use this option, the @ option, or the -template and -import options.
-template	*map_file*	If you are using the data and map file option, indicates that the next parameter is the map file name. Specify the full pathname of the map file. Use this option or the @ option or the -xml option. If you specify a map file, you must also specify an import file with the -import option.
-import	*data_file*	The next parameter is the import file name. Specify the full pathname of the data file. Use this option or the @ option or the -xml option. If you specify an import file, you must also specify a map file with the -template option.
-datetimeformat		Format to use for date and time information. Use this parameter to override the study configuration settings for date order format. Valid options are: • MMDDYYYY • DDMMYYYY • YYYYMMDD

Parameter	Variable	Description
-output	output_file	Full path and file name you want to give to the file. When you use this parameter, output messages are not displayed in the Output window.

Enhancing your data import

- **Log out of the InForm application server**—Log out of the InForm application server before running the import. You can keep the server running.
- **Stop the WWW Publishing Service**—Stop and restart the WWW Publishing Service to prevent others from entering data as you perform an import.
- **Run the Oracle update statistics script**—Run the Oracle update statistics script immediately after you import files.
- **Change the Home page**—Before you start the import, change the Home page of the study and warn users that they may experience slowness in the system while you are running the import.
- **Organize by subject**—If possible, sort the data you are importing by subject ID.

CHAPTER 4

InForm Data Export utility

In this chapter

Overview of the InForm Data Export utility

InForm Data Export utility output options

- **CDD**—Exports data into a Customer-Defined Database.
- **Name Value Pairs**—Creates a pipe-delimited file that consists of data path names and data values.

Note: Use of the InForm Data Export utility is limited to studies with fewer than 100,000 subjects.

You can use the CRF Submit application to export data in PDF format. The CRF Submit application is a stand-alone utility that allows you to:

- Create PDFs of subject data for FDA submission purposes. For example, you can create PDFs of subject data to submit to the FDA along with a New Drug Application (NDA).
- Archive PDFs of clinical data internally. For example, you can archive all clinical data for a single subject prior to moving the subject to a new site.

Running the InForm Data Export utility

All the InForm Data Export utility options run under the same executable.

Note: To run the InForm Data Export utility, the InForm application server does not need to be running.

To run the InForm Data Export utility:

1 Click **Start** > **Programs** > **Oracle® Health Sciences** > **InForm 6.1** > **InForm Data Export**.

The InForm Data Export main window appears.

2 In the **Books** section, select one of the following:

- Export All Books
- Export Frozen Case Books
- Export Locked Case Books

3 In the **Output Options** section, select a format to use for the export:

- CDD
- Name Value Pairs

For more information, see *InForm Data Export utility output options* (on page 170).

4 In the **Trial Name** field, type the name of the study from which to export information.

5 In the **Log Output Name** field, type the path and file name to give the log file that is created during the export, or click **Browse** and locate the file to use as the log file.

6 Click **Next**.

The following sections describe the input screens for each type of export:

- *Exporting data into a CDD* (on page 172).
- *Running the export for name value pairs* (on page 176).

Exporting data into a CDD

Overview of exporting data into a CDD

The CDD output option reads the CDD mapping parameters and all the data in the InForm database, then populates the CDD in the following order:

- Site data
- Comments
- Subject information
- CRF data

This is useful for repopulating the database when the CDD definition changes after a study starts.

Moving data to a CDD

1 Use the Central Designer application to generate a CDD definition that includes parameters that specify how to map data from the InForm database to the CDD.

2 Run the InForm Data Export utility using the CDD output option to read the data in the InForm database and populate the CDD database.

For more information, see *Running the export for CDD data* (on page 172).

Running the export for CDD data

Note: Before you run the InForm Data Export utility for CDD data, you should first stop the InForm application server.

1 Back up the existing CDD.

2 In the **Books** section, select one of the following:

- Export All Books
- Export Frozen Case Books
- Export Locked Case Books

3 In the **Output Options** section, click **CDD**.

4 In the **Trial Name** field, type the study name.

5 In the **Log Output Name** field, type the name of the log file to which to save the export, or click **Browse** to locate the file.

The CDD Export Options dialog box appears.

6 In the **DSN** field, do one of the following:

- Enter the name of the ODBC DSN that is defined for the CDD to which to output data.
- Create a new DSN.

For more information, see *Creating a new DSN for the export* (on page 173).

7 In the **User Name** and **Password** fields, type the user name and password that is used to access the CDD.

8 To drop the current database and create a new schema based on the RefName associated with the DSN, select the **Create New Schema** checkbox.

9 In the **Table Space** field, type the name of the tablespace in which to put the new schema.

10 Click **Next**.

The Export Summary screen appears.

WARNING: Make sure that all the information is correct before you proceed to the next step. When you run the InForm Data Export utility, all data in the current database will be lost.

11 Click **Finish**.

The InForm Data Export utility begins to populate the CDD and displays messages to indicate its progress.

Creating a new DSN for the export

1 In the CDD Export Options window, type the following information:

- In the **DSN** field, type the name for the ODBC DSN to create.
- In the **User Name** and **Password** fields, type the user name and password for accessing the CDD.

2 Click **Create DSN**.

The CDD Data Source window appears. Your DSN appears in the **CDD DSN** field.

3 From the **CDD RefName** drop-down list, select the RefName to associate with the new DSN.

4 In the **Database Server** field, type the name of the server to use.

5 Click **OK**.

The CDD Export Options window reappears, and the DSN, user name, and password appear in the appropriate fields.

Note: If you create a new DSN for the export, the new DSN is valid only in the InForm Data Import session you are in when you create it. It will be removed when you close the InForm Data Export utility. To create a DSN to use outside of the InForm Data Export utility, use the ODBC Manager or PFAdmin CONFIG CDD Setup commands. For more information on PFAdmin commands, see the *Study and Reporting Setup Guide*.

Exporting name value pairs

Overview of exporting name value pairs

The Name Value output option exports data from the InForm database into a plain text file. It is useful for extracting data for conversion or analysis.

Note: The Name Value output option exports only named values that occur in subject data; it does not export values from the Reg Docs or Visit Reports forms.

Output file format

In the output file that is created by running the export for name value pairs, fields are separated by pipes, and inapplicable fields are left blank, using the following data:

Name Value field	Description
Subject identifier	Subject initials followed by subject number in parentheses.
RefName path	Path of RefNames in the following order: • Visit. • Repeating formset index, which is used to indicate which instance of a recurring unscheduled visit is referenced. If the visit is not an unscheduled visit, the value is 1.000. • Form. • Repeating form index. • Section. • Itemset. • Itemset index, which is used to indicate which row of an itemset is referenced. • Item. • Control.
Normalized value	Value after conversion into the base units that are specified in the database.
Entered value	Value as entered for the control.

Output file format for associated forms

In the output file that is created by running the export for associated forms, fields are separated by pipes, and inapplicable fields are left blank, using the following data:

Associated form field	Description
Subject identifier	Subject initials followed by subject number in parentheses.
RefName path	Path of RefNames in the following order: • Visit. • Repeating formset index, which is used to indicate which instance of an unscheduled visit is referenced. If the visit is not an unscheduled visit, the value is 1.000. • Form. • Repeating form index. The path for the first form appears in this order, and the path for the associated form appears after the first form, surrounded by double quotation marks.
Normalized value	Value after conversion into the base units that are specified in the database.
Entered value	Value as entered for the control.

Example

The following export file fragment illustrates some of the data exported for the subject PA1(1).

```
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
```

The following export file fragment is an example of a repeating form:

```
ABC(101)|CommonCRF|1.000|L_ConMeds|291130301735008.000|L_ConMeds||0.000|CMHidde
nJ.CMHiddenJ||test
```

The following export file fragment is an example of an association:

```
BBB(111)|CommonCRF|1.000|LAE1|319514859682043.000| "BBB(111)|
CommonCRF|1.000|L_ConMeds|319587403827450.000"
```

Running the export for name value pairs

1 In the **Books** section, select one of the following:

- Export All Books
- Export Frozen Case Books
- Export Locked Case Books

2 In the **Output Options** section, click **Name Value Pairs**.

3 In the **Trial Name** field, type the study name.

4 In the **Log Output Name** field, type the name of the log file to which to save the export, or click **Browse** to locate the file.

The Name Value Pairs Export Options dialog box appears.

5 In the **File Name** field, type the name of the output file, including the NV extension, or click **Browse** to locate the file.

6 In the **Deleted Itemset Rows** section, select one of the following:

- To instruct the InForm Data Export utility not to output data from deleted forms or itemset rows, select **Do not output data from deleted Forms or Itemset rows**.
- To export data from deleted forms or itemset rows:
  1 Select **Output Deleted Forms and Itemset Rows**.
  2 In the **Deleted Prefix** box, specify the prefix to add to each deleted itemset row that the InForm Data Export utility exports.

7 Click **Next**.

The InForm Data Export utility builds the output file and displays messages to indicate the progress of the extraction.

Running the InForm Data Export utility from the command line

To run the InForm Data Export utility from the command line, use the following parameters.

Note: Oracle recommends that you run the InForm Data Export utility through PFConsole utility. For more information, see ***Running the PFConsole utility from the command line*** (on page 3).

Note: All command line applications, such as the MedML Installer utility, the InForm Data Import utility, and the InForm Data Export utility use the default product locale specified during the InForm installation, or through PFAdmin commands.

Parameter	Variable	Description
PFConsole utility		Starts PFConsole utility.
PFExport		Starts the InForm Data Export utility.
/accountparams "*path_to_password_file*"		When specified, includes the path to a text file that contains the user name and passwords required to run the command. If the accountparams option is not specified, the command prompts for the required user names and passwords. The format of the parameter file is parameter=value. There is a new line for each parameter, and there are no spaces on a line. You must manually create the text file. **Note:** Oracle recommends that you do not use this option. The utility prompts you to enter a username and password when you run the utility from the command line. This is the most secure way to provide a password.
-autorun		Runs the InForm Data Export utility in a command window. For more information, see *Running the PFConsole utility from the command line* (on page 3).
-?		Displays the usage statement.
-help		Displays the online Help for the InForm Data Export utility.
-Trial	trial name	The name of the study from which to export data.

Parameter	Variable	Description
-outfile	output log file name	Indicates that the next parameter is the name of an output file. Full pathname of an output file that contains the text of messages displayed by the InForm Data Export utility. Optional.
-cdd	ODBC DSN Name	Indicates that you are exporting data into a CDD. ODBC DSN name of the CDD to use as the export target of the CDD Output tool is required.
-CreatDSN	OraConnStr	For CDD export only, creates a new DSN within the CDD database.
-RefName	CDDRefName	For CDD export only, drops the current user and creates a new schema that is defined by the RefName.
-TBSP	OraTabSpace	For CDD export only, the name of the tablespace in which to put the new schema when creating a new DSN within the CDD database.
-crfhelp		Indicates that you want to export CRFHelp.
-nvfile	output nv file name	Indicates that you are running the InForm Data Export utility for name value pairs. Name of the export file that is created when you run the export tool for name value pairs is required if you specify the –nvfile flag.
-DelPrefix	string to prefix data marked as deleted	String that is added to a row of data in the export file to indicate that the data had been deleted from the InForm application prior to the export.
-OutputUN		Outputs UN for unknown date fields, if present.
-sysadmin		Name of the System Administrator with rights to export CRF Help. Indicates that the next parameter is the System Administrator ID. This ID is necessary to export CRFs.
-outputtype	ALL|EPIC|STANDARD	Indicates the type of export.
-customheader	text of header	For PDF format only, optional. Allows you to indicate the name of a custom header.

Example

The following is a sample syntax for running the InForm Data Export utility from the command line:

```
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 file format

This section of a DEL file shows the records for an investigator key that was changed from 6666 to 5555. This delete file was exported with comma-separated values turned on, meaning that a delete record is generated for every Visit/Visit Date for this investigator's subjects.

```
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,
```

CHAPTER 5

InForm Performance Monitor utility

In this chapter

Overview of the InForm Performance Monitor utility

The InForm Performance Monitor utility runs on the desktop where an InForm application server is running. When a session of the InForm application is active, the InForm Performance Monitor utility listens for and captures InForm application server messages about specific types of InForm activities, and displays the messages in a window where you can arrange and save them. You can use this utility to:

- Provide information that helps with performance tuning during study development and implementation.
- Capture performance data for troubleshooting.

Note: The InForm Performance Monitor utility is not intended for extended use in a production environment because it can impact server performance over time. Oracle suggests running the InForm Performance Monitor utility at 5 to 10 minute intervals.

Starting the InForm Performance Monitor utility

- Click **Start** > **Programs** > **Oracle® Health Sciences** > **InForm 6.1** > **InForm Performance Monitor**.

  The Performance Monitor window opens. If one or more InForm servers on the machine are running, the InForm Performance Monitor utility begins recording InForm application server messages.

Note: The InForm Performance Monitor utility is used by Oracle to assess and enhance the performance of a study, and evaluate server activities by quantifying the amount of time each activity requires. Although a brief description of its messages is detailed in ***Capturing performance statistics*** (on page 184), the use and interpretation of the InForm Performance Monitor utility is reserved by Oracle.

Capturing performance statistics

For each activity in a session of the InForm application, the InForm Performance Monitor utility captures the following data and displays it in the Performance Monitor window:

- **Trial ID**—Specifies the study that is associated with an InForm Performance Monitor utility message. If a message is not study-specific, it contains a value of 0.
- **Request ID**—Contains a numeric value. The InForm Performance Monitor utility messages with the same, non-zero RequestID value are associated with some common activity.
- **Time (ms)**—Elapsed time in milliseconds from the user-issued request to the server response.
- **Time of transaction**—The time at which the message was posted, in Greenwich Mean Time (GMT).
- **Tag**—Text string that describes the activity that is associated with the message.
- **Message**—Text string that provides detailed information that relates to the activity.

Viewing messages from specific subsystems

1 Select **View** > **Options**.

The Performance Monitor Options window appears.

2 In the **Messages to View** section, select one or more of the following filters to receive messages from only the specified subsystems.

- To select multiple subsystems, hold down the **Ctrl** key while you select.
- To select all options, click **Select All**.
- To clear the list of options, click **Clear All**.

Filter	Subsystem
IIS Request Data	The InForm ISAPI.
Trial package	The InForm study MTS package, by the InForm application server.
Rule package	The InForm rule package, by the InForm application server.
CDD package	The InForm CDD package, by the InForm application server.
UDA	The InForm database.
ODBC	The InForm ODBC.
Synchronization	Not supported.
Reports	Not supported.
Listing	The InForm listing.
InForm service	The InForm service.
PFImport	The InForm Data Import utility.

3 In the **Minimum Time** section, type the minimum number of milliseconds an action should take in order to be included in the list of messages.

4 Click **OK**.

Viewing messages from specific InForm servers

1 Select **View** > **Servers**.

The InForm Server Selection List dialog box appears.

2 Select the server from which to capture messages.

- To select multiple servers, hold down the **Ctrl** key while you select.
- To select all servers, click **Select All**.
- To clear the list of servers, click **Clear All**.

3 Click **OK**.

InForm Performance Monitor utility output options

Note: Before you select an output option, select the statistics to capture as described in ***Capturing performance statistics*** (on page 184).

Select any of the following output options for the InForm Performance Monitor utility messages:

Output option	Description	Action
View output online	Displays messages in the Performance Monitor window. (Default.)	Select **Output** > **Output To Window**
Stream output to a file	Sends messages to a file instead of (or in addition to) displaying them in the Performance Monitor window.	Select **Output** > **Stream To File**
Save a performance log	Saves the performance log as comma-separated text to a specified file.	Select **File** > **Save As**

Managing the InForm Performance Monitor utility data

Task	Action
Sort a column	Click the column heading bar.
Select a single message	Click the message.
Select multiple contiguous messages	Hold down the **Shift** key while clicking the messages.
Select multiple noncontiguous messages	Hold down the **Ctrl** key while clicking the messages.
Select all messages	Select **Edit** > **Select All**.
Copy selected messages	Select **Edit** > **Copy**, or click the **Copy** button in the toolbar.
Delete selected messages	Select **Edit** > **Delete**, or click the **Delete** button in the toolbar.
Clear all messages	Select **Edit** > **Reset**.

Examples of using the InForm Performance Monitor utility

Testing rule script efficiency

After you develop a set of rules for a study and add subject data to your test database, you can run the InForm Performance Monitor utility to:

- Verify that rules are firing when expected, and that they are running against the expected rule contexts.
- Check for unusually long rule execution times.
- Determine how much of a transaction submit time is occupied by rule processing.

Capturing rule processing data

1 In the Performance Monitor window, select **View** > **Options**.

The Performance Monitor Options dialog box appears.

2 Clear all filters except **Rule Package** to set the message filter to capture rule data.

3 Click **OK**.

4 Start the InForm application for your study.

5 Navigate to an item that is associated with a rule.

6 Edit the item, but do not click **Submit**.

7 In the Performance Monitor window, click **Edit > Reset** to clear the display.

8 In the InForm application, click **Submit**.

9 Return to the Performance Monitor window and check the messages.

Messages appear, indicating that:

- The rules fired as expected.
- The contexts for the rules were present.

The InForm Performance Monitor utility records the times that are required to check dependencies and to run the rules and calculations, as well as the total time to process rules on the form.

Note: To sort the messages by request times, click the **Time (ms)** column header. When you are gathering data on multiple rules, sorting by time can highlight rules that require unusually large amounts of time to process.

Reviewing SQL query performance

If response times have degraded as your study database accumulates data, you can use the InForm Performance Monitor utility to isolate long-running SQL queries. If you find long-running SQL queries, perform any of the following:

- Distribute tablespaces differently.
- Improve indexing on certain database tables.
- Run update statistics on the database.

Note: The InForm Performance Monitor utility is not intended for extended use in a production environment because it can impact server performance over time. Oracle suggests running the InForm Performance Monitor utility at 5 to 10 minute intervals.

Capturing SQL query performance data

1 From the Performance Monitor window, select **View** > **Options**.

The Performance Monitor Options dialog box appears.

2 Clear all filters except **UDA** to set the message filter to capture rule data.

3 Click **OK**.

4 Start the InForm application.

5 Navigate to an item that is associated with a query.

6 Edit the item, but do not click **Submit**.

7 In the Performance Monitor window, select **Edit > Reset** to clear the display.

8 Select **Output** > **Stream to File** to instruct the InForm Performance Monitor utility to stream the messages to a file, and specify the file name in which to store the messages.

9 In the InForm application, click **Submit**.

10 Return to the Performance Monitor window and check the messages.

If requested to do so by Oracle support, send them the saved message file.

Note: To filter the stream of messages by the time required to execute the request, use the **Minimum Time** section of the Performance Monitor Options dialog. The InForm Performance Monitor utility displays only messages for requests that require more than the specified number of milliseconds to execute.

To determine a normal request time, enter 0 in the **Discard actions less than** field in the **Minimum Time** section, and compare the data in the **Time** field on the Performance Monitor window for each captured message.

CHAPTER 6

InForm Report Folder Maintenance utility

In this chapter

Overview of the InForm Report Folder Maintenance utility

The InForm Report Folder Maintenance utility is a Windows application that can do the following:

- Copy Reporting and Analysis folders and their contents to target folders.
- Define and create folder structures when using a single Reporting and Analysis server for any of the following:
  - Multiple studies.
  - Multiple sponsors.
  - Multiple studies within multiple sponsors.
- Update any links to drill-down reports in InForm standard reports that were copied to a target location.
- Associate a copied report with a new Reporting and Analysis package at the target location.

  Note: This includes saved reports as well as standard report definitions.

You must have the Modify System Configuration right to run the InForm Report Folder Maintenance utility.

You install the InForm Report Folder Maintenance utility on the Reporting and Analysis (Cognos) server as part of the Reporting and Analysis installation and configuration. You can find it at this path:

```
\c10\bin\PFMTRSetupUtil.exe
```

The InForm Report Folder Maintenance utility copies only reports and report definitions. It does not copy the folders that contain the InForm Trial Management package, or any published study-specific clinical package. For more information about the association between reports and report packages, see *Report package association* (on page 199).

Note: You must configure a study in the Reporting and Analysis module before performing the procedures in this chapter. For more information, see the *Study and Reporting Setup Guide*.

Folder structure for multiple studies or sponsors

Setting up the initial folder structure

1 Import the standard reports archive. This archive includes all standard reports, as well as the InForm Trial Management package.

2 Publish a study-specific clinical package for clinical reports.

After you complete these steps, the InForm Report Folder Maintenance utility creates folders on the reporting server for the standard reports, and the reporting packages. These folders appear in the **Public Folders** tab of the Reporting and Analysis portal. The following folders appear after you initially set up a single study:

- CRF Reports
- Item Reports
- Query Reports
- Subject Reports
- Audit Trail Reports
- InForm Trial Management packages
- <*study_name*> clinical package

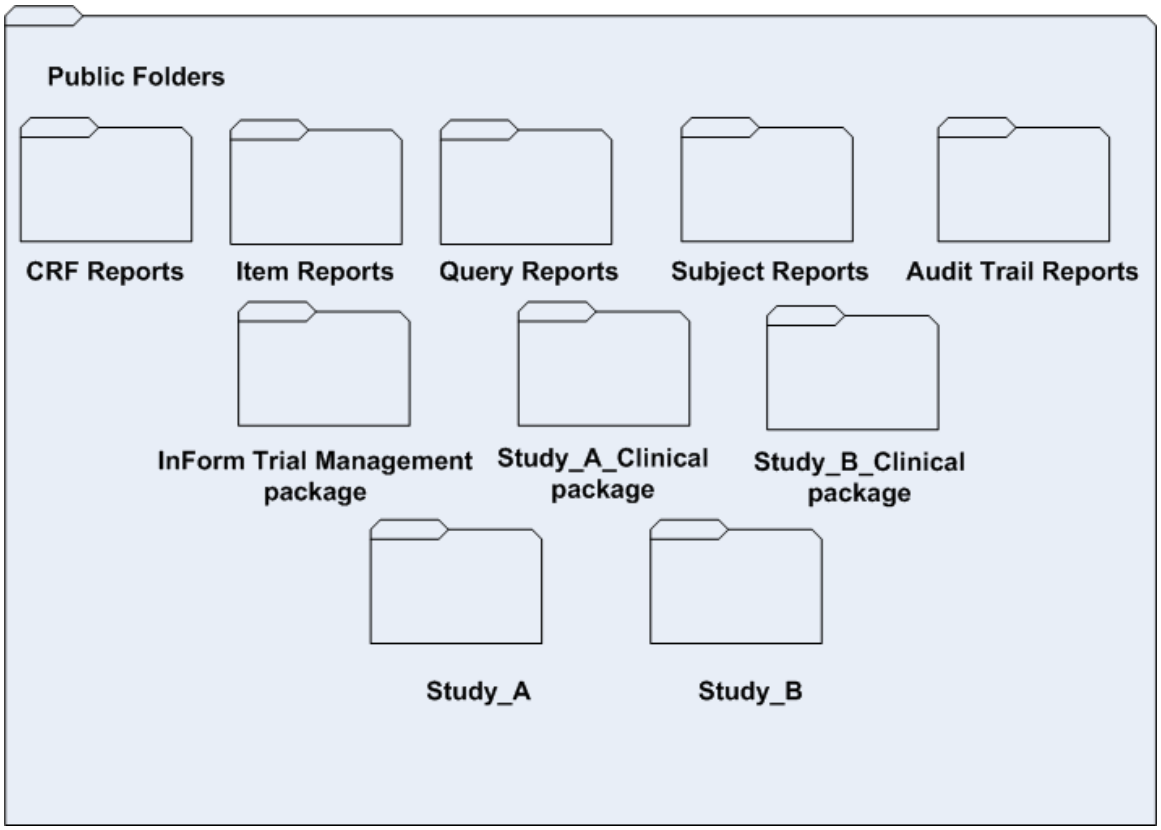
Folder structure for multiple studies

To set up several studies for a single sponsor, consider creating a separate folder for each study. PFRInit, the utility that you run when you set up the Reporting and Analysis module for a study, does this automatically.

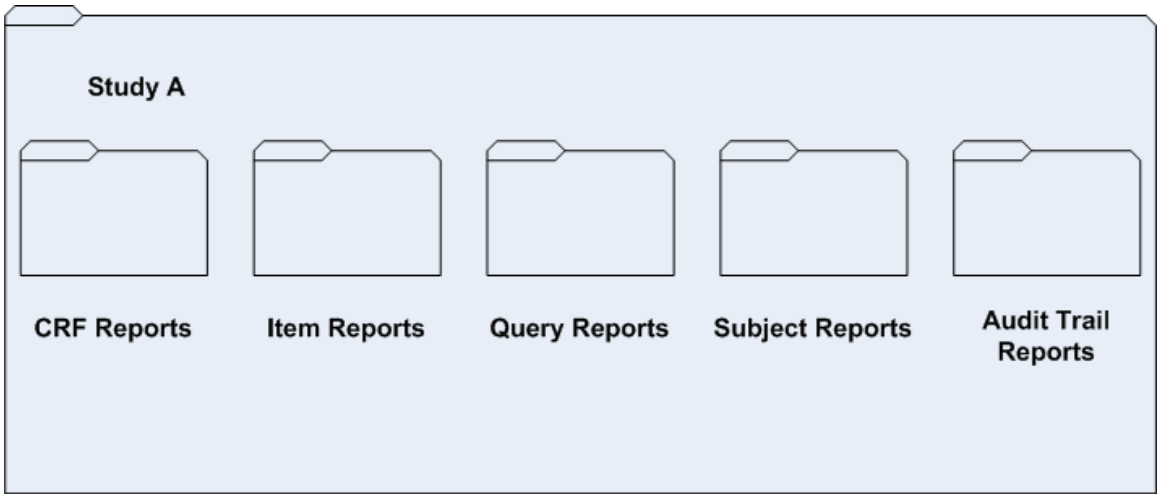
The following illustration shows what the **Public Folders** portal might contain in the Reporting and Analysis module with folders for two studies (Study A and Study B).

Note that this configuration uses three different packages:

- **InForm Trial Management package**—Shared by both Study_A and Study_B.
- **Study__A_Clinical package**—Used only for Study_A.
- **Study__B_Clinical package**—Used only for Study_B.



The contents of each study-specific folder are set up to include the default reporting folder structure.



This structure ensures that you can save study-specific properties, such as prompt values and report schedules, for a specified report.

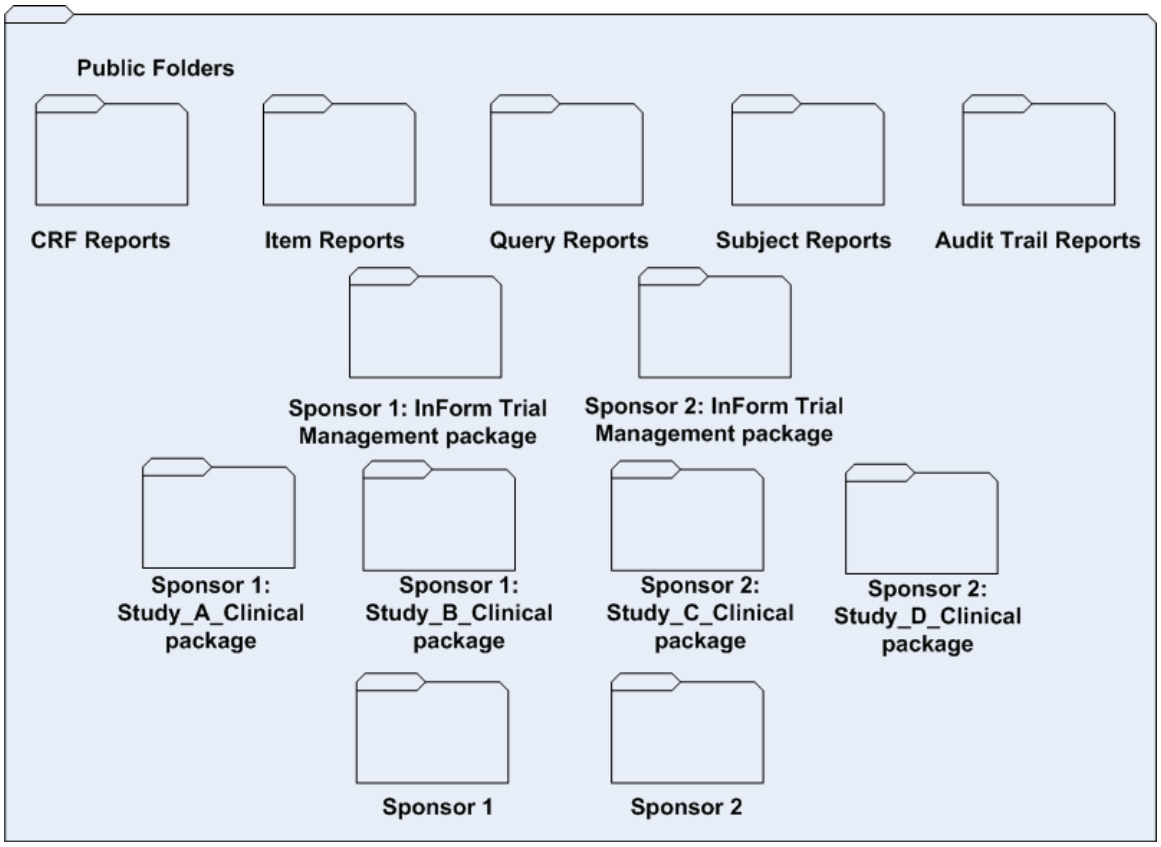
Folder structure for multiple sponsors, multiple studies

To set up folders for several sponsors on one reporting server, consider creating subfolders for each sponsor under the **Public Folders** tab. You can only do this if you use the InForm Report Folder Maintenance utility to perform all the configuration steps.

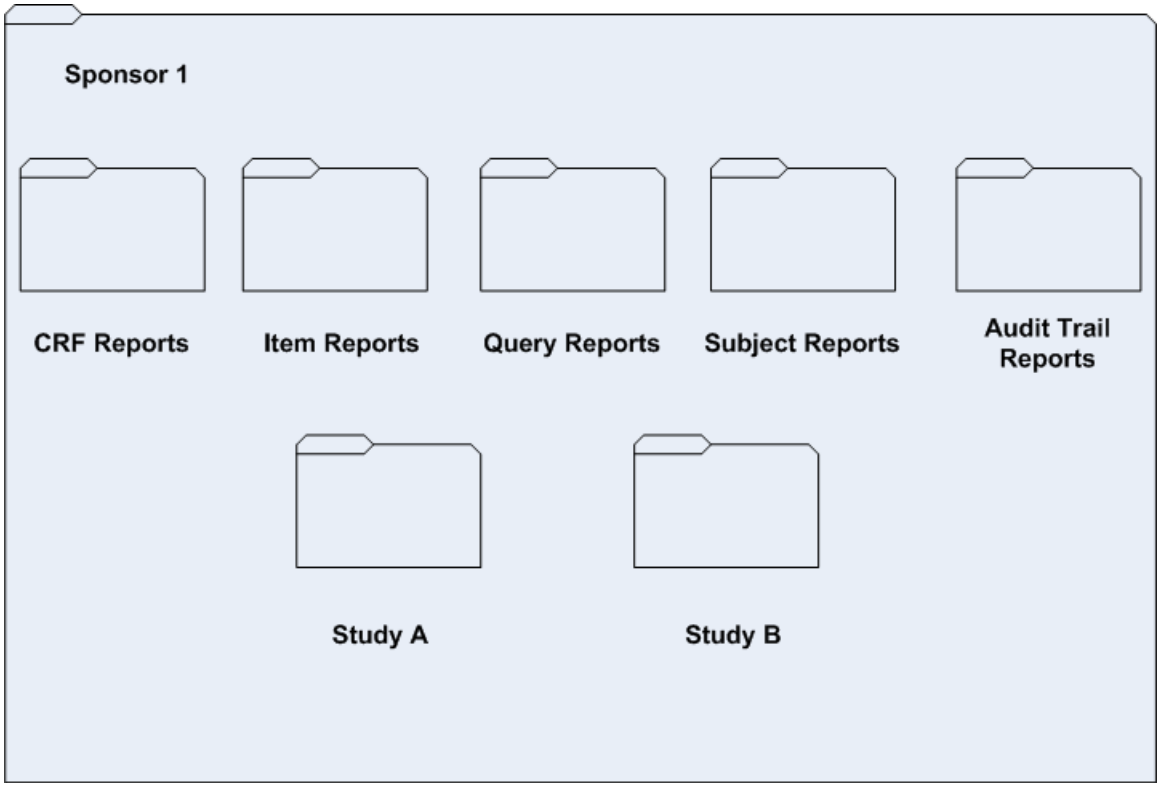
The following illustration shows how the **Public Folders** might look in the Reporting and Analysis module with folders for two different sponsors, each hosting two different studies.

The following example shows:

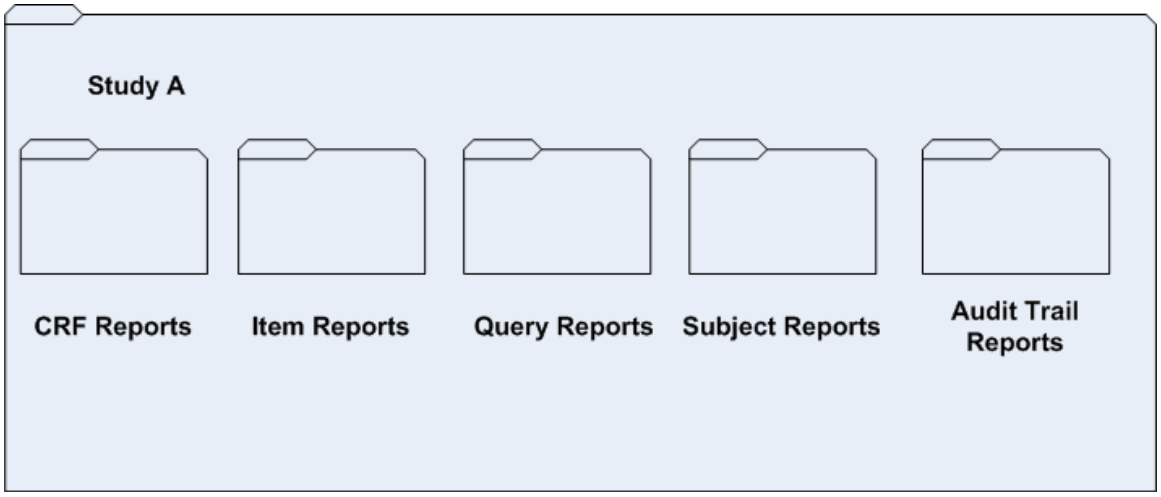
- Four clinical packages: two for Sponsor 1 and two for Sponsor 2.
- Two InForm Trial Management packages: one for Sponsor 1 and one for Sponsor 2.



The contents of each sponsor-specific folder are set up to include the default reporting structure for a single study. The following illustration shows the contents of the Sponsor 1 folder:



The Study_A folder contains all of the default standard report folders.



Setting up a folder structure for multiple studies or sponsors

Setting up report packages

All report packages must reside at the **Public Folders** level only; you cannot add a package to a subfolder. Therefore, if you are using several packages for different sponsors and studies, folders for each of these packages will appear in the **Public Folders** tab in the Reporting and Analysis portal.

Study-specific clinical package

Each study uses a unique study-specific clinical package. For details on how to publish a study-specific clinical package, see the *Study and Reporting Setup Guide*.

The InForm Trial Management package

Studies that share a sponsor can share an InForm Trial Management package. This way, any revisions or updates to the package automatically apply to all studies for the sponsor.

You import the InForm Trial Management package by importing the operational model and Reports deployment archive. When you imported this archive as part of your Reporting and Analysis installation, you imported both the standard reports folders and the InForm Trial Management package.

However, to revise packages for individual studies, you must import a study management package for each study.

To import InForm Trial Management packages for different sponsors or studies:

1 Log in to the study as the InForm system administrator.

2 Click **Reports**.

3 Select **Launch** > **Reporting Administration**.

The Administration page appears.

4 Click the **Configuration** tab.

5 Click **Content Administration**.

6 For the **Op package and reports** entry, click **Set properties**.

7 Click the **Import** tab.

8 Select the **InForm Trial Management** checkbox. Make sure that all other options are deselected.

9 Click the pencil icon next to **InForm Trial Management**.

10 Change the name of the InForm Trial Management package to reflect the sponsor or study that will be using the package.

11 Click **OK**.

12 Click **Return**.

The Administration page appears.

13 For the package you created, click the **Run with options - [package name]** icon ( ).

The Run with options - [package name] page appears.

14 In the **Time** section, specify when to run the import.

15 In the **Report Specification upgrade** section, select one of the following:

- Upgrade all report specifications to the latest version.
- Keep the existing report specification versions.

16 Click **Run**.

Creating new folders for multiple studies or sponsors

Report package association

Each report is associated with the package (either the InForm Trial Management package or a study-specific package) that was used to create it.

When you copy reports using the InForm Report Folder Maintenance utility, you can specify the package name to be used, if necessary. A report that you have copied to a new location must be associated with the package that will be used by the study or sponsor who works with the report.

For reports that you created with the InForm Trial Management package:

- **Multiple studies, single sponsor**—You are not required to specify a new package name; the InForm Trial Management package is installed with every study.
- **Multiple studies, multiple sponsors**—Each sponsor should provide an instance of the InForm Trial Management package; therefore, you may need to specify a new package name for the copied reports in the target location.

For reports that you created with the study-specific clinical package, you must always specify a new package name for the copied reports in the target location.

Report validation

The InForm Report Folder Maintenance utility validates copied reports against the associated packages when you copy the reports. Therefore, the packages must exist before you begin the copy operation. For more information, see *Setting up report packages* (on page 198).

The InForm Report Folder Maintenance utility stops when it encounters an error. You must correct the error, delete all objects within the target folder, and run the utility again to copy all reports. For example, although you can copy report definitions that contain clinical report elements from one study to another, the schema of the clinical portion of the Reporting and Analysis database is unique to each study; therefore, the clinical package that is generated for each study is unique. If a clinical report contains a report element that does not exist in the target package, the report cannot be validated after it is copied.

Report copy considerations summary

Source Package	Type of data in report	New package name required?	Report modifications required at the target location?
Study-specific clinical package	Study management data only.	Yes.*	No.
Study-specific clinical package	Clinical data, or a mixture of clinical and trial management data.	Yes.*	Depends on the clinical packages and their elements. You may have to alter the report to ensure that it can be validated and copied to the target location.
Single sponsor, multiple studies			
InForm Trial Management package	Study management data only.	No.	No.
Multiple sponsors, multiple studies			
InForm Trial Management package	Study management data only.	Yes. Each sponsor folder should have its own InForm Trial Management package.	No.

*The package must exist before you can copy the folders.

Copying report folders

1 On the Reporting and Analysis server, double-click the following file:

```
\crn\bin\PFMTRSetupUtil.exe
```

The InForm Report Folder Maintenance window appears.

2 Enter the following information:

- **User Name** and **Password**—User name and password to log in to the Reporting and Analysis server.
- **LDAP Namespace**—Name of the LDAP namespace that is used for authentication by Sun One Directory Server.
- **Copy from**—Click **Standard report folders** to copy standard report folders, or click **Custom folders** to copy custom report folders.
- **Path**—Type the path for the existing folder structure to copy. Do not include the folder name here.

  You can copy the source path from the source folder properties.
- **Folder prefix**—Specify the prefix that identifies the folders to copy. For example, if you used S1 as a prefix for all folders belonging to Study 1, specify S1 in the **Folder prefix** field to instruct the InForm Report Folder Maintenance utility to copy the folders with the S1 prefix.
- **Path**—Type the target path for the folder. Do not include the folder name.
- **New Folder prefix**—Specify a prefix for the target folder name. For example, if you are creating the folder structure for Study 2, you might type S2 as the prefix for the new folder.
- **New Folder name**—Type the name of the target folder if it is different from the name of the source folder. You can specify a new folder name (the folder will be created) or an existing folder.
- **Package name**—Type the name of the new package that should be associated with any study-specific reports. Since each study clinical package is unique, reports that contain clinical (study-specific) data must be associated with a new clinical package when you copy them to a new location.

  Note: The package must exist before you perform the copy.

3 Click **Create**.

The InForm Report Folder Maintenance utility begins to copy the folder structure to the new location. The **Status** section displays ongoing status and notifies you when the copy is complete.

Note: You can identify the top-level report folder for your company in the **Admin > System Configuration > Reporting user root** field in the user interface. This is the folder you see when you log in to the Reporting and Analysis module.

APPENDIX A

Sample data import XML

In this appendix

Overview of sample data import XML

The import file for the MedML file option is an XML file that contains tags that specify the type of processing to perform during import, and the import data destinations and values. Cut and paste these samples and use any text editor that creates plain text files to edit the data to create your own import files.

- *Importing screening and enrollment data* (on page 205).
- *Importing new subject clinical data* (on page 206).
- *Updating existing subject clinical data* (on page 207).
- *Deleting an Add Entry itemset* (on page 208).
- *Undeleting an Add Entry itemset* (on page 209).
- *Adding data to an unscheduled visit* (on page 210).
- *Transferring subject records* (on page 211).

Importing screening and enrollment data

This sample file below contains the necessary data tags to import screening and enrollment data for subject XYZ at site PF.

```
<?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>
```

Importing new subject clinical data

The sample file below contains the necessary data tags to import clinical data to subject XYZ at site PF.

```
<?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>
```

Updating existing subject clinical data

The sample file below contains the necessary data tags to edit or clear existing data about a subject.

```
<?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>
```

Deleting an Add Entry itemset

The sample file below contains the necessary data tags to delete an existing Add Entry itemset.

Note: You cannot delete or undelete a Repeating Data itemset.

```
<?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>
```

Undeleting an Add Entry itemset

The sample file below contains the necessary data tags to recover a deleted Add Entry itemset.

Note: You cannot delete or undelete a Repeating Data itemset.

```
<?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>
```

Adding data to an unscheduled visit

The sample file below contains the necessary data tags to add data to an unscheduled visit for a particular subject.

```
<?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>
```

Transferring subject records

This example shows the elements in a MedML file that is used to transfer several subjects. Note that each pair of PATIENTSITECHANGE attributes defines current and destination site information for one subject.

```
<?xml version="1.0"?>
<CLINICALDATA>
    <PATIENTSITECHANGE REASON="new subject address">
        <NEWSITE SITEMNEMONIC="MCLEAN"/>
        <CURRENTSITE SITEMNEMONIC="PF" PATIENTNUMBER="1003"/>
    </PATIENTSITECHANGE>
```

<!----For subject 1001 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 subject 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 Oracle site.>

```
    <PATIENTSITECHANGE REASON="new subject address">
        <NEWSITE SITEMNEMONIC="MCLEAN"/>
        <CURRENTSITE SITENAME="Phase Forward" PATIENTINITIALS="DDD"
            DUPLICATEORDER="2"/>
    </PATIENTSITECHANGE>
</CLINICALDATA>
```

About the documentation

Where to find the product documentation

The product documentation is available from the following locations:

- **My Oracle Support** (https://support.oracle.com)—*Release Notes* and *Known Issues*.
- **Oracle Technology Network** (http://www.oracle.com/technetwork/documentation/hsgbu-154445.html)—The most current documentation set, excluding the *Release Notes* and *Known Issues*.

If the software is available for download, the complete documentation set is available from the Oracle Software Delivery Cloud (https://edelivery.oracle.com).

All documents may not be updated for every InForm release. Therefore, the version numbers for the documents in a release may differ.

Documentation accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Documentation

Document	Description	Part number	Last updated
Release Notes	The *Release Notes* document describes enhancements and issues fixed in the current release, and other late-breaking information.	E61316-01	6.1.1
Known Issues	The *Known Issues* document provides detailed information about the known issues in this release, along with workarounds, if available.	E61318-01	6.1.1

Document	Description	Part number	Last updated
Secure Configuration Guide	The *Secure Configuration Guide* provides an overview of the security features provided with the Oracle® Health Sciences InForm application, including details about the general principles of application security, and how to install, configure, and use the InForm application securely.	E61320-01	6.1.1
Upgrade and Migration Guide	The *Upgrade and Migration Guide* provides instructions for upgrading and migrating the InForm software and InForm Portal software to the current InForm release, and for upgrading the Cognos software for use with the Reporting and Analysis module.	E61321-01	6.1.1
Installation Guide	The *Installation Guide* describes how to install the software and configure the environment for the InForm application and Cognos software.	E61322-01	6.1.1
Study and Reporting Setup Guide	The *Study and Reporting Setup Guide* describes how to perform the tasks that are required to set up an InForm study and configure the Reporting and Analysis module for the study.	E61323-01	6.1.1
User Guide	The *User Guide* provides an overview of the InForm application including details on multilingual studies, how to navigate through the user interface, how to manage a study-specific Home page with the InForm Portal application, and how to accomplish typical tasks you perform while running a clinical study. This document is also available from the user interface.	E61324-01	6.1.1
Reporting and Analysis Guide	The *Reporting and Analysis Guide* provides an overview of the Reporting and Analysis module. It includes a brief overview of the Reporting and Analysis interface, illustrates how to access the InForm Ad Hoc Reporting workspace, and describes the study management and clinical data packages available for creating reports. It also provides detailed descriptions of each standard report that is included with your installation.	E61326-01	6.1.1
Reporting Database Schema Guide	The *Reporting Database Schema Guide* describes the Reporting and Analysis database schema, and provides information on creating Reporting Database Extracts (RDEs).	E61327-01	6.1.1

Document	Description	Part number	Last updated
Utilities Guide	The *Utilities Guide* provides information about and step-by-step instructions for using the following utilities: • PFConsole utility • MedML Installer utility • InForm Data Import utility • InForm Data Export utility • InForm Performance Monitor utility • InForm Report Folder Maintenance utility This guide also provides reference information for the MedML elements and scripting objects that are used to import and export data to and from the InForm application, as well as sample data import XML.	E61328-01	6.1.1
MedML Installer utility online Help	The MedML Installer utility online Help provides information about, and step-by-step instructions for using, the MedML Installer utility, which is used to load XML that defines study components into the InForm database. This guide also provides reference information for the MedML elements and scripting objects that are used to import and export data to and from the InForm application, as well as sample data import XML. This document is also available from the user interface.	NA	NA
InForm Data Export utility online Help	The InForm Data Export utility online Help provides information about and step-by-step instructions for using the InForm Data Export utility, which is used to export data from the InForm application to the following output formats: • Customer-defined database (CDD) • Name value pairs This document is also available from the user interface.	NA	NA

Document	Description	Part number	Last updated
InForm Data Import utility online Help	The InForm Data Import utility online Help provides information about and step-by-step instructions for using the InForm Data Import utility, which is used to import data into the InForm application. This document is also available from the user interface.	NA	NA
Clinical Data API Guide	The *Clinical Data API Guide* provides information about submitting data to the InForm application in InForm ODM format.	E61329-01	6.1.1
Third Party Licenses and Notices	The *Third Party Licenses and Notices* document includes third party technology that may be included in or distributed with this product.	E61330-01	6.1.1
Secure Development Guide	The *Secure Development Guide* provides an overview of common security risks for developers using Application Programming Interfaces (APIs) with the Oracle® Health Sciences InForm application, and information on how to address those risks.	E72493-01	6.1.1