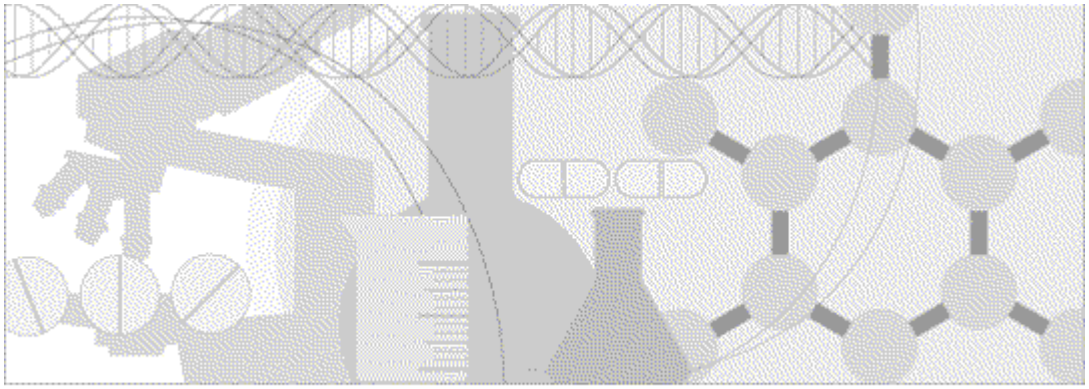
Integration Guide

Oracle® Health Sciences InForm Publisher Release 2.1.1



ORACLE

Part number: E96216-01

Contents

CHAPTER 1

InForm Publisher integration overview

In this chapter

What is InForm Publisher?

The InForm Publisher software is an application programming interface (API) for the InForm application. It supports integrations with Oracle and third-party applications by pulling selected data from the InForm application when triggered by specific data entered for a subject in the InForm application.

Like other APIs, the InForm Publisher interfaces use published web services interfaces to allow programmatic access to applications. This allows Oracle products to be tightly integrated with each other and with third-party products.

The data is published from a transaction queue, on a configurable schedule. To use the InForm Publisher software with an InForm study, you must:

1 Install the InForm Publisher software on the InForm application server.

2 Configure the software to work with the study and the target application.

For more information, see the *Installation Guide*.

The InForm Publisher interfaces

The InForm Publisher release includes the following interfaces that are available for programmatic access.

Interface	Description	For more information
Argus Safety Publisher	Publishes adverse event data for a study in InForm to the Oracle Argus Safety application, based on a data mapping in the study design that is configured in Central Designer. You can configure settings in InForm Publisher related to number of attempts and frequency of data publishing, and error logging.	*Oracle® Health Sciences InForm to Argus Integration: Study Design Requirements* *Oracle® Health Sciences InForm Publisher 2.1.1 Installation Guide*
ODM Extract Publisher	Provides access to receive InForm clinical data and metadata in CDISC ODM standard format. The ODM Extract Publisher can publish data to web services, local directory, FTP, or CTMS endpoints. InForm Publisher provides a way for customers to define events of interest and select the output format for publishing data from InForm studies. Transactions are based on custom events (for example, data added to a subject's study data or changes in subject status). A custom event is set up to provide input for applications requiring InForm data or status, and to provide the specific information needed as determined by customer receiving application requirements. The output adheres to the ODM Snapshot format.	*Oracle® Health Sciences InForm Publisher 2.1.1 Installation Guide*

CHAPTER 2

Working with custom events

In this chapter

Overview of custom events

What are custom events?

A custom event is a transaction initiated by a trigger event that causes data, metadata, status, specific information, and reason for transmission to be exported from an InForm study to one or more known destinations. A custom event includes:

- A trigger event, which initiates publishing. For example, enrollment of a subject can be a trigger event. Each custom event has at least one trigger.
- A definition of the data or status to be published.
- A destination.
- A priority. The default value is 1.
- Event output XML, which is sent to the receiving application.
- Optionally, a custom event can include a prerequisite event on which it depends. If an event is dependent on another event, the dependent event is only reported after the prerequisite event has been evaluated and only if the most recent state of the prerequisite event is true.

Output is in CDISC standard ODM Snapshot format. When you are working with Oracle Services to design a study, you can include multiple custom events, each with its own triggers, results, and targeted destination.

Custom events support a variety of customer applications, including metric reporting, clinical coding, clinical repositories, Oracle Health Sciences IRT On Demand, and Siebel Clinical Trial Management System On Demand.

Using custom events, you can:

- Create trigger events based on simple occurrences in the InForm application such as field changes, form or visit completions, or subject status changes.
- Allow a selective amount of data to be provided from a study when an event is triggered.
- Provide the latest data or status of a subject.
- Select data for the extract during study design or setup.

Elements of custom events

A custom event must contain the following:

- **Event Name**—A name that is given to the event and is exported with the custom event (for example, Coding Verbatim or CTMS New Subject Added).
- **Trigger Event**—An event in an InForm study that initiates publishing the custom event. The trigger supports all form items (hidden fields, calculated fields, and checkboxes).
- **Event Destination**—A name assigned to the destination (receiving) application of a custom event (for example, Customer Coding System). The application maps the name to the actual web service URL of the application.

  Note: The destination is transformed by the InForm Publisher application to the equivalent of an application URL or other web service endpoint.

- **Event Priority**—A value assigned to the event to specify its level of priority over other custom events. The default value is 1.
- **Event Result**—The set of data and status updates that is exported when a trigger event occurs (for example, Adverse Event, study, site, subject, form, and Inform coding target names created with Central Designer mappings).
- **Prerequisite event**—Optionally, an event on which the custom event depends.

Output of custom events

The InForm Publisher application evaluates and publishes custom events periodically. The frequency of evaluation and publication is configurable, but must be at least every 24 hours and no more than every 5 minutes. The default is every 15 minutes.

The event result is compatible with the InForm ODM Transaction Export format produced by InForm Adapter.

The data is transmitted as an outgoing SOAP message through a web service call or in a file sent by SFTP (Secure File Transfer Protocol). Output can also be written to a directory on the InForm application server. For configuration information, see the *Installation Guide*.

Since the receiving application needs the ODM metadata to interpret the event result, the InForm Publisher application exports the metadata.

- When SOAP is used, the receiving application can return a response code to tell the InForm Publisher application to send the metadata and administrative data after the event XML is received.
- When SFTP is used, a special property file at the receiving end tells the InForm Publisher application whether to send the metadata and administrative data.
- When any endpoint type except CTMS is used, metadata and administrative data can be obtained by specifying type and frequency of metadata and administrative data in InForm Publisher configuration.

Alternately, the receiving application can obtain the metadata and administrative data through the InForm Adapter ODM interface.

The InForm Publisher application maintains a history of the exports of custom events. When the

InForm Publisher application cannot evaluate or publish a custom event, the application retries evaluation and publication and saves the failures so that the custom event is not lost.

Because ODM extract messages published by the InForm Publisher application might contain patient data and must conform to HIPAA requirements, the data transmitted from the Oracle Cloud for Industry (OCI) hosted environment is sent over an encrypted transport channel.

Custom event triggers

Each custom event must have a trigger that initiates the publishing of the custom event.

Trigger events can be based on occurrences in the InForm application such as a change to the value in a field, form or visit completion, or a change to subject status. These triggers are saved in the study database. The InForm Publisher application periodically checks the transaction history to determine whether triggers occurred.

Unless noted in the Description column in the table below, an event results in an initial transmission when it first tests positive and an event results in a follow-up transmission when it tests negative after having previously been reported as positive.

Trigger	Associated with	Description
Data entered	Specific item	• Particular data for an item is entered or changed. • A data value change always results in a transmission.
Matches the codelist value	Specific control	A particular codelist value for a control is entered or changed.
Matches a query state: • Candidate • Opened • Answered • Closed • Deleted • Sponsor Conflict • Site Conflict	Specific item or form	The state of the query has changed to the state listed.
Source verification state change: Source Verified ← → Not Source Verified	Specific item	← → means that the state changes from one state to the other.
Itemset state change: • Delete ← → Undelete • Source Verified ← → Not Source Verified	Specific itemset	← → means that the state changes from one state to the other.

Trigger	Associated with	Description
Form state change: • Started • Complete ← → Incomplete • Frozen ← → Unfrozen • Locked ← → Unlocked • Signed ← → Unsigned • Source Verified ← → Not Source Verified • Deleted ← → Undeleted • SV Ready← → Not SV Ready • SV Partial ← → SV Complete	Specific form	• The state of the form has changed to the state shown. • ← → means that the state changes from one state to the other.
Data entry	Specific form	A data value change always results in a transmission.
Review state change	Specific form	The state of the form has changed to the state listed.
Visit state change: • Started • Complete ← → Incomplete • Frozen ← → Unfrozen • Locked ← → Unlocked • Signed ← → Unsigned • Source Verified ← → Not Source Verified	Specific visit	• The state of the query has changed to the state listed. • ← → means that the state changes from one state to the other.

Trigger	Associated with	Description
Subject state change: • Screened • Screen Failed • Enrolled • Enrollment Failed • Randomized • Complete ← → Discontinue • Site Transfer • Subject Number Change	Subject state	• The state of the subject has changed to the state shown. • ← → means that the state changes from one state to the other.

Limitations on triggers

Compound item—When a data change item trigger is set within a compound item, the trigger occurs when any items within the compound item gets edited.

Visit state triggers—These triggers (Frozen, Locked, Complete, and SV Complete) occur only when all expected forms in the visit have been started. Visit state should match the state in the InForm application.

Float data type—When a codelist value is float type and it includes decimal point with zero (.0), decimal point with zero (.0) cannot be included in the CODELISTVALUE column in the PFEX_ITEMTRIGGER table. Use an integer in this instance. For example, for 12.0, use 12.

Repeating instance output—Displays only the instance that triggered the output.

Combined criteria as triggers

The trigger event can be a combination of triggers connected by either AND or OR, but not both.

AND examples:

- Form is Complete AND form is SV'd.
- MRI date entered AND MRI Successful = Yes.
- Form1, Form2, AND Form3 are completed.

OR examples:

- On a specific form, such as the Demographic form, the OR could be used to specify if any field of a specific set of fields was entered or changed.
- Weight OR height OR age OR blood pressure OR . . .
- Visit1, Visit2, …OR Visit 16 is completed.

AND/OR rules

- An AND event trigger occurs when ALL triggers are positive.
- An OR event trigger occurs when any trigger is positive.
- An event results in an initial transmission when it first tests positive.
- An event results in a follow-up transmission when it tests negative after having previously been reported as positive.
- If any trigger element for an event includes a non-repeating type, then no elements in other triggers for that event may include a repeating type. Repeating types are items, such as repeating forms, visits, and itemsets, for which you can create multiple instances in the InForm application.
- If any trigger element for an event includes repeating types, all elements in other triggers for that event must include all of the same repeating types. Therefore, if you want to combine a non-repeating and a repeating type, Oracle suggests you create two separate events.
- If an OR event has all data change triggers at the item level, the InForm Publisher application sends a follow-up transmission when any of the triggers evaluates to true, even if the event OR trigger has already tested positive in the past.

AND/OR limitations

- Multiple subject states using AND/OR are not supported.
- Subject state can be combined with other triggers using AND/OR as long as the other triggers are not repeating types.

Combined triggers

Some triggers can only be used when combined with another trigger:

- **No open queries**—There are no queries on the form or that all queries are closed.
- **Item is empty**—An item is empty.
- **Item is not empty**—An item is not empty.

Defining data to export from the InForm application

Event result definition specifies the set of data and status updates the InForm Publisher application exports when an event occurs. You can specify what data to receive when an event is triggered. Data from multiple InForm objects may be specified for a single event.

You can request two types of output. Each has its own level of reporting available:

- **Status output**—Exports the current status of the specified object.
  - **Item Status**—Reports the current status of the specified item.
  - **Item Set Status**—Reports the current status of the specified item set.
  - **Visit Status**—Reports the current status of the specified visit.
  - **Subject Status**—Reports the current status of the specified subject.
  - **Form Status**—Reports the current status of the specified form.
- **Result output**—Exports the current results of the specified object.
  - **Item Result**—Reports the current data for the specified item in the context of the visit, form, section and itemset to which it belongs.
  - **Form Detail**—Reports the current detail for all items of the specified form and it includes all sections of the form.

Specifying aliases

You can specify an alias to use for mapping an item in the receiving application.

Custom data name and value pairs

The InForm Publisher application allows you to attach one or more name and value pairs to any event result. You can use this feature to send additional information to the receiving application.

Changes to custom events

You can modify the Event Alias, Event Destination, Trigger Event, Events Results, and Event Result refnames in an ongoing study. A change to a custom event takes effect from the time it is implemented in an InForm study. It does not affect previously sent events.

You can deactivate a custom event in an ongoing study.

How events are published

To use custom events, the InForm Publisher application must be installed on the InForm server. The InForm Publisher application:

- Monitors the InForm database for events that meet trigger event criteria.
- Retrieves data from the InForm application based on the fields specified in the event result.
- Pushes ODM Snapshot extracts to configured destinations through directory, web services, or SFTP.

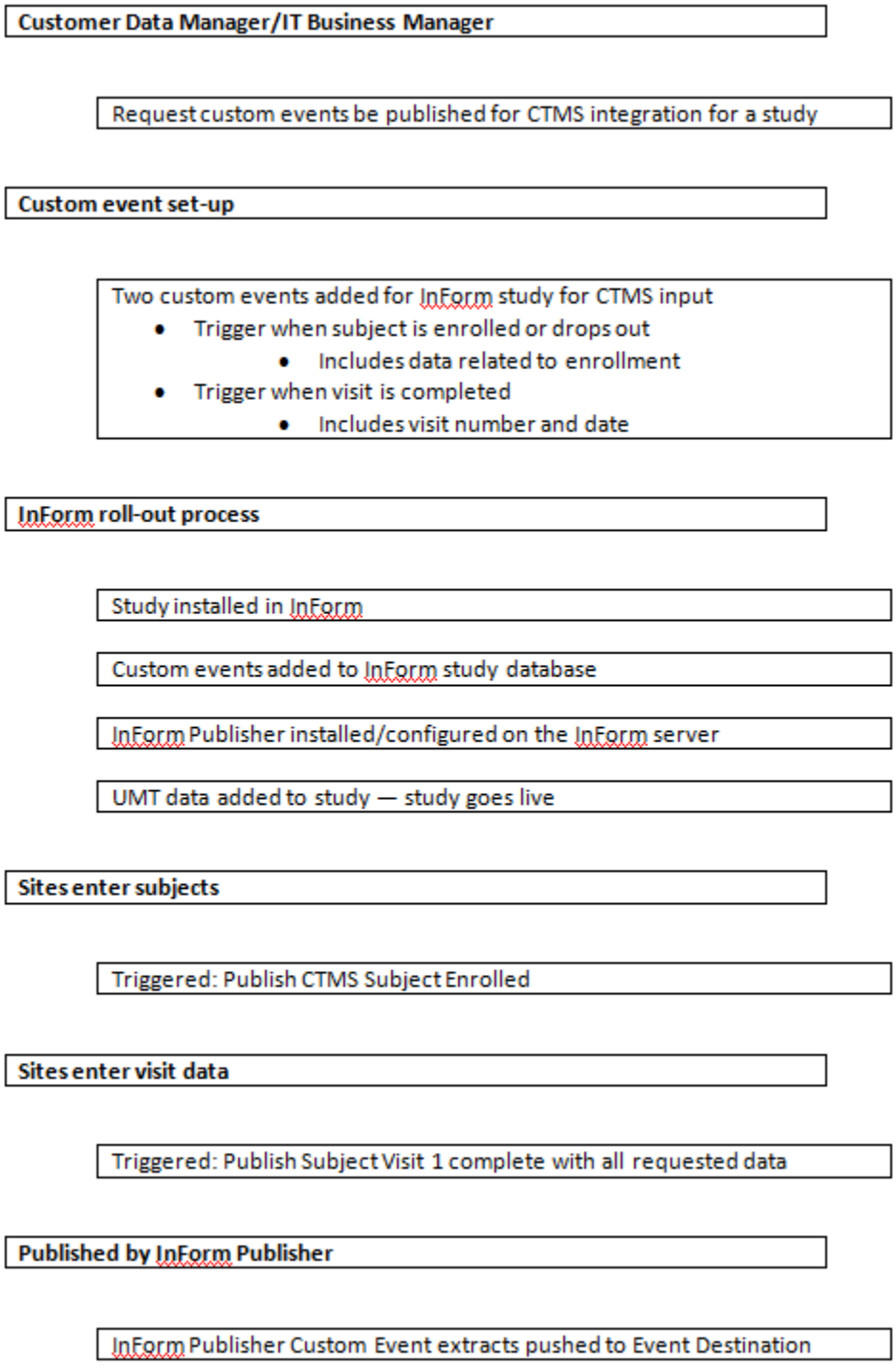
Note: Item blinding is not supported in ODM export. All selected items are exported.

Working with Oracle Services to implement custom event triggers

Customers must work with their Oracle Health Sciences Consulting (HSC) team, who can:

- Define custom event requirements for implementation in a study.
- Implement the custom event tables in the InForm database.
- Set configuration options.

The following shows the steps that might be followed to implement custom events for a sample project.


Customer Data Manager/IT Business Manager
Request custom events be published for CTMS integration for a study
Custom event set-up
Two custom events added for InForm study for CTMS input
Trigger when subject is enrolled or drops out
Includes data related to enrollment
Trigger when visit is completed
Includes visit number and date
InForm roll-out process
Study installed in InForm
Custom events added to InForm study database
InForm Publisher installed/configured on the InForm server
UMT data added to study — study goes live
Sites enter subjects
Triggered: Publish CTMS Subject Enrolled
Sites enter visit data
Triggered: Publish Subject Visit 1 complete with all requested data
Published by InForm Publisher
InForm Publisher Custom Event extracts pushed to Event Destination

CHAPTER 3

Defining how the receiving application receives ODM extract data

In this chapter

Using the web service

Custom event data is sent by the InForm Publisher application and received by target applications using web service transactions.

Writing the web service

ODMProcessorService Web Services Description Language (WSDL) defines how a receiving application receives ODM extract data from InForm Publisher. A receiving application must implement the web service using the WSDL provided. The web service defines a single operation, called receiveODMData.

The receiving application should provide implementation logic to receive ODM data and process it based on the application's requirements. Further processing might involve validating ODM data, storing the data in its database or file system, or transforming and sending the data to downstream applications.

High-level summary of the web service API

The following is a high-level summary of the web service API.

Service

Service	Operation
ODMProcessorService	receiveODMData

Operation—receive ODMdata

Service	Operation
ODMProcessorService	receiveODMData

Item	Description
Input	The InForm Publisher application sends ODM extract data XML in string format to the web service.
Output	The web service sends one of the following return codes to the InForm Publisher application:
SUCCESS*	After receiving ODM extract data, the receiving application sends a SUCCESS response.
ODMMETAREQUIRED	The receiving application can request ODM metadata.
ODMADMINREQUIRED	The receiving application can request ODM administrative data.

Item	Description
ODMMETAANDADMIN REQUIRED	The receiving application can request ODM metadata and administrative data.
Fault	If the web service is unable to process data for any reason, it returns a fault.

Note: The web service should return a success code after receiving either ODM extract data or ODM metadata and administrative data. The remaining codes should be used only when the receiving application needs administrative data, metadata, or both.

Sample web service request and response

Sample Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:test="http://test.odmextract.informpublisher.hsgbu.oracle.com/"
xmlns:s="s" xmlns:u="u">
   <soap:Body>
      <test:receiveODMData>
         <arg0><![CDATA[<ResponseODM
xmlns:pf="http://www.phaseforward.com/InFormAdapter/ODM/Extensions/3.0"
xmlns="http://www.phaseforward.com/InFormAdapter/ODM/3.0">
  <ODM Description="combo" FileType="Snapshot" CreationDateTime="2013-09-
16T18:37:54.0535222Z" ODMVersion="1.3.1" Originator="Oracle HSGBU"
SourceSystem="InForm" SourceSystemVersion="6.0" pf:MappingVersion="V3"
pf:HierarchicalOIDs="Yes" pf:Generator="InFormPublisher  v2.0.50727"
xmlns="http://www.cdisc.org/ns/odm/v1.3">
    <ClinicalData StudyOID="StudyDesign" MetaDataVersionOID="Study Design
0.0.4">
      <SubjectData SubjectKey="17647" pf:GUID="{4ABEE1F2-839F-44CC-B5FD-
81720BEE5A9A}">
        <SiteRef LocationOID="01" />
        <pf:SubjectStatus State="Enrolled" />
      </SubjectData>
      <pf:CustomEvent Name="TC100000027e" Destination="Service_1"
StudyVersion="Study Design 0.0.4" EventID="1" EventRev="1" />
    </ClinicalData>
  </ODM>
  <Status>OK</Status>
</ResponseODM>]]></arg0>
      </test:receiveODMData>
   </soap:Body>
</soap:Envelope>
```

Sample Response

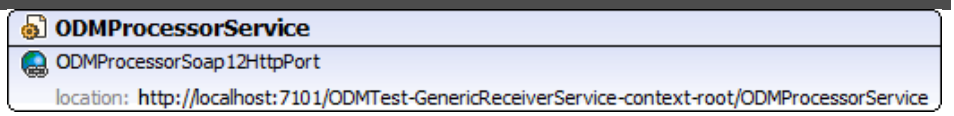
```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope">
  <S:Header>
    <To
xmlns="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressin
g/anonymous</To>
    <Action
xmlns="http://www.w3.org/2005/08/addressing">http://test.odmextract.informpublis
her.hsgbu.oracle.com/ODMProcessor/receiveODMDataResponse</Action>
    <MessageID xmlns="http://www.w3.org/2005/08/addressing">uuid:bfba08fd-ed9a-
4649-aa68-cddc94ed43a3</MessageID>
    <RelatesTo xmlns="http://www.w3.org/2005/08/addressing">urn:uuid:621b1526-
a917-4a8f-b27c-9d979683852c</RelatesTo>
  </S:Header>
  <S:Body>
```

```
    <ns2:receiveODMDataResponse
xmlns:ns2="http://test.odmextract.informpublisher.hsgbu.oracle.com/">
      <return>SUCCESS</return>
    </ns2:receiveODMDataResponse>
  </S:Body>
</S:Envelope>
```

WSDL documentation

Services

ODMProcessorService

Item	Description
Diagram	
Ports	ODMProcessorSoap12HttpPort
Binding:	***tns:ODMProcessorSoapHttp*** (on page 23)
Extensibility:	<soap12:address location="http://localhost:7101/ODMTest-GenericReceiverService-context-root/ODMProcessorService"/>

Source	`<wsdl:service name="ODMProcessorService">`
	`  <wsdl:port name="ODMProcessorSoap12HttpPort"`
	`binding="tns:ODMProcessorSoapHttp">`
	`  <soap12:address location="http://localhost:7101/ODMTest-`
	`GenericReceiverService-context-root/`
	`ODMProcessorService"/>`
	`  </wsdl:port>`
	`</wsdl:service>`

Bindings

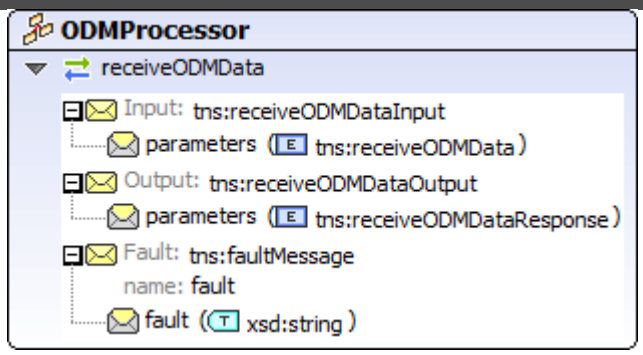
ODMProcessorSoapHttp

Item	Description
Diagram	
Type	***tns:ODMProcessor*** (on page 24)
Extensibility	<soap12:binding style="document" transport="http://www.w3.org/2003/05/soap/bindings/HTTP/"/>
Operations	receiveODMData
Extensibility:	<soap12:operation soapAction="http://test.odmextract.informpublisher.hsgbu.oracle.com/receiveODMData" soapActionRequired="false"/>

Input:	<soap12:body use="literal"/>
Output:	<soap12:body use="literal"/>
Used by	Port ***ODMProcessorSoap12HttpPort*** (on page 23) in Service ***ODMProcessorService*** (on page 23)
Source	`<wsdl:binding name="ODMProcessorSoapHttp" type="tns:ODMProcessor"> <soap12:binding style="document" transport="http://www.w3.org/2003/05/soap/bindings/HTTP/"/> <wsdl:operation name="receiveODMData"> <soap12:operation soapAction="http://test.odmextract.informpublisher.hsgbu.oracle.com/receiveODMData" soapActionRequired="false"/> <wsdl:input> <soap12:body use="literal"/> </wsdl:input> <wsdl:output> <soap12:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding>`

Porttypes

ODMProcessor

Item	Description
Diagram	
Operation	receiveODMData
Input:	***tns:receiveODMDataInput*** (on page 25)
Output:	***tns:receiveODMDataOutput*** (on page 25)
Fault:	***tns:faultMessage*** (on page 25)
Used by	
Binding:	***ODMProcessorSoapHttp*** (on page 23)

Source	<wsdl:portType name="ODMProcessor"> <wsdl:operation name="receiveODMData"> <wsdl:input message="tns:receiveODMDataInput" xmlns:ns2="http://www.w3.org/2006/05/addressing/wsdl" ns2:action="http://test.odmextract.informpublisher.hsgbu.oracle. com/receiveODMData"/> <wsdl:output message="tns:receiveODMDataOutput" xmlns:ns2="http://www.w3.org/2006/05/addressing/wsdl" ns2:action="http://test.odmextract.informpublisher.hsgbu.oracle. com/receiveODMDataResponse"/> <wsdl:fault name="fault" message="tns:faultMessage"/> </wsdl:operation> </wsdl:portType>

Messages

faultMessage

Item	Description
Parts	fault
Type:	xsd:string
Used by	
Operation:	***receiveODMData*** (on page 24)
in PortType:	***ODMProcessor*** (on page 24)
Source	<wsdl:message name="faultMessage"> <wsdl:part name="fault" type="xsd:string"/> </wsdl:message>

receiveODMDataInput

Item	Description
Parts	parameters
Element:	***tns:receiveODMData*** (on page 26)
Used by	
Operation:	***receiveODMData*** (on page 24)
in PortType:	***ODMProcessor*** (on page 24)
Source	<wsdl:message name="receiveODMDataInput"> <wsdl:part name="parameters" element="tns:receiveODMData"/> </wsdl:message>

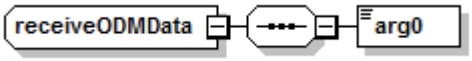
receiveODMDataOutput

Item	Description
Parts	parameters
Element:	***tns:receiveODMDataResponse*** (on page 27)

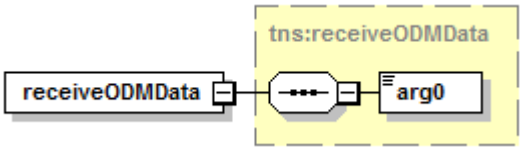
Used by	
Operation:	***receiveODMData*** (on page 24)
in PortType:	***ODMProcessor*** (on page 24)
Source	`<wsdl:message name="receiveODMDataOutput"> <wsdl:part name="parameters" element="tns:receiveODMDataResponse"/> </wsdl:message>`

Types

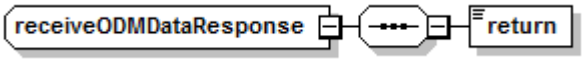
receiveODMData (Types and ComplexTypes)

Item	Description
Diagram	
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Children	***arg0*** (on page 30)
Used by	
Element:	***receiveODMData*** (on page 26)
Source	`<complexType name="receiveODMData"> <sequence> <element name="arg0" type="xsd:string"/> </sequence> </complexType>`

receiveODMData (Types and Elements)

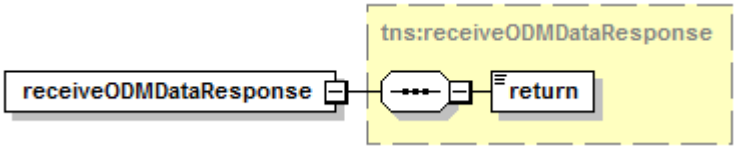
Item	Description
Diagram	
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Type	***tns:receiveODMData*** (on page 26)
Properties	
Content:	complex
Children	***arg0*** (on page 30)
Source	`<element name="receiveODMData" type="tns:receiveODMData"/>`

receiveODMDataResponse (Types and ComplexTypes)

Item	Description
Diagram	
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Children	***return*** (on page 29)
Used by	
Element:	***receiveODMDataResponse*** (on page 29)

Source	`<complexType name="receiveODMDataResponse">`
	`  <sequence>`
	`  <element name="return" type="tns:ReturnCodes"/>`
	`  </sequence>`
	`</complexType>`

receiveODMDataResponse (Types and Elements)

Item	Description
Diagram	
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Type	***tns:receiveODMDataResponse*** (on page 27)
Properties	
Content:	complex
Children	***return*** (on page 29)
Source	`<element name="receiveODMDataResponse"` `type="tns:receiveODMDataResponse"/>`

ReturnCodes

Item	Description
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Type	restriction of string
Properties	
Base:	xsd:string
Used by	
Element:	***receiveODMDataResponse/return*** (on page 29)
Facets	

Kind:	enumeration
Value:	SUCCESS
Annotation:	After receiving ODM extract data, the receiving application should send a SUCCESS response back to InForm Publisher by sending this return code.

Kind:	enumeration
Value:	ODMMETAREQUIRED
Annotation:	After receiving ODM extract data, the receiving application can request ODM metadata from InForm Publisher by sending this return code.

Kind:	enumeration
Value:	ODMADMINREQUIRED
Annotation:	After receiving ODM extract data, the receiving application can request ODM administrative data from InForm Publisher by sending this return code.

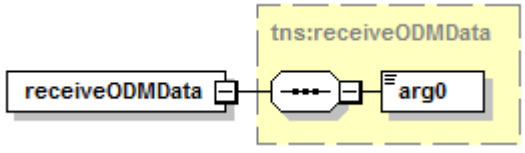
Kind:	enumeration
Value:	ODMMETAANDADMINREQUIRED
Annotation:	After receiving ODM extract data, the receiving application can request ODM metadata and administrative data from InForm Publisher by sending this return code.

Source

```
<simpleType name="ReturnCodes">
  <restriction base="xsd:string">
    <enumeration value="SUCCESS"/>
    <enumeration value="ODMMETAREQUIRED"/>
    <enumeration value="ODMADMINREQUIRED"/>
    <enumeration value="ODMMETAANDADMINREQUIRED"/>
  </restriction>
</simpleType>
```

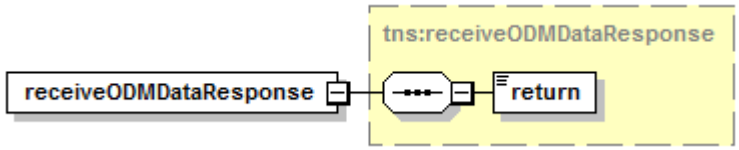
Elements

receiveODMData (Types and Elements)

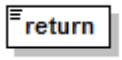
Item	Description
Diagram	
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Type	***tns:receiveODMData*** (on page 26)
Properties	

Content:	complex
Children	***arg0*** (on page 30)
Source	`<element name="receiveODMData" type="tns:receiveODMData"/>`

receiveODMDataResponse (Elements)

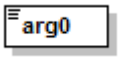
Item	Description
Diagram	
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Type	***tns:receiveODMDataResponse*** (on page 27)
Properties	
Content:	complex
Children	***return*** (on page 29)
Source	`<element name="receiveODMDataResponse" type="tns:receiveODMDataResponse"/>`

receiveODMDataResponse/return

Item	Description
Diagram	
Type	***tns:ReturnCodes*** (on page 27)
Properties	
Content:	simple
Facets	
Kind:	enumeration
Value:	SUCCESS
Annotation:	After receiving ODM extract data, the receiving application should send SUCCESS response back to InForm Publisher by sending this return code.
Kind:	enumeration
Value:	ODMMETAREQUIRED
Annotation:	After receiving ODM extract data, the receiving application can request ODM metadata from InForm Publisher by sending this return code.

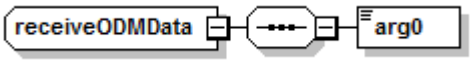
Kind:	enumeration
Value:	ODMADMINREQUIRED
Annotation:	After receiving ODM extract data, the receiving application can request ODM administrative data from InForm Publisher by sending this return code.
Kind:	enumeration
Value:	ODMMETAANDADMINREQUIRED
Annotation:	After receiving ODM extract data, the receiving application can request ODM metadata and administrative data from InForm Publisher by sending this return code.
Source	`<element name="return" type="tns:ReturnCodes"/>`

receiveODMData/arg0 (Element)

Item	Description
Diagram	
Type	string
Properties	
Content:	simple
Source	`<element name="arg0" type="xsd:string"/>`

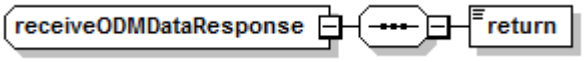
Complex types

receiveODMData (Types and ComplexTypes)

Item	Description
Diagram	
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Children	***arg0*** (on page 30)
Used by	
Element:	***receiveODMData*** (on page 26)

Source	`<complexType name="receiveODMData">`
	`  <sequence>`
	`  <element name="arg0" type="xsd:string"/>`
	`  </sequence>`
	`</complexType>`

receiveODMDataResponse (Types and ComplexTypes)

Item	Description
Diagram	
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Children	***return*** (on page 29)
Used by	
Element:	***receiveODMDataResponse*** (on page 29)

Source	`<complexType name="receiveODMDataResponse">`
	`  <sequence>`
	`  <element name="return" type="tns:ReturnCodes"/>`
	`  </sequence>`
	`</complexType>`

Simple types

ReturnCodes

Item	Description
Namespace	http://test.odmextract.informpublisher.hsgbu.oracle.com/
Type	restriction of string
Properties	
Base:	xsd:string
Used by	
Element:	***receiveODMDataResponse/return*** (on page 29)
Facets	
Kind:	enumeration
Value:	SUCCESS
Annotation:	After receiving ODM extract data, the receiving application should send a SUCCESS response back to InForm Publisher by sending this return code.
Kind:	enumeration
Value:	ODMMETAREQUIRED
Annotation:	After receiving ODM extract data, the receiving application can request ODM metadata from InForm Publisher by sending this return code.
Kind:	enumeration
Value:	ODMADMINREQUIRED

Annotation:	After receiving ODM extract data, the receiving application can request ODM administrative data from InForm Publisher by sending this return code.
Kind:	enumeration
Value:	ODMMETAANDADMINREQUIRED
Annotation:	After receiving ODM extract data, the receiving application can request ODM metadata and administrative data from InForm Publisher by sending this return code.

Source

```
<simpleType name="ReturnCodes">
  <restriction base="xsd:string">
    <enumeration value="SUCCESS"/>
    <enumeration value="ODMMETAREQUIRED"/>
    <enumeration value="ODMADMINREQUIRED"/>
    <enumeration value="ODMMETAANDADMINREQUIRED"/>
  </restriction>
</simpleType>
```

FTP option

To enable FTP publishing for ODM extract type subscriber, configure the following in the InFormPublisher.config file:

- EndPointType = "FTP"
- EndPoint—IP or hostname address for remote host.
- Username—Login credential for remote host.
- Password—Login credential for remote host.
- Local folder—Folder name (FtpLocalFolderPath) from which local files are picked up by ODM Extract SFTP transmitter.
- Remote folder—Folder name (RemoteFolder) on the remote host, where files will be sent.

FTP Attributes

```
.\PublisherAdmin.exe Subscriber set OESubcriber9
EndPoint=ftp://<hostname>:<portnumber> EndPointType=FTP FtpLocalFolderPath=<
local folder path> FtpRemoteFolderPath=<Remote folder path>
```

SecureEndPointCredentials

```
.\PublisherAdmin.exe Subscriber setCredentials OESubcriber9 User=<username>
<Enter password when prompted>
```

Requesting admin data and metadata

To write admin data or metadata to the Remote Folder, add a file called OdmConfig.properties to the local folder.

- If the file contains ReturnCode=1, the InForm Publisher application will send admin data during the next push interval.
- If the file contains ReturnCode=2, the InForm Publisher application will send metadata during the next push interval.
- If the file contains ReturnCode=3, the InForm Publisher application will send both admin data and metadata during the next push interval.
- If the file contains ReturnCode=4 or the ODMConfig.properties file does not exist, the InForm Publisher application will not sent either admin data or metadata during the next push interval.

Local Directory option

The InForm Publisher application can publish ODM Snapshot output file to a directory. This option is not available when Oracle hosts the application. However, customers who have the InForm application and the InForm Publisher application on premises can use this option.

CHAPTER 4

Export formats

In this chapter

ODM Snapshot format

The format of the of ODM custom event extract XML conforms to ODM 1.3.1 with new extensions that do not exist in the ODM XML from InForm Adapter. The ODM XML is in the Snapshot format and compatible with the transactional output. The custom events extract is in CDISC ODM Snapshot format. InForm ODM extensions designed for InForm Adapter ODM transactional output are also supported and have been incorporated into this guide. Audit trails are not included in the ODM Snapshot format.

Oracle extensions for ODM Snapshot

The ODM Extract Publisher is a unidirectional interface called by a customer web service from any outside application that needs to retrieve data from an InForm database in CDISC ODM standard format. Refer to www.cdisc.org for more information on the ODM standard.

The ODM Extract Publisher connects directly to the InForm database.

This section describes the Oracle extensions to the syntax and functionality of the ODM Extract Publisher of the InForm Publisher software. The ODM Extract Publisher focuses on the semantics of the function calls that are published by the interface, and the parameters that the published web methods use. Elements are included for:

- Custom events
- Clinical data
- Administrative data
- Metadata

Each section gives examples of XML for both the web method request and response.

ODM structure for custom event output

ODM element

The ODM element has the following attribute set for the predefined value in ODM Custom Event Extract output:

- pf:MappingVersion attribute is set to V3.
- ODMVersion attribute value is 1.3.1.
- FileType is set to Snapshot.
- FileOID is set to the new guid.
- Originator is set to "InForm Publisher."
- SourceSystem is set to "InForm."
- SourceSystemVersion is set to the database version of the InForm study for which the custom event extract is published.

ODM element example

```
<ODM Description="combosample" FileType="Snapshot" CreationDateTime="ignored"
ODMVersion="1.3.1" Originator="Oracle HSGBU" FileOID="" SourceSystem="InForm"
SourceSystemVersion="6.0" pf:MappingVersion="V3" pf:HierarchicalOIDs="Yes"
pf:Generator="InForm Publisher"
xmlns="http://www.cdisc.org/ns/odm/v1.3">
```

pf:CustomEvent element

This extension denotes the custom event for which the ODM XML is published.

XPath

```
ODM/ClinicalData/pf:CustomEvent
```

Attributes

Attribute	Description
Name	Custom context alias defined for this event.
Destination	Name of the destination that the InForm Publisher application uses to locate the endpoint to which it sends the custom event XML.
StudyVersion	Study version to which this custom event is assigned.
EventID	Internal numeric identifier of the custom event.
EventRef	Revision number of the custom event.

pf:CustomEvent example

```
<pf:CustomEvent Name="DataChangeTest" Destination="FrancaEndpoint"
StudyVersion="Study Design 0.0.9" EventID="18" EventRev="1" />
```

pf:Alias attribute

This attribute is the customer context alias for the event result included.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/@pf:
Alias
ODM/ClinicalData/SubjectData/StudyEventData/FormData/@pf:Alias
ODM/ClinicalData/SubjectData/pf:StudyEventStatus/@pf:Alias
ODM/ClinicalData/SubjectData/pf:SubjectStatus/@pf:Alias
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemGroupStat
us/pf:Alias
ODM/ClinicalData/SubjectData/StudyEventStatus/pf:FormStatus/pf:Alias
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/Item
Status/pf:Alias
```

pf:StudyEventStatus

This element reflects the visit status and includes the pf:FormStatus element for the form started in the visit.

XPath

```
ODM/ClinicalData/SubjectData/pf:StudyEventStatus
```

Attributes

Attribute	Description
StudyEventOID	Visit refname in the InForm application.
StudyEventRepeatKey	Visit index for the repeating visit. If the visit is not repeating, this attribute is not generated.
Complete	Yes=Visit is complete. No=Visit is incomplete.

Attribute	Description
Frozen	Yes=Visit is frozen. No=Visit is not frozen.
Lock	Yes=Visit is locked. No=Visit is unlocked.
SDV	Yes=Visit is source verified. No=Visit is not source verified.
Activated	Yes=Form is activated. No=Other status attributes, such as Complete, Frozen, and so on, are not generated.

pf:StudyEventStatus example

```
        <pf:StudyEventStatus Complete="No" Frozen="No" Lock="No" SDV="No"
StudyEventOID="vstUnschVisit" StudyEventRepeatKey="393232473548079">
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmCGI" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmECG" FormRepeatKey="393232471708079" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmHamD" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmVS" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmPE" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes" FormOID="frmDOV2" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmECG" FormRepeatKey="393232474285079" />
          <pf:CustomData Name="CTMSElementName" Value="VisitStatus" />
        </pf:StudyEventStatus>
```

MeasurementUnitRef

This element exists in the transactional output and will be populated for the item data if applicable.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/
MeasurementUnitRef
```

pf:NormalizedValue of ItemData

This extension attribute will be populated if there is a normalized value.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/
@NormalizedValue
```

Annotation

This element is either the form or item comment and is populated when there is a comment.

XPath for the Item comment

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/
Annotation
```

XPath for the form comment

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/
Annotation
```

pf:ItemGroupStatus

This extension element reflects the item group status.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/
ItemGroupStatus
```

Attributes

Attribute	Description
SVComplete	Yes=Itemset is source verified. No=Itemset is not source verified.
Deleted	Yes=Itemset is deleted. No=Itemset is not deleted.

pf:ItemGroupStatus example

```
<pf:ItemGroupStatus SVComplete="No" Deleted="No" />
```

pf:ReasonIncomplete

This extension element already exists in the transactional output. It will be populated in the custom event extract if the reason incomplete.

Form Reason Incomplete XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/Annotation/pf:ReasonIncompl
ete
```

Item Reason Incomplete XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/Anno
tation/pf:ReasonIncomplete
```

Critical attribute of ItemStatus

This element tells whether or not the item is SV critical in the InForm application. The value is True when the item is SV critical and False when it is not. Otherwise, it is StudyDefault.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/Item
```

```
Status/@Critical
```

CustomData

This is the extension element to include a string attached to an element in the custom event XML.

XPath

```
ODM/ClinicalData/SubjectData/pf:SubjectStatus/pf:CustomData
```

Example

```
   <pf:SubjectStatus State="Enrolled">
          <pf:CustomData Name="CustomDataName" Value="CustomDataValue" />
        </pf:SubjectStatus>
```

XPath

```
ODM/ClinicalData/SubjectData/pf:StudyEventStatus/pf:CustomData
```

Example

```
        <pf:StudyEventStatus Complete="No" Frozen="No" Lock="No" SDV="No"
StudyEventOID="vstBASE">
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes" FormOID="MEDHISTiSet" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes" FormOID="frmCGI" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes" FormOID="frmDOV" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmPEBase" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmHamD" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes" FormOID="frmDem" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes" FormOID="frmINCEXC" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmRAND" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmVS" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmPI" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes" FormOID="frmECGBase" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes" FormOID="frmPM" />
          <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="No" FormOID="frmPreg" />
          <pf:CustomData Name="CTMSElementName" Value="VisitStatus" />
        </pf:StudyEventStatus>
```

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/pf:CustomData
```

Example

```
          <FormData FormOID="frmDem">
            <ItemGroupData ItemGroupOID="frmDem.sctDemographics">
              <ItemData
ItemOID="frmDem.sctDemographics.DateofBirth_DEM.DateofBirth_DEM" Value="1975-12-
25T-:-:-+00:00" pf:FormattedDateValue="1975-12-25TUNK:UNK:UNK+00:00"
pf:Alias="vstBase.frmDem" />
              <ItemData ItemOID="frmDem.sctDemographics.Race.Race" Value="3"
pf:Alias="vstBase.frmDem" />
              <ItemData
ItemOID="frmDem.sctDemographics.ScreeningDate_DEM.ScreeningDate_DEM"
Value="2011-10-26T-:-:-+00:00" pf:FormattedDateValue="2011-10-
26TUNK:UNK:UNK+00:00" pf:Alias="vstBase.frmDem" />
              <ItemData ItemOID="frmDem.sctDemographics.Height.Height"
Value="153.0" pf:NormalizedValue="153" pf:Alias="vstBase.frmDem">
                <MeasurementUnitRef MeasurementUnitOID="cm" />
                <Annotation SeqNum="1" pf:DBUID="20401" pf:GUID="{7725F52A-204D-
4A07-A4D9-1980F0D41C97}">
                  <Comment>2nd comment</Comment>
                </Annotation>
              </ItemData>
              <ItemData ItemOID="frmDem.sctDemographics.OnsetAge.OnsetAge"
Value="37.0" pf:Alias="vstBase.frmDem" />
              <ItemData ItemOID="frmDem.sctDemographics.Age.Age" Value="46"
pf:Alias="vstBase.frmDem" />
              <ItemData ItemOID="frmDem.sctDemographics.Gender.Gender" Value="2"
pf:Alias="vstBase.frmDem" />
              <ItemData ItemOID="frmDem.sctDemographics.AgeGroup.AgeGroup"
Value="4" pf:Alias="vstBase.frmDem" />
              <pf:InFormItemData
InFormItemOID="frmDem.sctDemographics.DateofBirth_DEM" Name="DateofBirth_DEM">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData InFormItemOID="frmDem.sctDemographics.Race"
Name="Race">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData
InFormItemOID="frmDem.sctDemographics.ScreeningDate_DEM"
Name="ScreeningDate_DEM">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData InFormItemOID="frmDem.sctDemographics.Height"
Name="Height">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData InFormItemOID="frmDem.sctDemographics.OnsetAge"
Name="OnsetAge">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData InFormItemOID="frmDem.sctDemographics.Age"
Name="Age">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData InFormItemOID="frmDem.sctDemographics.Gender"
Name="Gender">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData InFormItemOID="frmDem.sctDemographics.AgeGroup"
Name="AgeGroup">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
            </ItemGroupData>
            <ItemGroupData ItemGroupOID="frmDem.PatientMedicalHistory">
              <ItemData
ItemOID="frmDem.PatientMedicalHistory.MedicalHistory.MedicalHistory" Value="A
brief summary of the medical history here." pf:Alias="vstBase.frmDem" />
              <pf:InFormItemData
InFormItemOID="frmDem.PatientMedicalHistory.MedicalHistory"
Name="MedicalHistory">
```

```
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
            </ItemGroupData>
            <ItemGroupData ItemGroupOID="frmDem.sctFamilyHistoy">
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.MaritalStatus.MaritalStatus.NeverMarried"
IsNull="Yes" pf:Alias="vstBase.frmDem" />
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.MaritalStatus.MaritalStatus.Married"
Value="Married" pf:Alias="vstBase.frmDem" />
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.MaritalStatus.MaritalStatus.Separated"
IsNull="Yes" pf:Alias="vstBase.frmDem" />
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.MaritalStatus.MaritalStatus.Divorced"
IsNull="Yes" pf:Alias="vstBase.frmDem" />
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.MaritalStatus.MaritalStatus.Widowed"
IsNull="Yes" pf:Alias="vstBase.frmDem" />
              <ItemData ItemOID="frmDem.sctFamilyHistoy.Children.Children"
Value="1" pf:Alias="vstBase.frmDem" />
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression.FamilyHistoryofDepress
ion.1" Value="1" pf:Alias="vstBase.frmDem">
                <Annotation SeqNum="1" pf:DBUID="20567" pf:GUID="{06673504-7BD4-
4505-A87C-A52406327D2F}">
                  <Comment>test comment</Comment>
                </Annotation>
              </ItemData>
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression.FamilyHistoryofDepress
ion.2" Value="2" pf:Alias="vstBase.frmDem">
                <Annotation SeqNum="1" pf:DBUID="20567" pf:GUID="{06673504-7BD4-
4505-A87C-A52406327D2F}">
                  <Comment>test comment</Comment>
                </Annotation>
              </ItemData>
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression.FamilyHistoryofDepress
ion.3" Value="3" pf:Alias="vstBase.frmDem">
                <Annotation SeqNum="1" pf:DBUID="20567" pf:GUID="{06673504-7BD4-
4505-A87C-A52406327D2F}">
                  <Comment>test comment</Comment>
                </Annotation>
              </ItemData>
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression.FamilyHistoryofDepress
ion.1.FamilyHistoryYes.Grandparent" IsNull="Yes" pf:Alias="vstBase.frmDem">
                <Annotation SeqNum="1" pf:DBUID="20567" pf:GUID="{06673504-7BD4-
4505-A87C-A52406327D2F}">
                  <Comment>test comment</Comment>
                </Annotation>
              </ItemData>
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression.FamilyHistoryofDepress
ion.1.FamilyHistoryYes.Parent" IsNull="Yes" pf:Alias="vstBase.frmDem">
                <Annotation SeqNum="1" pf:DBUID="20567" pf:GUID="{06673504-7BD4-
4505-A87C-A52406327D2F}">
                  <Comment>test comment</Comment>
                </Annotation>
              </ItemData>
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression.FamilyHistoryofDepress
ion.1.FamilyHistoryYes.Parent_Sibling" Value="Parent_Sibling"
pf:Alias="vstBase.frmDem">
                <Annotation SeqNum="1" pf:DBUID="20567" pf:GUID="{06673504-7BD4-
4505-A87C-A52406327D2F}">
                  <Comment>test comment</Comment>
                </Annotation>
              </ItemData>
              <ItemData
```

```
ItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression.FamilyHistoryofDepress
ion.1.FamilyHistoryYes.Sibling" Value="Sibling" pf:Alias="vstBase.frmDem">
                <Annotation SeqNum="1" pf:DBUID="20567" pf:GUID="{06673504-7BD4-
4505-A87C-A52406327D2F}">
                  <Comment>test comment</Comment>
                </Annotation>
              </ItemData>
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression.FamilyHistoryofDepress
ion.1.FamilyHistoryYes.Other" Value="Other" pf:Alias="vstBase.frmDem">
                <Annotation SeqNum="1" pf:DBUID="20567" pf:GUID="{06673504-7BD4-
4505-A87C-A52406327D2F}">
                  <Comment>test comment</Comment>
                </Annotation>
              </ItemData>
              <ItemData
ItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression.FamilyHistoryofDepress
ion.1.FamilyHistoryYes.Other.OtherSpecify" Value="What is this"
pf:Alias="vstBase.frmDem">
                <Annotation SeqNum="1" pf:DBUID="20567" pf:GUID="{06673504-7BD4-
4505-A87C-A52406327D2F}">
                  <Comment>test comment</Comment>
                </Annotation>
              </ItemData>
              <pf:InFormItemData
InFormItemOID="frmDem.sctFamilyHistoy.MaritalStatus" Name="MaritalStatus">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData InFormItemOID="frmDem.sctFamilyHistoy.Children"
Name="Children">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData
InFormItemOID="frmDem.sctFamilyHistoy.FamilyHistoryofDepression"
Name="FamilyHistoryofDepression">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
            </ItemGroupData>
            <ItemGroupData ItemGroupOID="frmDem.EmploymentInfo">
              <ItemData ItemOID="frmDem.EmploymentInfo.Employed.Employed"
Value="1" pf:Alias="vstBase.frmDem" />
              <ItemData ItemOID="frmDem.EmploymentInfo.Occupation.Occupation"
Value="5" pf:Alias="vstBase.frmDem" />
              <ItemData ItemOID="frmDem.EmploymentInfo.AutoFileName"
IsNull="Yes" />
              <pf:InFormItemData InFormItemOID="frmDem.EmploymentInfo.Employed"
Name="Employed">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
              <pf:InFormItemData
InFormItemOID="frmDem.EmploymentInfo.Occupation" Name="Occupation">
                <pf:ItemStatus SourceVerified="No" Critical="StudyDefault" />
              </pf:InFormItemData>
            </ItemGroupData>
            <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes" />
            <pf:CustomData Name="CTMSElementName" Value="FormDemData" />
          </FormData>
```

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/pf:FormStatus/pf:CustomData
```

Example

```
          <FormData FormOID="frmDem">
            <pf:FormStatus Activated="Yes" Frozen="No" Locked="No" Deleted="No"
SVReady="No" SVPartial="No" SVComplete="No" SVSelected="Yes" Signed="No"
Completed="Yes">
              <pf:CustomData Name="CTMSElementName" Value="DemFormStatus" />
```

```
          </pf:FormStatus>
        </FormData>
```

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/pf:CustomData
```

Example

```
          <ItemGroupData
ItemGroupOID="frmDOSE.sctDosingRecordEntry.sctDosingRecordEntry"
ItemGroupRepeatKey="404831346876015">
            <pf:ItemGroupStatus SVComplete="Yes" Deleted="No" />
            <pf:CustomData Name="CTMSElementName" Value="ItemsetStatus" />

          </ItemGroupData>
```

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/pf:C
ustomData\
```

Example

```
            <ItemData
ItemOID="frmDOV2.sctFormstobecompleted.Checkallforms.Checkallforms.ECG"
IsNull="Yes" pf:Alias="CheckBoxGroup">
              <pf:CustomData Name="CTMSElementName" Value="ItemData" />
            </ItemData>
```

Association

This element exists in the transactional output. The custom event extract populates it when there is an association between the two forms.

IsNull

This element indicates whether or not the item or form is null. The value is Yes when the form or item does not have any data. Otherwise, the element is not generated.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/ItemData/
@IsNull
```

IsNull example

```
<ItemData ItemOID="frmPE.sctPhysicalExamination_PE.Respiratory" IsNull="Yes" />
<FormData FormOID="frmVS" pf:IsNull="Yes" />
```

Elements for clinical data

FormattedDateValue

The FormattedDateValue element returns data that differentiates between null and unknown date values in the InForm software.

XPath

```
ODM/ClinicalData/ItemData/pf:FormattedDateValue
```

- UNK indicates an unknown date part.
- NUL indicates a null date part.

To maintain backward compatibility and conform to ODM standard, "-" is still used to express a value of unknown or null in the Value attribute. If differentiation between null and unknown is required, use the pf:FormattedDateValue extension.

For more information, see ***ItemData*** (on page 57).

FormStatus element

The FormStatus element refers to the state of a form in the InForm software. A FormStatus element is returned if a change has been made to a form's state; for example, if the form was frozen, locked, or marked SV ready.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/pf:FormStatus
```

Parent element	Child element
FormData	odm:AuditRecord.

Attributes

Attribute	Description
Activated	A flag that indicates whether the form has been activated. type=odm:YesOrNo. Optional.
SVPartial	A flag that indicates whether the form has been partially source verified. type=odm:YesOrNo. Optional.
Frozen	A flag that indicates whether the form is frozen. type=odm:YesOrNo. Optional.
Locked	A flag that indicates whether the form is locked. type=odm:YesOrNo. Optional.
Deleted	A flag that indicates whether the form is deleted. This attribute is returned only when complianceMode="Loose." type=odm:YesOrNo Optional.

Attribute	Description
SVComplete	A flag that indicates whether the form is source verification complete. type=odm:YesOrNo. Optional.
SVReady	A flag that indicates whether the form is source verification ready. type=odm:YesOrNo Optional.
SVSelected	A flag that indicates whether partial source verification has been selected for the form. type=odm:YesOrNo Optional.
Signed	A flag that indicates whether the transaction caused the signature status of the form to change. type=odm:YesOrNo **Yes**—the transaction caused the signature status of the form to change to "signed." **No**—the transaction caused the signature status of the form to change to "unsigned."

ItemStatus element

The ItemStatus element refers to the status of an item on a form. This element is returned if an item has been source verified or if an itemset has been deleted or undeleted.

XPath

If an item has been source verified:

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/pf:ItemSta
tus
```

If an itemset has been deleted or undeleted:

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/InformItem
Data/pf:ItemStatus
```

Parent elements	Child element
Either of the following: • InFormItemData (if the item has been source verified). • ItemGroupData (if the itemset has been deleted).	odm:AuditRecord.

Attributes

Attribute	Description
SourceVerified	A flag that indicates whether the item has been source verified. type=odm:YesOrNo. Optional.
Deleted	A flag that indicates whether the itemset has been deleted. type=odm:YesOrNo. Optional.

Query element

The query element represents a query in the InForm software.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/ItemGroupData/pf:InFormItem
Data/pf:Query
```

Parent element	Child element
InFormItemData.	***QueryStatus*** (on page 49).

Attributes

Attributes	Description
DBUID	The location OID of the query.
OID	The identifier of the query. type=odm:oid. Optional
Text	The text of the query. type=xs:string. Optional
Type	The type of query. type=QueryType. Required.

QueryStatus element

The QueryStatus element represents the state of a query. The element contains an audit record that explains when and why the change of state happened.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/formData/ItemGroupData/InFormItemData/pf:Query/pf:QueryStatus
```

Parent element	Child element
pf:Query	odm:AuditRecord.

Attributes

Attribute	Description
Status	type=QueryState. Required. Values: Open, Answered, closed, Candidate, Reissued, or Deleted.
Text	type=String.

ReasonIncomplete element

The ReasonIncomplete element represents the reason that an item in the InForm software does not have a value.

XPath

```
ODM/ClinicalData/SubjectData/StudyEventData/FormData/Annotation/pf:ReasonIncomplete
```

Parent element	Child element
Annotation	CodeListOID="ReasonIncomplete."

ReasonIncomplete is a hard-coded codelist that defines the options. The code list is returned in the metadata and ItemData.

SubjectStatus element

The SubjectStatus element indicates that a state change has occurred to a patient, subject, or CRB.

XPath

```
ODM/ClinicalData/SubjectData/pf:SubjectStatus
```

Parent element	Child element
SubjectData	odm:AuditRecord.

Attributes

Attribute	Description
Frozen	A flag that indicates whether the subject is frozen. type=odm:YesOrNo. Optional.
Locked	A flag that indicates whether the item is locked. type=odm:YesOrNo. Optional.
SVReady	A flag that indicates whether the item is source verification ready. type=odm:YesOrNo. Optional.
State	The state of the subject. Values are: Screened, ScreenFailed, Enrolled, EnrollmentFailed, EnrollmentOverride, Randomized, Complete, or DroppedOut. type=SubjectState. Optional.
ScreenFailReason	The reason that the subject failed the screening. type=xs:string. Optional.
EnrollFailReason	The reason that the subject failed to enroll. type=xs:string. Optional.
EnrollOverrideReason	The reason that the enrollment was overridden. type:xs:string. Optional.

MetaDataVersionRef

Attribute	Value/Map to InForm
StudyOID	Study Name. Entity: Site. Property: Study.

Attribute	Value/Map to InForm
MetaDataVersionOID	Volume Edition Description. Entity: Site. Property: Study Version. If the form has no study version, MetaDataVersionOID="Undefined."
EffectiveDate	Site accept date. Entity: Site. Property: Study Version Approval Date.

Example—MetaDataVersionRef

```
<MetaDataVersionRef StudyOID="Hypertension Study"
   MetaDataVersionOID="Study Version Basic" EffectiveDate="1998-06-30" />
```

ClinicalData

Attribute	Value/Map to InForm
StudyOID	Study Name.
MetaDataVersionOID	Study Version. If the form has no study version, MetaDataVersionOID="Undefined."

Example—ClinicalData

```
<ClinicalData StudyOID="Depression Study Update" MetaDataVersionOID="Study
Version 1.0" />
```

SubjectData

Attribute	Value/Map to InForm
SubjectKey	Patient DBUID.
GUID	Subject identifier.
Revision	Entity: Subject. Property: Revision.
SiteRef	Entity: Patient Move. Property: *no property listed.*

Signature

Indicates that the CRB has been signed for the subject.

SiteRef

Site for the subject. In the event of a subject site move, a SubjectData element that contains a new SiteRef and no StudyEventData element is sent.

pf:SubjectPool

Represents the source verification pool in the InForm application.

`SubjectData/pf:SubjectPool`

Attributes	Value/Map to InForm
PoolNumber	The InForm Pool Number. type=xs:string. Required. One of the following values, representing the available pools that are hardcoded in the InForm software: 0—PF_SVELIGIBLE_POOL 1—PF_SVAUTO_SELECT_POOL 2—PF_SVMANUAL_FIRSTN_POOL 3—PF_SVMANUAL_UISELECT_POOL 4—PF_SVMANUAL_IMPORT_POOL 99—PF_SVEXCLUDED_POOL
PoolRefName	The ref name of the pool. type=xs:string. Required. One of the following values: **Eligible**—PoolNumber=0, indicating the pool is SV Eligible. **AutoSelect**—PoolNumber=1, indicating a pool whose members are automatically selected based on settings determined by the site administrator. **ManualFirstN**—PoolNumber=2, indicating a pool in which the first <n> number of subjects are automatically selected. **ManualUISelect—**PoolNumber=3, indicating that the pool members are added by being selected in the UI. **ManualImport—**PoolNumber=4, indicating an imported file that contains the pool members. **Excluded**—PoolNumber=99, indicating that the pool is excluded from source verification.

pf:SubjectPoolChange

Reflects the change in the subject pool. Includes the subjects placed in the pool.

`SubjectData/pf:SubjectPoolChange`

Attributes	Value/Map to InForm
LocationOID	Includes the subjects placed in the pool.
PoolNumber	Pool number.

pf:CriticalForm

Represents a form that must be source verified.

`SubjectData/pf:CriticalForm`

Attributes	Value/Map to InForm
FormRef	FORMGUID.
LocationOID	Site OID.
IsCritical	ISCRITICAL.

pf:ReviewState

Represents the state and stage of the progress of data review in the InForm application.

`SubjectData/pf:ReviewState`

Attributes	Value/Map to InForm
pf:DBUID	dbuid.
RefName	Review state name.
StateValue	State value.
Revision	Revision number.
Name element	Translated text.
Mnemonic element	Mnemonic.

pf:ReviewStage

Represents one of the three review stages of the review state.

`SubjectData/pf:CriticalForm`

Attributes	Value/Map to InForm
RefName	Stage ref name.

Attributes	Value/Map to InForm
StageValue	Stage value.
Activated	0 or 1.
Revision	Revision number.

Example—SubjectData

```
<ClinicalData StudyOID="StudyDesign" MetaDataVersionOID="Study Design 0.0.2"
pf:TransactionGuid="{6EAEA8C3-68D1-4314-881C-5AA2A5575308}"
pf:Revision="326048162773035.000">
     <SubjectData SubjectKey="14266" TransactionType="Upsert"
     pf:GUID="{83A9C428-8475-4AEF-A169-37D4F1E52975}">
     <SiteRef LocationOID="13341" />
     <StudyEventData StudyEventOID="enroll">
        <AuditRecord>
           <UserRef UserOID="enes" />
           <LocationRef LocationOID="Unknown" />
           <DateTimeStamp>2011-05-02T21:56:02+00:00</DateTimeStamp>
           <ReasonForChange>Visit Started</ReasonForChange>
        </AuditRecord>
        <FormData FormOID="frmEnr">
           <ItemGroupData ItemGroupOID="frmEnr.sctSubjectNum">
              <ItemData ItemOID="frmEnr.sctSubjectNum.SubNum_Enr.SubNum_Enr"
              Value="AAA">
                 <AuditRecord>
                    <UserRef UserOID="enes" />
                    <LocationRef LocationOID="13341" />
                    <DateTimeStamp>2011-05-02T21:56:02+00:00</DateTimeStamp>
                    <ReasonForChange>Enrollment Entry</ReasonForChange>
                    <SourceID>RDINFAPP261</SourceID>
                 </AuditRecord>
              </ItemData>
           </ItemGroupData>
        </FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
```

Candidate

`SubjectData/Candidate`

Attribute	Value/Map to InForm
DOB	The date of birth of the subject.
ScreeningDate	The date of the screening.
PatientGUID	The GUID of the subject.
Revision	The revision number of the subject.
SiteGUIDREF	The GUID of the site.
Initials	The initials of the subject.
State	The state of the subject.
ScreeningNumber	The subject's screening number.

Attribute	Value/Map to InForm
EnrollmentNumber	The subject's enrollment number.
ScreeningFailure	The reason for screening failure.
EnrollmentFailure	The reason for enrollment failure.
EnrollmentOverride	The reason that the enrollment failure was overridden.
PatientNumberRevision	The revision number of the subject number.

Example—Candidate

```
pf:Candidate SubjectInitials="GMC" EnrollmentNumber="08-002"
ScreeningNumber="SCR-08-1" ScreeningFailure="" EnrollmentFailure=""
EnrollmentOverride="" DateOfBirth="1959-04-18T00:00:00+00:00"
ScreeningDate="2004-01-01T00:00:00+00:00" PatientRevision="94995391031040.000">
   <AuditRecord>
      <UserRef UserOID="msilva" />
      <LocationRef LocationOID="Unknown" />
      <DateTimeStamp>2004-01-05T16:36:31+00:00</DateTimeStamp>
      <ReasonForChange>Screening Log Entry</ReasonForChange>
   </AuditRecord>
</pf:Candidate>
```

StudyEventData

An activated visit is indicated by a StudyEventData element that contains no child FormData element.

Attribute	Value/Map to InForm
StudyEventOID	Form-relative visit refpath.
StudyEventRepeatKey	Visit index (for repeating visits only).
TransactionType	• With Loose compliance mode: Upsert. • With Strict compliance mode: Insert, Upsert, or Remove.
DBUID	Entity: Visit. Property: DBUID.
Revision	Entity: Visit. Property: Revision.

Example—StudyEventData

```
<StudyEventData StudyEventOID="vstBASE" />
```

FormData

Attribute	Value/Map to InForm

Attribute	Value/Map to InForm
FormOID	Form RefName.
FormRepeatKey	Form index (for repeating forms only).

Signature

Indicates that the form has been signed for the subject.

Example—FormData

```
<FormData FormOID="LAE1" FormRepeatKey="243960146618003"
TransactionType="Remove">
   <pf:FormStatus Deleted="Yes">        Yes/No
      <AuditRecord>
         <UserRef UserOID="system" />
         <LocationRef LocationOID="Unknown" />
         <DateTimeStamp>2008-09-24T19:46:51+00:00</DateTimeStamp>
      </AuditRecord>
   </pf:FormStatus>
</FormData>
```

ItemGroupData

Attribute	Value/Map to InForm
ItemGroupOID	Form-relative RefPath for the section.
ItemGroupRepeatKey	Present when the section contains an ItemSet and specifies the ItemSet index.

Example—ItemGroupData

```
<ItemGroupData ItemGroupOID="LAB" ItemGroupRepeatKey="134562926897091" />
```

Annotation

```
FormData/Annotation
ItemData/Annotation
```

Attribute/Child	Value/Map to InForm
SeqNum	Comment ID.
Comment	Comment text.
Comment/ReasonIncomplete	Reason that the form or item is incomplete.
GUID	Entity: Comment. Property: GUID.
DBUID	Entity: Comment. Property: DBUID.

Example—Annotation

```
<Annotation SeqNum="9985">
   <Comment>test item comment</Comment>
</Annotation>
```

ItemData

Attribute/Child	Value/Map to InForm
ItemOID	Item RefName.
Value	ItemContext/control values.
DBUID	• Entity: Item. Property: DBUID. • Entity: Control. Property: DBUID.
Revision	• Entity: Item. Property: Revision. • Entity: Control Property: Revision.
pf:FormattedDateValue	Indicates a value of null or unknown in the date part. • UNK indicates an unknown date part. • NUL indicates a null date part. **Note:** To maintain backward compatibility and conform to ODM standard, "-" is still used to express a value of unknown or null in the Value attribute. If differentiation between null and unknown is required, use the pf:FormattedDateValue extension.

Example—ItemData

Example for unknown date part:

```
<ItemData ItemOID="LAE1.LsctAE.LitmAESTDT.LAESTDTTM"Value="2009-04-19T-:-:-
+00:00" pf:FormattedDateValue="2009-04-19TUNK:UNK:UNK+00:00" />
```

Example for null date part:

In the example, the hour, minute, and second are null.

```
<ItemData ItemOID="LAE1.LsctAE.LitmAESTDT.LAESTDTTM"Value="2009-04-19T-:-:-
+00:00" pf:FormattedDateValue="2009-04-19TNUL:NUL:NUL+00:00" />
```

ItemStatus

If an item has been source verified:

`InformItemData/ItemStatus`

If an itemset has been deleted or undeleted:

`ItemGroupData/ItemStatus`

Attribute/Child	Value/Map to InForm
Source Verified	Item has been source verified.
Deleted	Indicates whether the item has been deleted.

Example—ItemStatus

```
<FormData FormOID="frmDOV">
   <ItemGroupData ItemGroupOID="frmDOV.DOV">
      <pf:InFormItemData InFormItemOID="frmDOV.DOV.DOV" Name="DOV">
         <pf:ItemStatus SourceVerified="Yes"> "> (Yes or No are the 2 values this
         can be)
            <AuditRecord>
               <UserRef UserOID="system" />
               <LocationRef LocationOID="Unknown" />
               <DateTimeStamp>2008-09-17T20:32:49+00:00</DateTimeStamp>
            </AuditRecord>
         </pf:ItemStatus>
      </pf:InFormItemData>
   </ItemGroupData>
</FormData>
```

InFormItemData

`InformItemData/ItemStatus`

Attribute/Child	Value/Map to InForm
InformItemOID	Form-relative RefPath for the item.
Name	Item RefName.
ItemStatus	Status of the item in the InForm software; for example, SVReady.
Query	Query element against an item in the InForm software.

Example—InformItemData

```
<pf:InFormItemData InFormItemOID="frmDOSE.sctDOSE.itsDOSE.itmDOSEToDate"
Name="itmDOSEToDate">
   <pf:Query OID="{00E6AD12-0848-4F71-AB61-ED0F845D1404}" Text="Stop Date is
   not equal to or after the Start Date; please verify" Type="Autoquery"
   pf:DBUID="75369" pf:Revision="244386598500043.000">
      <pf:QueryStatus Status="Open">
         <AuditRecord>
            <UserRef UserOID="autoquery" />
```

```
            <LocationRef LocationOID="Unknown" />
            <DateTimeStamp>2008-09-29T18:09:58+00:00</DateTimeStamp>
            <ReasonForChange>Stop Date is not equal to or after the Start
            Date; please verify</ReasonForChange>
         </AuditRecord>
      </pf:QueryStatus>
   </pf:Query>
</pf:InFormItemData>
```

Query

`InformItemData/Query`

Attribute/Child	Value/Map to InForm
DBUID	Entity: Query. Property: DBUID.
OID	Query identifier.
Text	The text of the query.
Type	Autoquery or User.
Revision	Revision.
Query/QueryStatus	The status of the query.

Example—Query

```
<pf:Query OID="{08016623-5E3D-417C-93BB-33573CF1E1DC}" Text="Data does not match
source" Type="User">
   <pf:QueryStatus Status="Open">
      <AuditRecord>
         <UserRef UserOID="system" />
         <LocationRef LocationOID="Unknown" />
         <DateTimeStamp>2008-03-17T19:17:22+00:00</DateTimeStamp>
         <ReasonForChange>Data does not match source</ReasonForChange>
      </AuditRecord>
   </pf:QueryStatus>
</pf:Query>
```

Elements for admin data

GroupDef element

The GroupDef element defines a user group in an InForm study. AdminData contains a collection of GroupDef elements.

Xpath

`ODM/AdminData/pf:GroupDef`

Parent element	Child element
AdminData	pf:Right.

Attributes

Attribute	Description
OID	The refname of the item. type=odm:oid. Required.
Name	The name of the group as displayed in the InForm software. type=odm:text. Required.
GroupType	The type of the group, such as Rights, Reporting, Query, Signature. type=xs:String. Required.

Right element

The pf:Right element is a child element of the pf:GroupDef element. One pf:Right element exists for each right defined in the user group.

Xpath

```
ODM/AdminData/pf:GroupDef/pf:Right
```

Parent element	Child element
pf:GroupDef	None.

Attributes

Attribute	Description
Description	The description of the right. type=xs:string. Required.

User

Attribute/Child	Value/Map to InForm
OID	User Name.
UserType	User Type.
LoginName	User Name.
DisplayName	User DisplayName.

Attribute/Child	Value/Map to InForm
FullName	User first name + User last name. **Note:** Not all cultures use FirstName + LastName as the full name. The full name can be derived from FirstName and LastName elements that are available in the same response. The consumer of DownloadAdminDataODM should rely on FirstName and LastName elements to get the full name or display name.
FirstName	User first name.
LastName	User last name.
Organization	
Address/StreetName	User Address 1. User Address 2.
Address/City	User City.
Address/StateProv	User State Province.
Address/County	User Country.
Address/PostalCode	User postal code.
Address/OtherText	
Email	User email.
Picture	
Pager	User beeper.
Fax	User fax.
Phone	User phone number.
LocationRef/LocationOID	Site Name. Entity: User. Property: Site.
GroupRef	One GroupRef child element exists for each group to which the user belongs. Contains a GroupOID attribute.
DBUID	Entity: User. Property: DBUID.
GUID	User Identifier.
Revision	Entity: User. Property: Revision.

Attribute/Child	Value/Map to InForm
LoginName	Entity: User. Property: Login Name.
UserType	Entity: User. Property: User Type.
StudyLocale	Locale of study. Derived from dcv_thingdata in the study database.
Active	Indicates whether the user state is active.
Terminated	Indicates whether the user state is terminated.

Example—User

```
<User OID="mmeyer" UserType="Investigator" pf:DBUID="11816" pf:GUID="{09379D50-
EB2B-46B1-8BDC-F2F7F6A6363D}" pf:StudyLocale="en-US" pf:Active="Yes"
pf:Terminated="No">
           <LoginName>mmeyer</LoginName>
           <DisplayName>Marianne Meyer</DisplayName>
           <FullName>Marianne Meyer</FullName>
           <FirstName>Marianne</FirstName>
           <LastName>Meyer</LastName>
           <Address>
             <Country>USA</Country>
           </Address>
           <Email>email@domain.test</Email>
           <Fax>(555) 555-1212</Fax>
           <Phone>(555) 555-1212</Phone>
           <LocationRef LocationOID="13341" />
           <pf:GroupRef GroupOID="6536" />
           <pf:GroupRef GroupOID="13940" />
      </User>
```

GroupDef

GroupRef elements reference GroupDef elements by matching the GroupRef GroupOID attribute to the GroupDef OID attribute.

GroupRef is a child of ***User*** (on page 60).

Attribute/Child	Value/Map to InForm
OID	The DBUID of the group.
NAME	Name of group as displayed in the InForm software.
GroupType	Type of group as defined in the InForm software.
Right	List of rights. Contains a list of Right elements, each of which contains a Description.

Right

`AdminData/pf:GroupDef/pf:Right`

Attribute	Value/Map to InForm
Description	The right defined in the group.

Example—GroupDef

```
<pf:GroupDef OID="5954" Name="System Creator Group" GroupType="Rights">
   <pf:Right Description="Create User"/>
   <pf:Right Description="Terminate User" />
</pf:GroupDef>
```

Location

To ensure that the audit records that are generated by sponsor users comply with the ODM standard, a Location with OID **Unknown** is automatically defined in the ODM that is returned for a study.

Attribute	Value/Map to InForm
OID	Site DBUID.
Name	Site name.
LocationType	Site. Entity: Site. Property: Type.
Revision	Entity: Site. Property: Revision. **Note:** Location Revision is the same as transaction only if the transaction is revising the location.
GUID	Site identifier.
SiteCountry	Entity: Site. Property: Country.
SiteTimezone	Entity: Site. Property: Timezone.
Locale	Locale of the site.
pf:SVAutoSelectRate	The value comes from a query: For more information, see ***Subject pool selection*** (on page 52).
pf:SVFirstNSubjects	The value comes from a query: For more information, see ***Subject pool selection*** (on page 52).

Mnemonic

`Location/Mnemonic`

Attribute	Value/Map to InForm
TranslatedText	The site mnemonic that does not come from the pf_resourcedata. This mnemonic can have any Unicode characters. It should not contain the xml:lang attribute.

Example—Location

```
<Location
   LocationType="Site" Name="(01) Massachusetts General Hospital"
   OID="10127" pf:SiteCountry="USA"
      pf:SiteTimezone="(GMT-05:00) Eastern Time (US &amp;amp; Canada)">
   <MetaDataVersionRef
      EffectiveDate="2005-04-01T00:00:00+00:00"
      MetaDataVersionOID="StudyVersion020108"
      StudyOID="Depression Study"/>
   <Mnemonic
      xmlns="http://www.phaseforward.com/InFormAdapter/ODM/Extensions/2.0">
   <TranslatedText
      xmlns="http://www.cdisc.org/ns/odm/v1.3">01</TranslatedText>
   </Mnemonic>
</Location>
```

SignatureDef

Attribute	Value/Map to InForm
OID	Signature DBUID.
Methodology	Electronic.
Meaning	CRB meaning, CRF meaning, or Group Name.
LegalReason	CRB Affidavit, CRF Affidavit, or PF_DEFAULT_CRFAFFADAVIT.

Meaning

CRB meaning, CRF meaning, or Group Name.

`SignatureDef/Meaning`

Attribute	Value/Map to InForm
TranslatedText	Language-specific text from the study.

LegalReason

CRB Affidavit, CRF Affidavit, or PF_DEFAULT_CRFAFFADAVIT.

`SignatureDef/Meaning`

Attribute	Value/Map to InForm
TranslatedText	Language-specific text from the study.

Example—SignatureDef

```
<SignatureDef OID="CRA Sigs" Methodology="Electronic">
   <Meaning>Approved</Meaning>
   <LegalReason>
      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. 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. To this I do attest by supplying my password and
      clicking the button marked Submit below.
   </LegalReason>
</SignatureDef>
```

Elements for metadata

Mapping of localized items

The following items are localized in the InForm software. The translation for each item is placed under the TranslatedText element in the ODM that is returned.

Item name	Element name in ODM.xml
The symbol of measurement unit.	MeasurementUnit/Symbol
Item text.	itemDef/Question
Radio button list, dropdown list, list box, control group, or checkbox list that is associated with a control in the InForm software.	CodeList/CodeListItem/Decode
Form mnemonic.	FormDef/pf:Mnemonic
Visit mnemonic.	StudyEventDef/pf:Mnemonic
Legal Reason.	SignatureDef/pf:LegalReason
Meaning.	SignatureDef/pf:Meaning

Site mnemonic is also placed under the TranslatedText element. However, it is not localized in pf_resourcedata in the InForm application. It is returned as the fallback because there is no language associated with it.

InFormItemDef element

The InFormItemDef element is the metadata definition for the Inform item, and explains which controls (ItemRefs) are contained in the item.

In the Oracle implementation, an ItemDef maps to an InForm control.

XPath

`ODM/Study/MetaDataVersion/pf:InformItemDef`

Parent element	Child element
MetaDataVersion	odm:ItemRef.

Attributes

Attribute	Description
OID	The refname of the item. type=odm:oid. Required.
Name	type=odm:text. Required.
SVRequired	Indicates whether source verification is required. type=odm:YesOrNo. Optional.
DBUID	The location OID.
GUID	Visit identifier.
Revision	Revision.
UUID	UUID.

InFormItemRef element

The InFormItemRef element shows the ordering of the InForm item within a section.

XPath

`ODM/Study/MetaDataVersion/ItemGroupDef/pf:InFormItemRef`

Parent element	Child element
ItemGroupDef	None.

Attributes

Attribute	Description

Attribute	Description
InFormItemOid	The refname of the item. type=odm:oid. Required.
OrderNumber	Display order of the item within the section. type=odm:integer. Optional.
Mandatory	Indicates whether the item is mandatory. type=odm:YesOrNo. Required.

Study

Attribute	Value/Map to InForm
OID	InForm Study Name.

Example—Study

```
<Study OID="Hypertension Study" />
```

GlobalVariables

Child	Value/Map to InForm
StudyName	InForm Study Name.
StudyDescription	Volume Edition DesignNotes.
ProtocolName	ProtocolVersion.

Example—GlobalVariables

```
<GlobalVariables>
   <StudyName>Hypertension Study</StudyName>
   <ProtocolName>Protocol XYZZY</ProtocolName>
</GlobalVariables>
```

BasicDefinitions

BasicDefinitions contains the InForm measurement unit definitions.

MeasurementUnit

Attribute	Value/Map to InForm
OID	Unit RefName.
Name	Unit Symbol.
DBUID	Entity: Units. Property: DBUID.
GUID	Unit identifier.
Revision	Entity: Units. Property: Revision.

TranslatedText

`BasicDefinitions/MeasurementUnit/Symbol/TranslatedText`

Attribute	Value/Map to InForm
Body text	Unit Symbol.

TranslatedText specifies the xml:lang for each language, and the element without xml:lang. This value is used when the culture is unspecified or unavailable for a given value.

Example—BasicDefinitions

```
<BasicDefinitions>
   <MeasurementUnit OID="Inches" Name="Inches">
      <Symbol>
         <TranslatedText>in</TranslatedText>
         <TranslatedText xml:lang="en-US">in</TranslatedText>
         <TranslatedText xml:lang="ja-JP">in</TranslatedText>
      </Symbol>
   </MeasurementUnit>
   <MeasurementUnit OID="Centimeter" Name="Centimeter">
      <Symbol>
         <TranslatedText>cm</TranslatedText>
      </Symbol>
   </MeasurementUnit>
</BasicDefinitions>
```

MetaDataVersion

Attribute	Value/Map to InForm
OID	Study version RefName.
Name	Study version display name.
ApprovalDate	Date that the study was approved.

Attribute	Value/Map to InForm
DBUID	Entity:StudyVersion. Property: Revision.
GUID	Study version identifier.
Revision	Entity:StudyVersion. Property: Revision.

StudyEventRef

Attribute	Value/Map to InForm
StudyEventOID	Visit RefName.
OrderNumber	Visit order number (for user interface display).
Mandatory	Always **No**.

Example—MetaDataVersion

```
<MetaDataVersion OID="Study Version Basic" Name="Protocol XYZZY">
   <Protocol>
      <StudyEventRef StudyEventOID="Visit1" OrderNumber="1" Mandatory="No" />
      <StudyEventRef StudyEventOID="Visit2" OrderNumber="2" Mandatory="No" />
      <StudyEventRef StudyEventOID="Visit3" OrderNumber="3" Mandatory="No" />
      <StudyEventRef StudyEventOID="Visit4" OrderNumber="4" Mandatory="No" />
   </Protocol>
</MetaDataVersion>
```

StudyEventDef

Attribute	Value/Map to InForm
OID	Visit RefName.
Name	Visit RefName.
Repeating	**Yes** for repeating visits.
Type	Scheduled, Unscheduled, or Common.
Category	PatientVisit, Screening, Enrollment, CommonCRF, or Relation.
DBUID	Entity: Visit. Property: DBUID.
GUID	Visit identifier.
Revision	Entity: Visit. Property: Revision.
UUID	Entity: Visit. Property: UUID.

Attribute	Value/Map to InForm
Dynamic	Visit is a dynamic visit.
VisitStartHours	Visit start hours set in Central Designer. For example, if the value for visit 2 is 8, visit 2 should occur one day after visit 1.

Mnemonic

`StudyEventDef/Mnemonic`

Attribute	Value/Map to InForm
TranslatedText	Display mnemonic for visit.

FormRef

Attribute	Value/Map to InForm
FormOID	Form RefName.
OrderNumber	Form page order (for user interface display).
Mandatory	Always **No**.
AltFormOID	Specifies the OID of the Alt form for the form that is identified by the current FormOID.

Example—StudyEventDef

```
<StudyEventDef OID="vstFinal" Name="vstFinal" Repeating="No" Type="Scheduled"
Category="Enrollment" pf:DBUID="10057" pf:GUID="{8BA4EB52-CCFD-4567-92CD-
AFFA16947926}" pf:Revision="250599557260035" pf:UUID="D882CE3A-0F42-11D2-A419-
00A0C963E0AC">
   <FormRef FormOID="frmDOV" OrderNumber="1" Mandatory="No" />
   <FormRef FormOID="frmSC" OrderNumber="2" Mandatory="No" />
      <pf:Mnemonic>
         <TranslatedText>Final</TranslatedText>
      </pf:Mnemonic>
</StudyEventDef>
```

FormDef

Attribute	Value/Map to InForm
FormOID	Form RefName.
Name	Form RefName.
Repeating	**Yes** if the form is a repeating form.
DBUID	Entity: Form. Property: DBUID.

Attribute	Value/Map to InForm
GUID	Form identifier.
Revision	Entity: Form. Property: Revision.
UUID	Entity: Form. Property: UUID.
Dynamic	**Yes** if the form is a dynamic form.
OrderNumber	The order of the form (for presentation in the user interface).

Mnemonic

`FormDef/Mnemonic`

Attribute	Value/Map to InForm
TranslatedText	The display mnemonic for the form.

ItemGroupRef

Attribute	Value/Map to InForm
ItemGroupOID	Form-relative RefPath for the section.
OrderNumber	Section Order.
Repeating	Section Repeating, when **Yes** indicates that the section contains an itemset.

Example—FormDef

```
<FormDef OID="frmDOV" Name="frmDOV" Repeating="No" pf:DBUID="7642"
pf:GUID="ab9274c1-b46c-4d3d-baf6-a0a8bbcf6a06" pf:Revision="250599336073035"
pf:UUID="D882CE3B-0F42-11D2-A419-00A0C963E0AC">
   <ItemGroupRef ItemGroupOID="frmDOV.sctVisitDateandTime" OrderNumber="1"
   Mandatory="No" />
   <pf:Mnemonic>
      <TranslatedText>DOV</TranslatedText>
   </pf:Mnemonic>
</FormDef>
```

ItemGroupDef

Attribute	Value/Map to InForm
OID	Form-relative RefPath for the section.
Name	Section RefName.
Repeating	Section Repeating; **Yes** when the section contains an ItemSet.

Attribute	Value/Map to InForm
ItemSetDBUID	Entity: ItemSet. Property: DBUID.
DBUID	Entity: Section. Property: DBUID.
GUID	Form identifier.
Revision	Entity: Section. Property: Revision.
UUID	Entity: Section/Itemset. Property: UUID.
ItemSetRefName	Present when the section contains an itemset; specifies the RefName of the ItemSet in the section.
ItemSetRevision	Entity: ItemSet. Property: Revision.
SVRequired	Present when the section contains an ItemSet and the ItemSet requires source verification.

InFormItemRef

`InformItemData/ItemStatus`

Attribute	Value/Map to InForm
InFormOID	Form-relative RefPath of an item.
Mandatory	**Yes,** if the item is mandatory.
OrderNumber	Display order of the item within the section.

Example—InformItemRef

```
<pf:InFormItemRef InFormItemOId="frmCGI.sctCGIImp.itmCGIImp" OrderNumber="1"
Mandatory="Yes" />
```

ItemRef

Attribute	Value/Map to InForm
ItemOID	Form-relative RefPath for a control within an item.
OrderNumber	Control order within an item.
Mandatory	Control is required.

Example—ItemGroupDef

```
<ItemGroupDef OID="frmDOV.sctVisitDateandTime" Name="sctVisitDateandTime"
Repeating="No" pf:DBUID="7639" pf:GUID="{0bf10cac-c53e-4926-8843-9e8fda1569d4}"
pf:Revision="250599335838035" pf:UUID="ABCFA388-223A-11D2-A426-00A0C963E0AC">
   <ItemRef
   ItemOID="frmDOV.sctVisitDateandTime.DateandTimeofVisit_DOV.DateandTimeofVisit
   _DOV" OrderNumber="1" Mandatory="Yes" />
   <pf:InFormItemRef
   InFormItemOId="frmDOV.sctVisitDateandTime.DateandTimeofVisit_DOV"
   OrderNumber="1" Mandatory="Yes" />
</ItemGroupDef>
```

InFormItemDef

Attribute/Child	Value/Map to InForm
OID	The refname of the item. type=odm:oid. Required.
Name	Form-relative RefPath for the item. type=odm:text. Required.
SVRequired	**Yes,** if source verification is required for the item. type=odm:YesOrNo. Optional.
DBUID	Entity: Item. Property: DBUID.
GUID	Entity: Item. Property: GUID.
Revision	Entity: Item. Property: Revision.
UUID	Entity: Item. Property: UUID.

Example—InformItemDef

```
<pf:InFormItemDef OID="frmDOV.sctVisitDateandTime.DateandTimeofVisit_DOV"
Name="DateandTimeofVisit_DOV" SVRequired="Yes" pf:DBUID="7648"
pf:GUID="{005A7BA1-AEA9-42E4-83F3-3E2358F68646}" pf:Revision="250599336526035"
pf:UUID="BD991BBF-B0A4-11D2-80E3-00A0C9AF7674">
   <ItemRef
   ItemOID="frmDOV.sctVisitDateandTime.DateandTimeofVisit_DOV.DateandTimeofVisit
   _DOV" OrderNumber="1" Mandatory="Yes" />
</pf:InFormItemDef>
```

ItemDef

Attribute	Value/Map to InForm
OID	Form-relative RefPath for the control.
Name	Item RefName.
DataType	See ***Data Type Table*** (on page 75).
Length	See ***Data Type Table*** (on page 75).
SignificantDigits	See ***Data Type Table*** (on page 75).
Revision	• Entity: Control. Property: Revision. • Entity: Item. Property: Revision.
GUID	Item identifier.
DBUID	• Entity: Control. Property: DBUID. • Entity: Item. Property: DBUID.
UUID	Entity: Control. Property: UUID.
ParentOID	The parent of each ItemDef that is not a top-level ItemDef.
ParentType	Defines the type of parent control. If value of parent type is: • **ItemDef**, then pf:ParentOID refers to the parent ItemDef's OID. • **CodeListItem**, then pf:ParentOID refers to the parent CodeListItem's pf:SelectionOID.
SelectionOID	An OID that uniquely identifies the selection in a radio group.
SelectionOwner	For each ItemDef that is the child of a radio group selection, the OID of the CodeList that contains the codelist selection.
ItemDefType	Indicates the control type. Values: RadioGroup, Checkbox, Text, Calculated, PullDown, Date.
Question	Question text for the item that contains the control.
SignificantDigits	See ***Data Type Table*** (on page 75).
CheckboxGroupRefName	Present when the control is contained in a checkbox group and contains the RefName of the checkbox group.

Data Type Table

InForm control type	Definition
Simple Control	• Data Type is defined by the element's type. • Length: text 255, integer 10, float 308. **Notes:** ▪ Integer is always returned as 10 (the maximum number of digits in a 32-bit integer). ▪ Float is always returned as 308 (the maximum exponent for a 64-bit floating-point number). • Significant Digit: 4.
Pulldown Control	References a codelist with the allowable values.
Text Control	• Data Type is defined by control data type (Pf_TextControl/DataType). • Length: text Pf_TextControl/MaxLength, integer 10, float 308. **Notes:** ▪ Integer is always returned as 10 (the maximum number of digits in a 32-bit integer). ▪ Float is always returned as 308 (the maximum exponent for a 64-bit floating-point number). • Significant Digit: Pf_TextControl/PFPrecision.
Radio Control	References a codelist with the allowable values.
Group Control	Represented as an ItemDef per check option with the name of the option included in the form-relative RefPath in the ItemDefOID.
Datetime Control	incompleteDatetime.
Calculated Control	Text type with length = 255.

MeasurementUnitRef

Attribute	Value/Map to InForm
MeasurementUnitOID	Unit RefName.
NormalizationUnit	**Yes,** if the referenced unit is for normal values.

Question

Child	Value/Map to InForm
TranslatedText	Item question. A TranslatedText element exists for each study language.

Example—ItemDef

```
<ItemDef OID="frmDOSE.sctDOSE.itsDOSE.itmDOSEFromDate.dtmDOSE" Name="dtmDOSE"
DataType="incompleteDatetime">
   <Question>
      <TranslatedText>Start date</TranslatedText>
   </Question>
</ItemDef>
```

CodeList

Attribute	Value/Map to InForm
OID	Control RefName. For example, if an item contains a drop-down list, the RefName of the drop-down list will be used. If file type is Transactional, the ConrolRefName is hierarchical: form.section.item.control
Name	• If file type is Snapshot, Name is the same as OID. • If file type is Transactional, Name is the same as the ControlRefName without the hierarchical prefixing.
DataType	Type of Control. See ***Data Type Table*** (on page 75).

Example—CodeList

File type is Snapshot:

```
<CodeList OID="rdcEXCLC" Name="rdcEXCLC" DataType="text"/>
```

File type is Transactional:

```
<CodeList OID="frmPreg.sctPreg.itmPregTest.itmPregTest.rdcPregDone"
Name="rdcPregDone" DataType="text">
```

CodeListItem

Attribute	Value/Map to InForm
CodedValue	Value used to map the CodeListItem to an ItemData value in ODM ClinicalData.

Attribute	Value/Map to InForm
pf:SelectionOID	An OID that uniquely identifies the selection in a radio group. The value is a concatenation of the CodeList OID and the CodeListItem CodedValue.

Example—CodeListItem

```
<CodeListItem CodedValue="2"
pf:SelectionOID="frmPreg.sctPreg.itmPregTest.itmPregTest.rdcPregDone.2">
   <Decode>
      <TranslatedText>Positive</TranslatedText>
   </Decode>
</CodeListItem>
```

Oracle attributes that are applied to standard ODM elements

The file **PFextensions.xsd** contains the attribute extensions for attributes that are defined in the **PhaseForward_ODM.xsd** file. These schema files are located in the **<Installation_path>/ODM/bin/xsd** directory.

Standard ODM element	Oracle attribute	Type
ClinicalData	TransactionalGuid	xs:string
CodeListItem	SelectionOID	odm:oid
FormDef	UUID	xs:string
FormRef	AltFormOID	xs:string
ItemData	NormalizedValue	odm:value
ItemDef	CheckboxGroupRefName	xs:string
ItemDef	InFormDateTime	xs:boolean
ItemDef	PFVarName	xs:string
ItemDef	ParentOID	odm:oidref
ItemDef	ParentType	pf:ParentType
ItemDef	ItemDefType	pf:ItemDefType
ItemDef	SelectionOwner	odm:oidref
ItemDef	UUID	xs:string
ItemGroupDef	ItemSetRefName	xs:string
ItemGroupDef	SVRequired	odm:YesOrNo
ItemGroupDef	UUID	xs:string
Location	Mnemonic	xs:string
Location	SiteCountry	xs:string
Location	SiteTimezone	xs:string

Standard ODM element	Oracle attribute	Type
Location	Locale	xs:language
Location	SVAutoSelectRate	xs:integer Value 1-100
Location	SVFirstNSubjects	xs:integer
MeasurementUnitRef	NormalizationUnit	odm:YesOrNo
MetaDataVersion	ApprovalDate	xs:boolean
ODM	MappingVersion	xs:string
ODM	HierarchicalOIDs	xs:string
ODM	InFormAdapterVersion	xs:string
SignatureDef	GroupName	xs:string
SignatureDef	Meaning	xs:string
SignatureDef	LegalReason	xs:string
FormDef StudyEventDef	Dynamic	odm:YesOrNo
StudyEventDef	Optional	odm:YesOrNo
StudyEventDef	UUID	xs:string
User	StudyLocale	xs:language

Revision, DBUID, and GUID attributes

DBUID, GUID, and Revision attributes are Oracle extension attributes on their respective elements.

Metadata

In metadata, the following elements include the pf:DBUID, pf:GUID, and pf:Revision attributes.

- MetaDataVersion
- StudyEventDef
- FormDef
- ItemGroupDef
- ItemDef
- MeasurementUnit
- InFormItemDef

In metadata, the following elements include the pf:UUID attribute.

- StudyEventDef
- FormDef
- ItemGroupDef
- ItemDef
- InFormItemDef

For more information, see the individual elements listed in ODM Mapping.

Clinical data and Association data

ODM data elements consist of Clinical data nodes and Association nodes. These nodes contain a new pf:Revision attribute that applies to all the data in the node unless the child node contains a pf:Revision attribute of its own.

The DBUID and GUID attributes for the data items within these nodes are defined in the metadata except as noted:

- ***Annotation*** (on page 56).
- ***Location*** (on page 63).
- ***Query*** (on page 59).
- ***SubjectData*** (on page 51).
- ***MetaDataVersion*** (on page 68).
- ***User*** (on page 60).
- ***Site\|Location*** (on page 63).

AdminData

The User element includes pf:DBUID, pf:GUID, pf:Revision, and pf:StudyLocale extensions.

Site\Location includes the pf:GUID, pf:Revision, and pf:Locale extensions.

- The location OID is the DBUID.
- Location Revision is the same as transaction only if the transaction is revising the location.

For more information, see the individual elements listed in ODM Mapping.

Extensions that show control hierarchy and type

To define details for controls, the ODM Export interface provides the following extensions to the ODM elements ***ItemDef*** (on page 74) and ***CodeListItem*** (on page 76).

Oracle extension to ItemDef	Description
pf:ParentOID	The parent of each ItemDef that is not a top-level ItemDef.
pf:ParentType	Defines the type of parent control. If value of parent type is: • **ItemDef**, then pf:ParentOID refers to the parent ItemDef's OID. • **CodeListItem**, then pf:ParentOID refers to the parent CodeListItem's pf:SelectionOID.
pf:SelectionOwner	For each ItemDef that is the child of a radio group selection, the OID of the CodeList that contains the codelist selection.
pf:ItemDefType	Indicates the control type. Values: RadioGroup, Checkbox, Text, Calculated, PullDown, Date.

Oracle extension to CodeListItem	Description
pf:SelectionOID	An OID that uniquely identifies the selection in a radio group. The value is a concatenation of the CodeList OID and the CodeListItem CodedValue.

CHAPTER 5

InForm to Siebel CTMS integration

In this chapter

Overview

The InForm to Siebel CTMS integration provides accurate and efficient tracking of subject status in Siebel CTMS by automatically sending subject, visit, and activity data from the InForm application to Siebel CTMS. The InForm to Siebel CTMS integration also provides accurate and timely information to avoid early or late payment or overpayment to sites.

Events of interest to Siebel CTMS include such things as:

- Sending subject data when the subject screening date is entered in the InForm application.
- Sending subject data when data is entered on InForm objects.
- Sending visit completion data when a visit is completed in the InForm reporting view.
- Sending visit completion data when data is entered on InForm objects.
- Sending visit completion data when a custom form status is attained in the InForm application.
- Sending activity completion data when data is entered on InForm objects.

Software required for InForm to Siebel CTMS integration

To export data from the InForm application to Siebel CTMS, you must use:

- InForm 6.0 or above.
- Siebel CTMS 8.1.1 (Innovation Pack 2014), patch set 1.

Defining data to export from InForm to Siebel CTMS

To send data from the InForm Publisher application to Siebel CTMS, you define custom events. You use triggers to define the condition for creating a subject, setting the status of a subject, or sending visit or activity completion information. You specify event result for each subject attribute and use the pf:Alias attribute to indicate where you want the value to go in Siebel CTMS. You use the custom data name and value pairs to send over subject or status information.

For information on defining data to export from the InForm application, see ***Defining data to export in the InForm application*** (on page 14).

InForm rules

- The value entered in the InForm item that maps to Subject Number in Siebel CTMS must not exceed 10 characters.
- The value entered in the InForm item that maps to Enrollment ID in Siebel CTMS must not exceed 30 characters.
- The value entered in the InForm item that maps to Randomization ID in Siebel CTMS must not exceed 30 characters.

Specifying aliases

You can associate custom data on a result at any level. The InForm Publisher application allows you to have multiple name and value pairs on any result.

For example:

- You can capture cost items, such as device part numbers at visit or activity level, to make payments for supplies used. To do this, you add a form field to the study to capture device data. When an event occurs, a custom pf:Alias value at the visit or activity level causes device information to be sent.
- You can send external identifiers, such as patient electronic medical record number, as part of subject enrollment data to the receiving application. To do this, you add a form field to capture the external identifier. When an event occurs, a custom pf:Alias value at the subject level causes the information to be sent.

The InForm to Siebel CTMS integration requires that you specify values as aliases for the results that must populate the appropriate fields in Siebel CTMS.

For example, to create a subject, you must send SubjectNumber, EncounterDate and ScheduleDate.

Use the following case-sensitive aliases for the InForm to Siebel CTMS mapping:

- ClinicalSubject
    - Comments
    - EncounterDate
    - EligibleFlag
    - EmailAddress

- EnrollmentId
- EarlyTerminatedDate
- EarlyTerminationReason
- FaxNumber
- PhoneNumber
- ProtocolDeviation
- ProtocolViolation
- RandomizedDate
- RandomId
- ScreenFailureDate
- ScreenFailureReason
- ScheduleDate
- SubjectNumber
- WithdrawnDate
- WithdrawnReason
- SubCustom<1..25>

- InFormedConsent
  - InFormedConsent
- ClinicalSubjectStatus
  - ScreenDate
  - EnrollmentDate
  - StatusDate
- VisitActivityCompletion
  - MissedFlag
  - CompletedDate
  - ActCustom<1..25>

Note: Siebel CTMS is a highly customizable application. You can send up to 25 items at the subject level and 25 at the visit level to Siebel CTMS by using the SubCustom<1..25> and ActCustom<1..25> aliases.

To receive this information on the Siebel CTMS side, you must add the fields to the integration object and the workflow used by the ClinicalSubject web service for updating the subject information.

For more information, see the Siebel CTMS documentation.

Custom fields

The InForm Publisher application checks for custom data defined on the event results and uses the event results to invoke the Siebel CTMS web service with the appropriate values.

For example:

- **ScreenedStatus, EnrolledStatus, Status**—Send status information to Siebel CTMS.
- **VisitClinicalItem, ActivityClinicalItem**—Send visit or activity completion information to Siebel CTMS.

Sending subject status to Siebel CTMS

The first time a subject is created in Siebel CTMS, it needs to have a status. When you define a custom event to create a subject in Siebel CTMS, you must also specify custom data for a status with which to create the subject.

- The name can be ScreenedStatus, EnrolledStatus or Status.
- The value in the name and value pair needs to be the Language Independent Code (LIC) value for the status in Siebel CTMS. If you use status tracking visits in Siebel CTMS, when you send visit completion status information to Siebel CTMS for a status tracking visit, the subject is automatically assigned that status.
- If you want a status of Screen Failure, you must send over the ScreenFailureDate and the ScreenFailureReason using the pf:Alias attribute on the event result.
- For Early Withdrawal Status, you must send over the WithdrawnDate and the WithdrawnReason using the pf:Alias attribute on the event result, and the subject is assigned that status in Siebel CTMS.
- For Early Termination Status, you must send over the EarlyTerminatedDate and the EarlyTerminatedReason using the pf:Alias attribute on the event result, and the subject is assigned that status in Siebel CTMS.

Sending visit or activity-completion status to Siebel CTMS

You must define an event to send Visit completion or Activity completion status to Siebel CTMS. You define the trigger based on your requirements for completion for the visit or activity.

- In the event results, specify a date to be used as the visit completion date. You must define the pf:Alias as CompletedDate for that field.
- Attach a custom name and value pair to an event result, where:
  - name=VisitClinicalItem.
  - value=<the clinical item value in the subject visit template in Siebel CTMS>.
- If you are sending activity completion information, you need to have two custom data name and value pairs on the event result. In addition to VisitClinicalItem, you need a name and value pair where:
  - name=ActivityClinicalItem.
  - value=<clinical item for the activity in the Subject Visit Template in Siebel CTMS>.

Specifying codelist value mapping

Some data that you send to Siebel CTMS is a list of values in Siebel and a codelist in the InForm application. You must map the InForm codelist code to the Siebel CTMS Language Independent Code (LIC) so that the InForm Publisher application can send the correct value to the Siebel CTMS web service.

The MappingName element contains the pf:Alias name for the field. The InForm element contains a single codelist value from the InForm application. The CTMS element contains a single codelist value from Siebel CTMS.

The following table contains examples of data conversion for codelists.

Example of mapping to InForm and CTMS

MappingName	InForm	CTMS
ScreenFailureReason	Inclusion Criteria Not Met	Did not meet criteria
ScreenFailureReason	Exclusion Criteria Not Met	Did not meet criteria
ProtocolDeviation	Yes	Y
ProtocolDeviation	No	N
EligibleFlag	True	Y
EligibleFlag	False	N

Use the following case-sensitive values for specifying codelist value mapping:

- EarlyTerminationReason
- EligibleFlag
- ProtocolDeviation
- ProtocolViolation
- ScreenFailureReason
- WithdrawnReason

For information on loading the InForm mapping table for a study, see the *Installation Guide*.

Configuring the Siebel CTMS subscriber

After you define event triggers and results on a study, you must configure a subscriber. Subscriber configuration specifies the frequency of event evaluation, which events are to be evaluated, and web service endpoint details.

For more information, see the *Installation Guide*.

Siebel CTMS rules

- The CTMS study ID in Siebel CTMS must be identical to the InForm study name.
- The Protocol Site number in Siebel CTMS must be identical to the Site Mnemonic in the InForm application.
- Each event must be mapped to only one visit or visit activity in Siebel CTMS.
- The Schedule Date used for scheduling subject visits in Siebel CTMS must be greater than or equal to the Encounter Date.
- Siebel CTMS accepts only one subject status per message. You cannot send both Screened and Enrolled statuses at one time as a subject is entered into the InForm application.

For more information, see the Siebel CTMS documentation.

Example of InForm to Siebel CTMS integration

For Study A, you want to track enrollment counts, early withdrawals and study completion. You also want visit completion information so that you can pay the sites.

- You are using a Subject Visit Template (SVT) in Siebel CTMS that uses status tracking visits. In the SVT in Siebel CTMS, you have the maximum amount of cycles (4) defined.
- The study has a variable number of dosing visit cycles, depending upon the subject response to the drug.
- You are using repeating visits in the InForm application and have an item on one of the forms in the visit to enter the cycle number of the visit.

Note: All events that you must define will have the same destination, which will be an endpoint alias defined for a subscriber with a CTMS endpoint type.

The custom events you must define include:

Subject enrollment event

- **Event Name**—SubjectEnrollment.
- **Priority**—1.
- **Prerequisite**—None.
- **Trigger**—Can be defined on the enrollment form and can be based on a dataValueChange(enrollmentdate) and on the enrollmentDate not being empty.
- **Results**—Define a result for every item collected in the InForm application that you want to send to Siebel CTMS.
  - EnrollmentDate with pf:Alias=EnrollmentDate,EncounterDate,ScheduleDate.
  - SubjectNumber with pf:Alias=SubjectNumber.
  - Custom data name and value pair defined on enrollmentDate as name=EnrolledStatus value=enrolled.

Note: The value is the language independent code in Siebel CTMS for the status Enrolled.

This event creates a subject in Siebel with a status of Enrolled.

Early withdrawal event

- **EventName**—EarlyWithdrawal.
- **Priority**—2.
- **Prerequisite**—None.
- **Trigger**—dataValueChange(WithdrawalDate) and withdrawalDate is not empty.
- **Results**:
  - SubjectNumber with pf:Alias=SubjectNumber.
  - EnrollmentDate with pf:Alias=EnrollmentDate,EncounterDate,ScheduleDate.
  - WithdrawalDate with pf:Alias=WithdrawnDate.
  - WithdrawReason with pf:Alias=WithdrawnReason.

This event withdraws the subject in Siebel CTMS.

Subject completion event

- **Event Name**—SubjectCompletion.
- **Priority**—2.
- **Prerequisite**—None.
- **Trigger**—VisitComplete status for Completion visit in the InForm application.
- **Results**:
  - DOV.visitDate with pf:Alias=CompletedDate.
  - Custom data with name and value pair: name=VisitClinicalItem value=<clinical item of the completion visit in Siebel SVT>.

This marks the visit as complete and specifies the date of visit as the completion date. If this visit is a status tracking visit in Siebel CTMS SVT, the subject status is set to StudyComplete.

Cycle visit completion event

You must define an event for each set of cycles.

- **EventName**—Cycle1VisitComplete.
- **Priority**—2.
- **Prerequisite**—None.
- **Trigger**—VisitComplete status for Cycle visit and dataValueMatch for Cycle item=1.
- **Results**:
  - DOV.visitDate with pf:Alias=CompletedDate.
  - Custom data with name and value pair: name=VisitClinicalItem value=<clinical item of the completion visit in Siebel SVT for cycle 1 visit>.

This marks the cycle 1 visit as complete in Siebel CTMS and specifies the date of visit as the completion date.

Completion visit date change event

For each of the visits, you must define an event to send over changes in the completed date for the visit.

- **EventName**—CompletionVisitDateChange
- **Priority**—3
- **Prerequisite**—SubjectCompletion event

  Note: Because a prerequisite event has been specified, the event will only fire if the visit had already been sent over as completed to Siebel CTMS.

- **Trigger**—dataValueChange on DOV.visitDate and DOV.visitDate is not empty.
- **Results**:
  - DOV.visitDate with pf:Alias=CompletedDate.
  - Custom data with name and value pair: name=VisitClinicalItem value=<clinical item of the completion visit in Siebel SVT>.

If the date of visit changes for the subject completion visit in the InForm application and the subject visit completion event has already been sent to Siebel CTMS, the completed date is updated in Siebel CTMS for this visit.

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 Help Center (***https://docs.oracle.com/en/industries/health-sciences/inform-publisher/index.html***)—The most current documentation set.

Documentation accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website (**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 or Support Cloud. For information, visit ***http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info*** if you are hearing impaired.