15 Importing Data

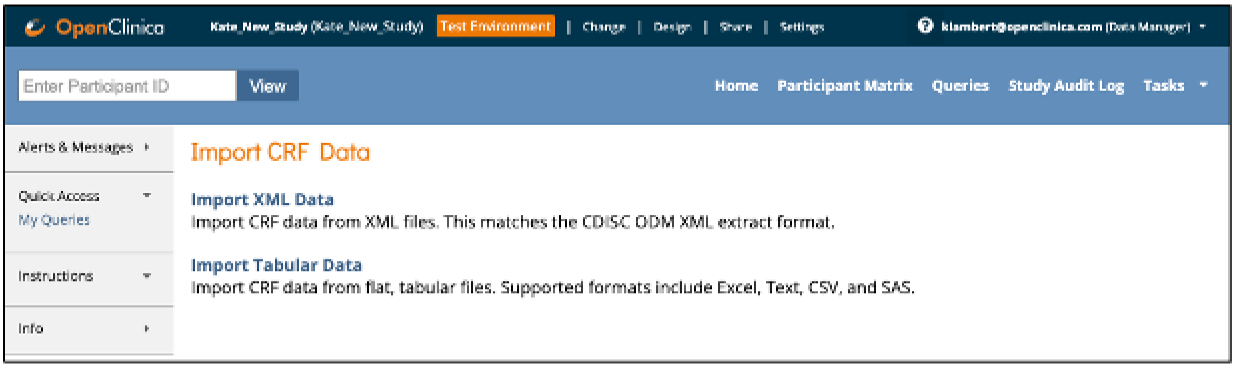
This page describes how to import data into electronic Case Report Forms (eCRFs) using CDISC ODM XML or tabular formats.

□ **Notes:**

- You can upload file attachments (e.g., PDF) for import, but it must be done with the OpenClinica API. It is not possible to import files via the Import CRF Data screen. For more information, refer to **Clinicaldata (Import and Retrieve Clinical Data).**
- You cannot import into a Events or Forms that have a status of **Locked**, **Archived**, or **Removed**. If data for a Participant is imported into Events or Forms, the status of the Participant will be **Available**.

To Access the Import Data Screens:

1. Click **Tasks** in the Study Runner header row.
2. Select **Import Data**.
3. Choose either:
    - **Import XML CRF Data,** or
    - **Import Tabular Data**



Import CRF Data

Import XML Data
Import CRF data from XML files. This matches the CDISC ODM XML extract format.

Import Tabular Data
Import CRF data from flat, tabular files. Supported formats include Excel, Text, CSV, and SAS.

Creating an Event Scheduling File:

1. Open Excel and create a new file.
2. Add the following column headings:
    1. **Participant ID**
    2. **StudyEventOID**
    3. **StartDate**
3. In the **Participant ID** column, enter each Participant OID.
4. In the **StudyEventOID** column, enter each Event OID.
5. In the **StartDate** column, enter the start date of each Event in **YYYY-MM-DD** format.
6. Save the file as a **CSV** file.

□ The CSV file is specific to each site, so the Participant List in the file must match the Participant List at the site.

For Information on Importing Data, See the Following Sections:

15.1 Importing XML Files

You can download a sample XML file on the Import XML CRF Data screen.

You can import the following types of data:

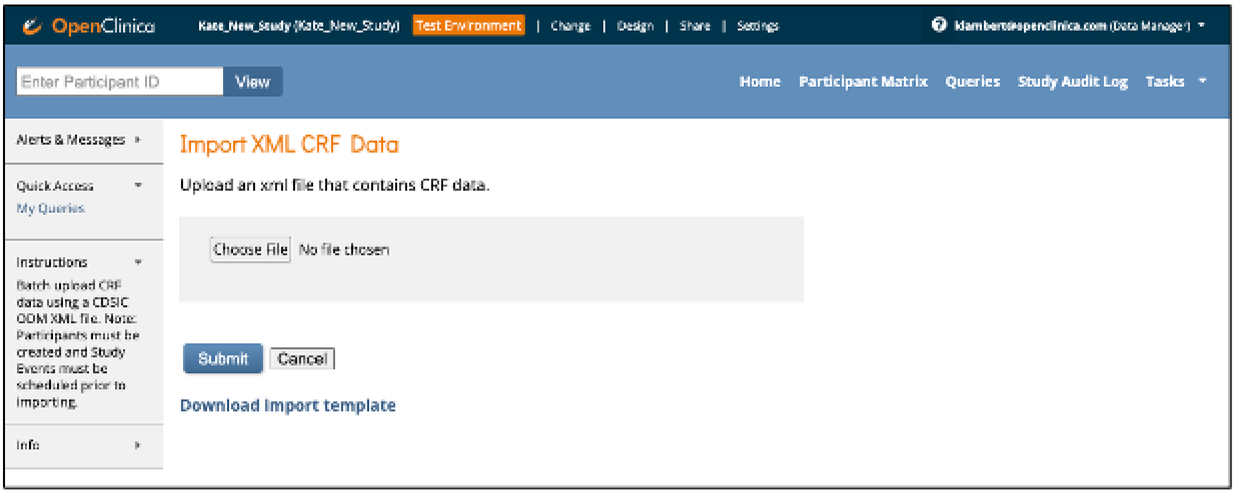
- Item data
- SDV Status
- Annotation
- Query
- Signature
- Form
- Study Event
- Participant
- Item Group

Import jobs appear on the **Bulk Actions Log**. The **Type** column displays the imported data type, as listed above.

To Import XML CRF Data:

1. Click **Choose File**, and select an XML file.
2. Click the **Submit** button.
3. The screen displays a message informing you that the job has started running.

You can click the **Tasks** button and select **Bulk Actions Log** to check the status of the job. You can view, download, or delete the import from there.



Import XML CRF Data

Upload an xml file that contains CRF data.

Batch upload CRF data using a CDSIC ODM XML file. Note: Participants must be created and Study Events must be scheduled prior to importing.

Below are some general rules when building the import XML file:

The Import File Must Contain the Following Namespaces:

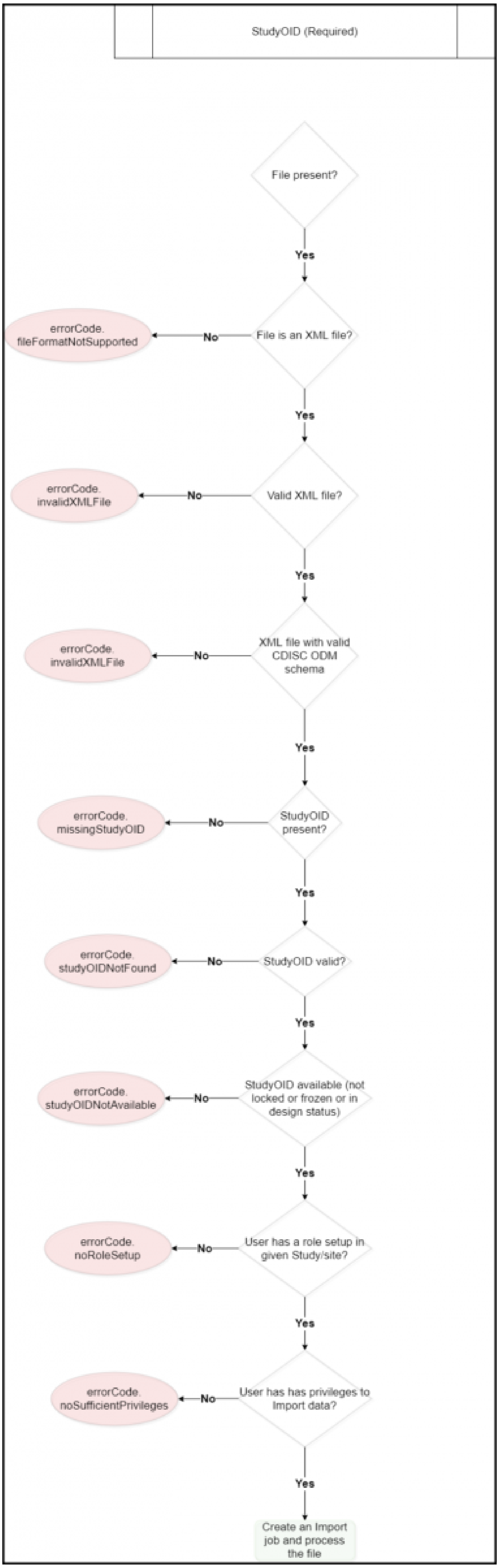
```
 <?xml version="1.0" encoding="UTF-8"?>

<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3"
xmlns:OpenClinica="http://www.openclinica.org/ns/odm_ext_v130/
v3.1" ODMVersion="1.3">
```

In addition to the above, FileOID, FileType, Description, and CreationDateTime information can also be provided within the **<ODM>** tag. This information is not used by the OpenClinica system. However, you can use these properties for internal documentation.

- The Import file should only contain one **<ClinicalData>** tag pair. Though the Import file will not generate an error if it contains multiple **<ClinicalData>** tag pairs, the system only processes the data in the first **<ClinicalData>** tag pair and ignores the others.
- The Import file can contain data for multiple subjects as long as they are all between the opening and closing tags of the first **<ClinicalData>** tag. To allow for multiple subjects, copy everything between the opening and closing tags of **<SubjectData>**, including the tags, and then paste it after the closing tag of **</SubjectData>**. Then update the OIDs and data in the new section.
- The Import file can contain multiple **<StudyEventData>**, **<FormData>**, **<ItemGroupData>** and **<ItemData>** tags as long as their hierarchy is maintained. Copy and paste these sections as needed to import multiple events, forms, item groups, and items.
- When a repeating, visit-based event definition contains multiple forms, then all the forms of that event repeat/ordinal can be combined within the same **<StudyEventData>** tag with one repeat/ordinal value.
    - For every new study event repeat/ordinal for a participant, the **<StudyEventData>** tag construct needs to be repeated within that participant's **<SubjectData>** tags.
    - When a repeat ordinal is not provided for a repeating study event, then the system automatically generates a new repeat/ordinal of that event for that participant.
- When a common event definition contains multiple forms, then every form should be provided with its own **<StudyEventData>** tag and a new event/repeat ordinal. When an ordinal is not provided then the system creates a new repeat for the given event.
- Import is not allowed into a form when any of the following conditions are met:
    - Study/Site is **In Design**, **Frozen**, or **Locked** status
    - Study Event (or repeat thereof) is in **Locked** status, has been removed for the participant, or is no longer available in the study
    - Form is in **Completed** status, has been removed for the participant, or is no longer available in the study

Below is an image that displays the workflow for importing an XML file:


StudyOID (Required)
File present?
Yes
File is an XML file?
No
errorCode. fileFormatNotSupported
Yes
Valid XML file?
No
errorCode. invalidXMLFile
Yes
XML file with valid CDISC ODM schema
No
errorCode. invalidXMLFile
Yes
StudyOID present?
No
errorCode. missingStudyOID
Yes
StudyOID valid?
No
errorCode. studyOIDNotFound
Yes
StudyOID available (not locked or frozen or in design status)
No
errorCode. studyOIDNotAvailable
Yes
User has a role setup in given Study/site?
No
errorCode. noRoleSetup
Yes
User has has privileges to Import data?
No
errorCode. noSufficientPrivileges
Yes
Create an Import job and process the file

Download the CDISC XML import template on the **Import XML CRF Data** screen.

The XML file can also be generated using an existing Participant's casebook.

For example, if data for one of the Participants has already been captured in a Form, and now the data needs to be entered into forms for other Participants, the user can follow the steps below to quickly generate the XML file to import the clinical data into other participants.

To Generate an XML File With an Existing Participant Casebook:

1. Go the **Participant Details** screen for the Participant with the data that was captured.
2. Scroll to the bottom, and click on the **Download data** link.
3. Select the CDISC XML format of your choice, and uncheck all the checkboxes
4. Click **Run now**.
5. Once the import process is complete, save the casebook as an XML file in your local directory.
6. Open the file in a Notepad application.
7. Change the Participant ID and OID values, along with the values of the items for other Participant. If data for multiple Participants needs to be captured, then copy and paste everything between the opening and closing brackets of **<SubjectData>** tag, and edit the Participant ID, OID, and item values for the other Participant.
8. Save the file, and call the Import API using this file.

Please refer to the sections below for details about each data tag in the CDISC ODM XML:

- Subject Data Tag
- Study Event Data Tag
- Item Group Data Tag
- Item Data Tag

Subject Data Tag

The SubjectData tag contains information about the participant for which the given clinical data are provided.

```
Usage:  <SubjectData SubjectKey="SS_OID" OpenClinica:StudySubjectID="PID"
OpenClinica:Status="Available">
```

This tag has the following properties:

SubjectKey

- **SubjectKey** represents the unique identifier (Participant OID) of the participant whose data is being imported in the given study.
  - Participant OID is a system-generated ID that does not change once assigned to a participant.
- **SubjectKey** is conditionally required. Either **SubjectKey** or **OpenClinica:StudySubjectID** must be provided.
- When **SubjectKey** and **OpenClinica:StudySubjectID** are both provided, the values for both must represent the same participant. If they do not, an appropriate error code is returned in the import log file.
- If you are not providing a **SubjectKey**, remove the **SubjectKey="SS_OID"** portion of this line of code.

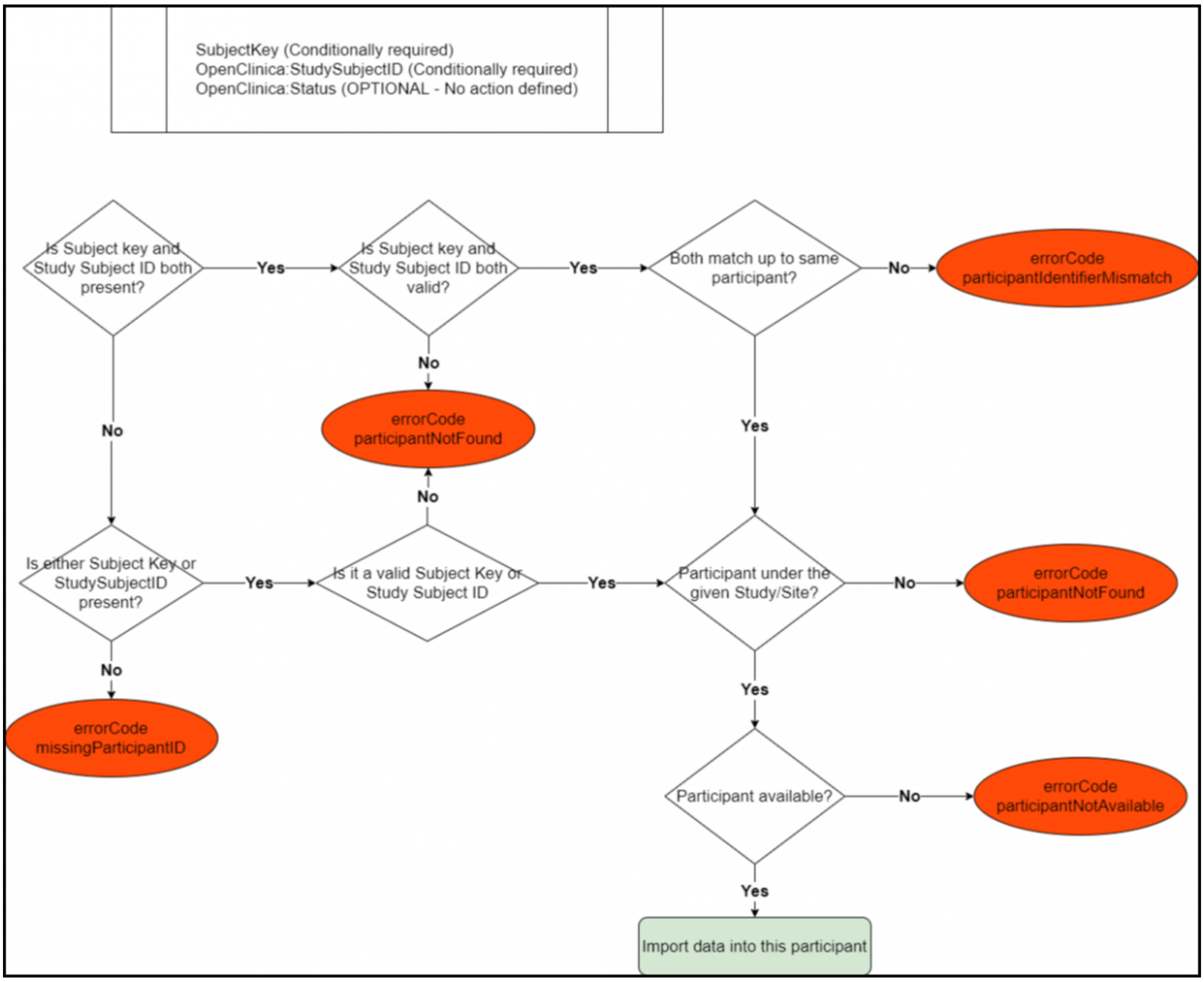
OpenClinica:StudySubjectID

- **OpenClinica:StudySubjectID** represents the ID of the participant (PID) whose data you're importing into the given study.
  - PID can be either automatically generated by the system when the participant is added to the study, or manually created at that time. This ID can be changed by the site users or the data manager during the study.
- This property is conditionally required. Either this key or the SubjectKey property must be provided.
- When **SubjectKey** and **OpenClinica:StudySubjectID** are both provided, the values for both must represent the same participant. If they do not, an appropriate error code is returned in the import logfile.
- If you are not providing **OpenClinica:StudySubjectID**, remove **the OpenClinica:StudySubjectID="PID"** portion of this line of code.

OpenClinica:WorkflowStatus

- This property represents the status of the participant whose data is being imported into the given study.
- The value for this property is not used, but does no harm if left in the code.

Below is an Image that Displays the SubjectData Tag Workflow and Potential Error Messages:


SubjectKey (Conditionally required)
OpenClinica:StudySubjectID (Conditionally required)
OpenClinica:Status (OPTIONAL - No action defined)
Is Subject key and Study Subject ID both present?
Yes
Is Subject key and Study Subject ID both valid?
Yes
Both match up to same participant?
No
errorCode participantIdentifierMismatch
No
errorCode participantNotFound
No
Yes
No
Is either Subject Key or StudySubjectID present?
Yes
Is it a valid Subject Key or Study Subject ID
Yes
Participant under the given Study/Site?
No
errorCode participantNotFound
No
errorCode missingParticipantID
Yes
Participant available?
No
errorCode participantNotAvailable
Yes
Import data into this participant

Below is a List of all the Error Codes and their Descriptions:

- **errorCode.missingParticipantID:** Neither **SubjectKey** nor **OpenClinica:StudySubjectID**

property is present.
- **errorCode.participantNotFound:** No participant is found in the given study for the given **SubjectKey** or **OpenClinica:StudySubjectID** value(s).
- **errorCode.participantIdentiferMismatch: SubjectKey** and **OpenClinica:StudySubjectID** both have values but the values represent two different participants in the given study.
- **errorCode.participantNotAvailable:** The participant has a **Removed** status in the given study

Study Event Data Tag

The **StudyEventData tag** contains information about the event for which the given clinical data are provided.

```
Usage:  <StudyEventData StudyEventOID="SE_EVENT1"
OpenClinica:EventName="Event1 name" StudyEventRepeatKey="1"
OpenClinica:StartDate="2019-06-14" OpenClinica:EndDate="2019-06-14"
OpenClinica:WorkflowStatus="initial data entry">
```

This tag has the following properties:

- StudyEventOID (Required)
- OpenClinica:EventName (IGNORED)
- StudyEventRepeatKey (Conditionally required)
- OpenClinica:StartDate (Conditionally required)
- OpenClinica:EndDate (OPTIONAL)
- OpenClinica:WorkflowStatus (IGNORED)

StudyEventOID

- StudyEventOID is REQUIRED and represents the unique identifier (OID) of the event into which the data are to be imported.
- When **StudyEventOID** is missing, or no matching event is found for the given **StudyEventOID**, then the system returns an appropriate error code in the import log file and data for this event is not imported.

OpenClinica:EventName

- OpenClinica:EventName represents the name of the event.
- Value for OpenClinica:EventName is not used, but does no harm if left in the code.

StudyEventRepeatKey

- **StudyEventRepeatKey** is conditionally required.
- **StudyEventRepeatKey** represents the repeat key/ordinal for a particular study event for a participant.
- For a repeating event:
  - When the given data needs to be updated for a particular study event repeat/ordinal then this property is required with a valid repeat/ordinal number.
  - When this property is missing then the system automatically generates a new repeat/ordinal for the repeating event. For a visit-based event, **OpenClinica:StartDate** is required. If it is missing then an appropriate error code is returned in the log file
- For a non-repeating event:

    - **StudyEventRepeatKey** is not used. The system inserts or updates the data in the non-repeat event.
    - If repeat key is provided and the system finds issues with the value, an appropriate error code is returned in the import log file and no data are imported for that study event.

OpenClinica:StartDate

- **OpenClinica:StartDate** represents the start date of a visit-based event.
- The value of this property must be in **yyyy-MM-dd** format (e.g., **2019-01-31**).
- For visit-based events:
    - When **StartDate** is provided and a repeat key is not provided, the system automatically creates a new repeat using the provided **StartDate** value.
    - If NEITHER a **StartDate** NOR a repeat key is provided, then an appropriate error is returned and data is not imported for that event.
    - When **StartDate** is provided with a repeat key/ordinal value of an existing event repeat, the value of this property is not used and has no impact on the study event or import functionality.
- For common events:
    - **OpenClinica:StartDate** is not used, but does no harm when left in the code.

OpenClinica:EndDate

- This property represents the end date of a visit-based event.
- The value of this property must be in **yyyy-MM-dd** format (e.g., **2019-01-31**).
- Visit-based events:
    - When the system is expected to automatically generate a new visit based event repeat/ordinal and **StartDate** is provided then the **OpenClinica:EndDate** value is set as the end date of the newly generated visit-based event repeat.
    - This property is optional and when not provided has no impact on the import functionality.
    - When **EndDate** is provided with a repeat key/ordinal value of an existing event repeat, the value of this property is not used and has no impact on the study event or import functionality.
- Common events:
    - End date is not used and has no impact on the import functionality for common events.

OpenClinica:WorkflowStatus

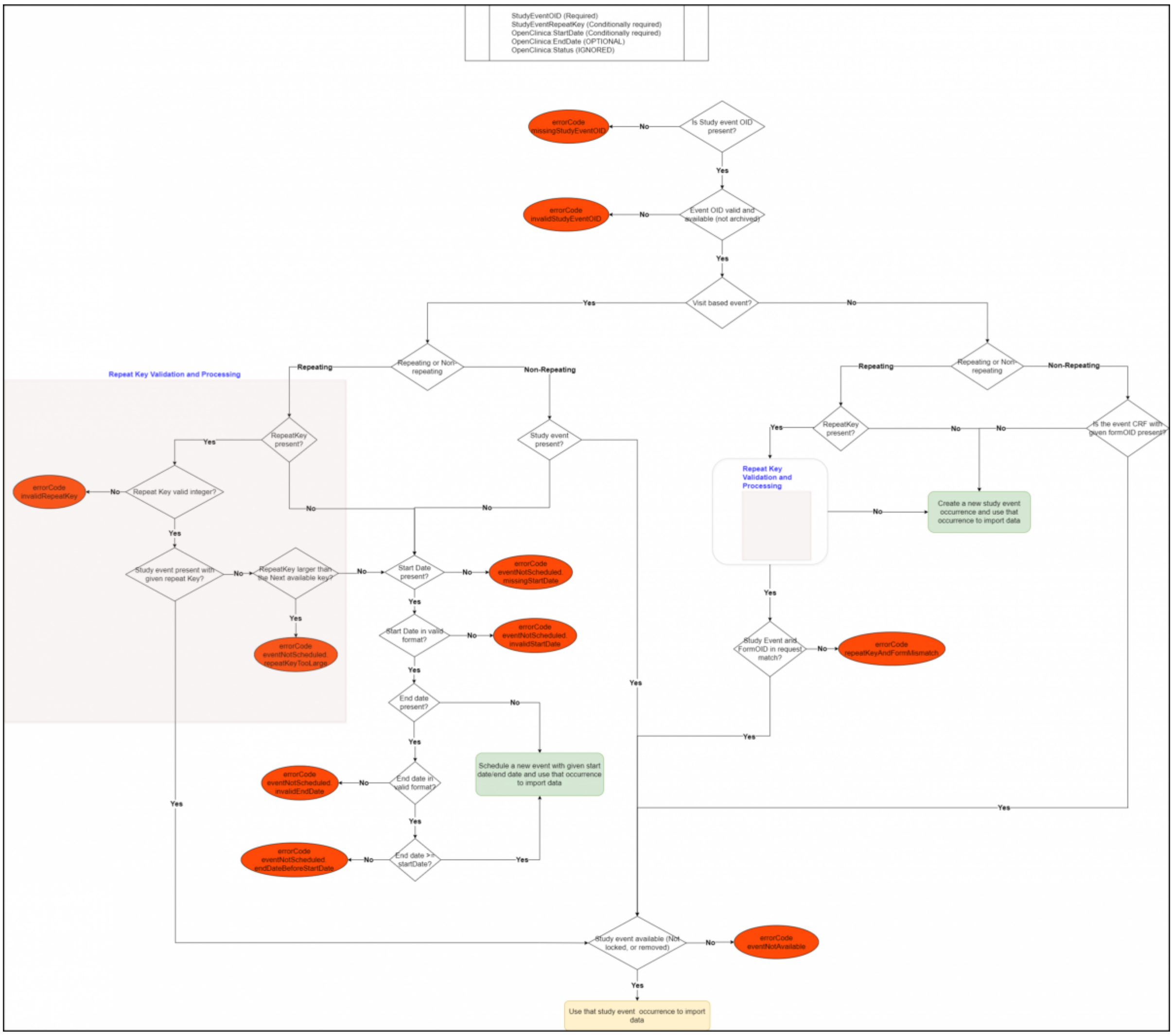
- This property represents the status of a study event.

- The value for this property is not used during the import process and has no impact on the import functionality.

If there are any errors during the import process with the value(s) provided for any of the required or conditionally required properties in the **StudyEventData** tag, the system returns an appropriate error code in the import log file and data are not imported for that participant.

Note: *The previous property of* ***OpenClinica:Status*** *is still available to aid in compatibility. However, this will only be available for a limited time.*

Below is an Image that Displays the StudyEventData tag Workflow and Potential Error Messages:


StudyEventOID (Required)
StudyEventRepeatKey (Conditionally required)
OpenClinica:StartDate (Conditionally required)
OpenClinica:EndDate (OPTIONAL)
OpenClinica:Status (IGNORED)
Is Study event OID present?
No
errorCode missingStudyEventOID
Yes
Event OID valid and available (not archived)
No
errorCode invalidStudyEventOID
Yes
Visit based event?
Yes
No
Repeating or Non-repeating
Repeating
Non-Repeating
Repeat Key Validation and Processing
RepeatKey present?
Yes
No
Repeat Key valid integer?
No
errorCode invalidRepeatKey
Yes
Study event present with given repeat Key?
No
RepeatKey larger than the Next available key?
Yes
errorCode eventNotScheduled. repeatKeyTooLarge
No
Study event present?
No
Yes
Start Date present?
No
errorCode eventNotScheduled. missingStartDate
Yes
Start Date in valid format?
No
errorCode eventNotScheduled. invalidStartDate
Yes
End date present?
No
Schedule a new event with given start date/end date and use that occurrence to import data
Yes
End date in valid format?
No
errorCode eventNotScheduled. invalidEndDate
Yes
End date >= startDate?
No
errorCode eventNotScheduled. endDateBeforeStartDate
Yes
Repeating or Non-repeating
Repeating
Non-Repeating
RepeatKey present?
Yes
No
No
Is the event CRF with given formOID present?
Repeat Key Validation and Processing
No
Create a new study event occurrence and use that occurrence to import data
Yes
Study Event and FormOID in request match?
No
errorCode repeatKeyAndFormMismatch
Yes
Yes
Yes
Study event available (Not locked, or removed)
No
errorCode eventNotAvailable
Yes
Use that study event occurrence to import data

Below is a list of all the error codes and their descriptions:

- **errorCode.missingStudyEventOID:** studyEventOID is not present in the **StudyEventData** tag.
- **errorCode.invalidStudyEventOID:** The given study event OID does not represent a valid study event OID.
- **errorCode.invalidRepeatKey:** The given study event repeat key value is not a valid integer.
- **errorCode.eventNotScheduled.repeatKeyTooLarge:** The event repeat is provided which needs to be scheduled, but the given repeat key is larger than the next available repeat that should be scheduled.
- **errorCode.eventNotScheduled.missingStartDate:** The system is unable to schedule the visit-based event (or repeat thereof) due to a missing start date.
- **errorCode.eventNotScheduled.invalidStartDate:** The system is unable to schedule the visit-based event (or repeat thereof) because the given start date value not valid or not in valid format.
- **errorCode.eventNotScheduled.invalidEndDate:** The system is unable to schedule the visit-based event (or repeat thereof) because the given end date value not valid or not in valid format.
- **errorCode.repeatKeyAndFormMismatch:** The repeat key is present for a common event, but the form that's provided with the given repeat key does not represent an existing pair in the system for the given participant.

- **errorCode.eventNotAvailable:** The study event or study event repeat provided is in a status that does not allow any further data entry. (e.g. **Locked**, **Skipped**, or **Stopped**).

Form Data Tag

The FormData tag contains information about the event form (e.g. eCRF) for which the given clinical data are provided.

```
Usage:  <FormData FormOID="F_ECRF1" OpenClinica:FormName="Event Form1"
OpenClinica:FormLayoutOID="v1.0" OpenClinica:WorkflowStatus="initial data
entry">
```

This Tag has the Following Properties:

- **FormOID** (Required)
- **OpenClinica:FormName** (IGNORED)
- **OpenClinica:FormLayoutOID** (Conditionally required) -- Default form version is assigned to the event form when **OpenClinica:FormLayoutOID** is not provided.
- **OpenClinica:WorkflowStatus** (OPTIONAL) -- The form status is set to **initial data entry** after data import when **OpenClinica:WorkflowStatus** is not provided.

FormOID

- **FormOID** is REQUIRED and it represents the unique identifier (OID) of the event form in which the data are to be imported.
- When FormOID is missing or there is no matching form found, the system returns an appropriate error code in the log file and data are not imported into the referenced form.

OpenClinica:FormName

- **OpenClinica:FormName** represents the name of the event form.
- Value for this property is not used, but no there is no harm if it is left in the code.

OpenClinica:FormLayoutOID

- **OpenClinica:FormLayoutOID** is optional.
- **OpenClinica:FormLayoutOID** represents the form version that gets assigned to the form after importing the data.
- **If OpenClinica:FormLayoutOID** is not provided, the system assigns the default form version for the given participant.
- If the given value is not valid, or no matching version exists for the given form, an appropriate error code is returned in the log file and data are not imported into the referenced form.

OpenClinica:WorkflowStatus

- **OpenClinica:WorkflowStatus** represents the status of the form after the data has been imported successfully.
- Possible values for this property are: **initial data entry** and **complete**. If any other value is provided, the system will return an appropriate error code in the import log file and data are not imported into the referenced form.
- If no value is provided, then after successful import of the data, the form status is set as **initial data entry**.
- If there are any errors when importing any of the item data values for the form, the form

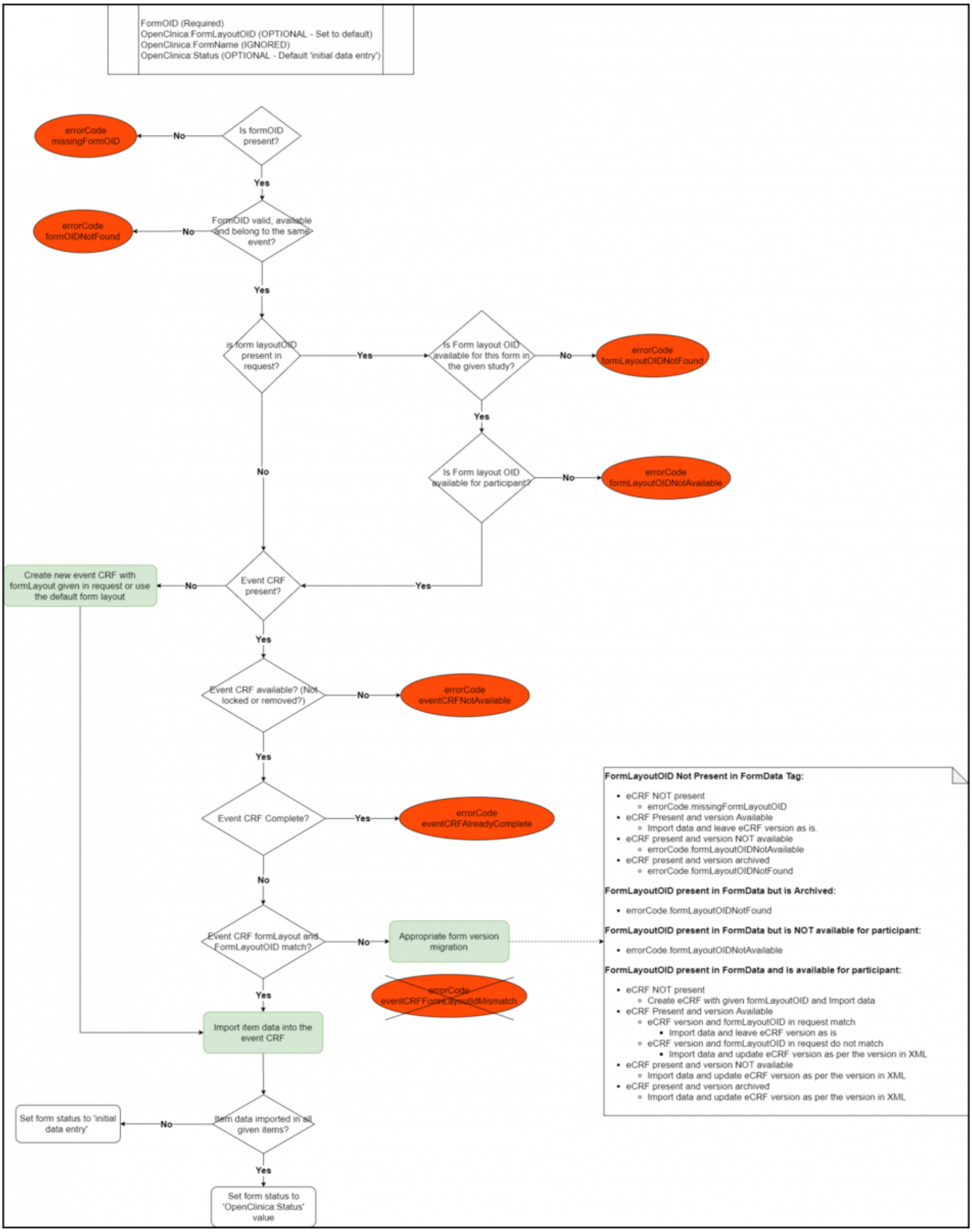
status is set as **initial data entry**; even if the provided status is **complete**.

OpenClinica:SdvStatus

- **OpenClinica:SdvStatus** represents the Source Data Verification (SDV) status on a form. When importing XML for adding or updating data on a form, SDV verification changes to the items and form occur based on SDV configurations for the form.
- **OpenClinica:SdvStatus="Verified"** will SDV Verify the form on upload.
- **OpenClinica:SdvStatus="Not Verified"** will update the form SDV Status as "Not Verified" on upload.
- The SDV Status will not update forms with the SDV Configuration of **Not Applicable**, but will update forms with all other SDV configurations (Not Required, Partial Required, 100% Required, and Item-Level).
- Item-Level SDV forms marked as Verified will also mark all **SDV Required** items on the forms as **Verified**.
- Items on Item-Level SDV forms are not affected when the form is marked as **Not Verified**.

During import, if the system finds any issues with the value(s) provided for any of the required or conditionally required properties, an appropriate error code is returned in the log file and data are not imported for that participant.

Below is an Image that Displays the SubjectData tag Workflow and Potential Error Messages:


FormOID (Required)
OpenClinica:FormLayoutOID (OPTIONAL - Set to default)
OpenClinica:FormName (IGNORED)
OpenClinica:Status (OPTIONAL - Default 'initial data entry')
Is formOID present?
No
errorCode missingFormOID
Yes
FormOID valid, available and belong to the same event?
No
errorCode formOIDNotFound
Yes
is form layoutOID present in request?
Yes
Is Form layout OID available for this form in the given study?
No
errorCode formLayoutOIDNotFound
Yes
Is Form layout OID available for participant?
No
errorCode formLayoutOIDNotAvailable
Yes
No
Event CRF present?
No
Create new event CRF with formLayout given in request or use the default form layout
Yes
Event CRF available? (Not locked or removed?)
No
errorCode eventCRFNotAvailable
Yes
Event CRF Complete?
Yes
errorCode eventCRFAlreadyComplete
No
Event CRF formLayout and FormLayoutOID match?
No
Appropriate form version migration
errorCode eventCRFFormLayoutIdMismatch
Yes
Import item data into the event CRF
Item data imported in all given items?
No
Set form status to 'initial data entry'
Yes
Set form status to 'OpenClinica:Status' value
FormLayoutOID Not Present in FormData Tag:
eCRF NOT present
errorCode.missingFormLayoutOID
eCRF Present and version Available
Import data and leave eCRF version as is.
eCRF present and version NOT available
errorCode.formLayoutOIDNotAvailable
eCRF present and version archived
errorCode.formLayoutOIDNotFound
FormLayoutOID present in FormData but is Archived:
errorCode.formLayoutOIDNotFound
FormLayoutOID present in FormData but is NOT available for participant:
errorCode.formLayoutOIDNotAvailable
FormLayoutOID present in FormData and is available for participant:
eCRF NOT present
Create eCRF with given formLayoutOID and Import data
eCRF Present and version Available
eCRF version and formLayoutOID in request match
Import data and leave eCRF version as is
eCRF version and formLayoutOID in request do not match
Import data and update eCRF version as per the version in XML
eCRF present and version NOT available
Import data and update eCRF version as per the version in XML
eCRF present and version archived
Import data and update eCRF version as per the version in XML

Below is a List of All the Error Codes and Their Descriptions:

- **errorCode.missingFormOID:** The FormOID is not present in the FormData tag.
- **errorCode.formOIDNotFound:** The FormOID supplied does not represent a valid form OID.
- **errorCode.formLayoutOIDNotFound:** The form version supplied does not represent a valid form version.
- **errorCode.formLayoutOIDNotAvailable:** The form version supplied is not available for the site participant.
- **errorCode.formNotAvailable:** The form is not available either because it has been removed for the participant or from the study altogether.
- **errorCode.formAlreadyComplete:** The form is already in completed status for the

participant.

Item Group Data Tag

The ItemGroupData tag contains information about the item group in the given form for which the clinical data are provided.

```
Usage:  <ItemGroupData ItemGroupOID="IG_GROUP1"
OpenClinica:ItemGroupName="Item Group1" itemGroupRepeatKey="1">
```

This Tag has the Following Properties:

- ItemGroupOID (Required)
- OpenClinica:ItemGroupName (IGNORED)
- ItemGroupRepeatKey (Conditionally required) -- When not provided, default form version is assigned.

ItemGroupOID

- ItemGroupOID is REQUIRED and it represents the unique identifier (OID) of the item group in the form for which data are to be imported.
- When ItemGroupOID is missing or there is no matching item group found in the given form, the system returns an appropriate error code in the log file and data are not imported into the given item group.

OpenClinica:ItemGroupName

- OpenClinica:ItemGroupName represents the name of the item group.
- The value for OpenClinica:ItemGroupName is not used, but does no harm if left in the code.

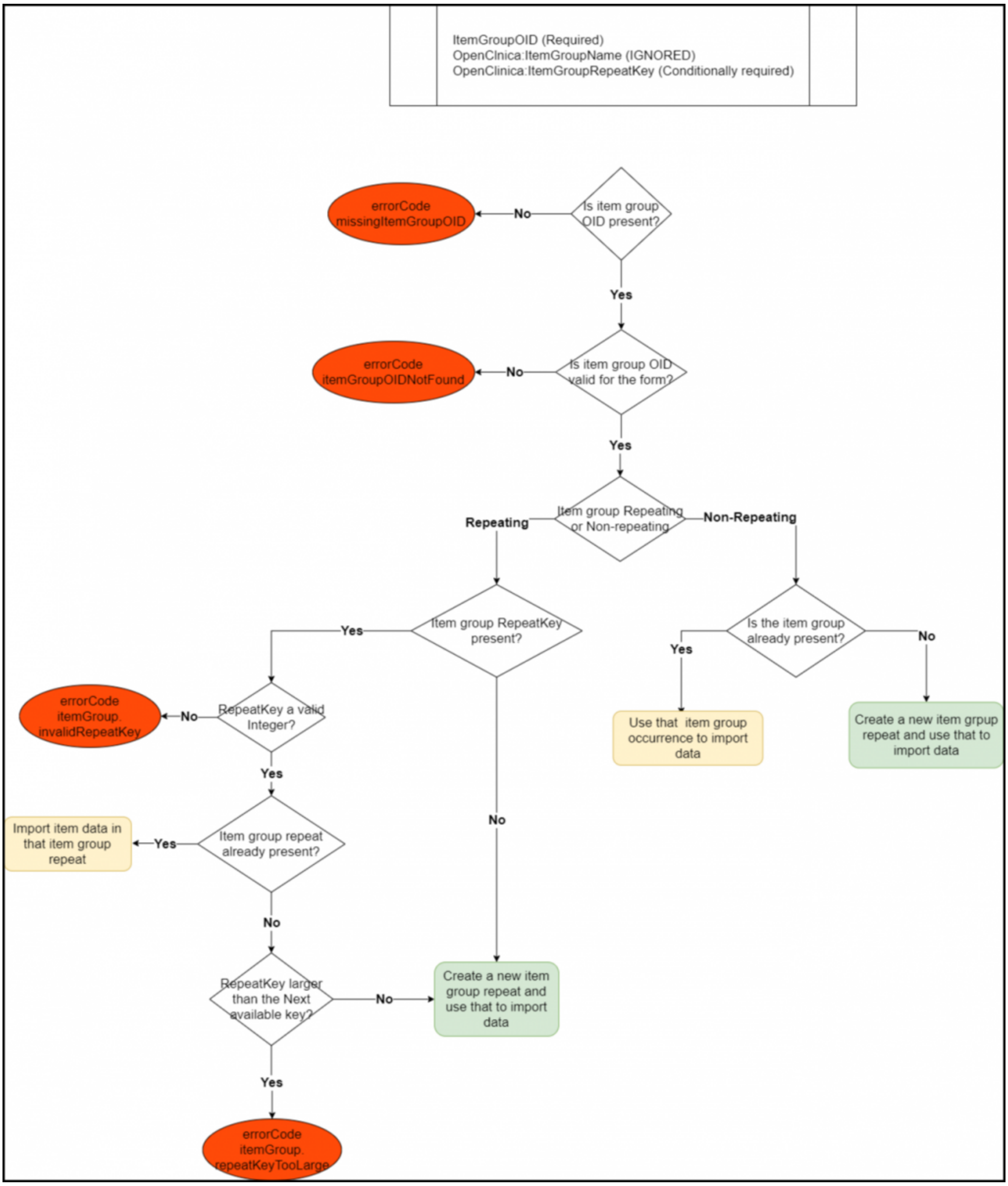
ItemGroupRepeatKey

- ItemGroupRepeatKey is optional.
- ItemGroupRepeatKey represents the repeat key/ordinal of the repeating item group.
- Repeating item group:
  - When ItemGroupRepeatKey is not provided for a repeating item group then a new repeat is automatically generated and data gets imported into that new item group repeat.
  - When a value is provided for ItemGroupRepeatKey and there is an existing repeat key with the same value, the system updates data in that item group repeat provided in the XML.
  - When a value is provided for this property but is invalid, an appropriate error code is returned in the import log file and data are not imported for the given item group repeat.
- Non-repeating item group:
  - The ItemGroupRepeatKey is not expected for non-repeating groups. Even if it is provided, the system will ignore the property and data will be inserted/updated into the first "repeat" of the non-repeating item group.

During import, if system finds any errors with the values provided, appropriate error code is returned in the import log file and data are not imported for that participant.

Below is an Image that Displays the ItemGroupData Tag Workflow and Potential Error

Messages:


ItemGroupOID (Required)
OpenClnica:ItemGroupName (IGNORED)
OpenClinica:ItemGroupRepeatKey (Conditionally required)
Is item group OID present?
No
errorCode missingItemGroupOID
Yes
Is item group OID valid for the form?
No
errorCode itemGroupOIDNotFound
Yes
Item group Repeating or Non-repeating
Repeating
Non-Repeating
Item group RepeatKey present?
Yes
RepeatKey a valid Integer?
No
errorCode itemGroup. invalidRepeatKey
Yes
Item group repeat already present?
Yes
Import item data in that item group repeat
No
RepeatKey larger than the Next available key?
No
Create a new item group repeat and use that to import data
Yes
errorCode itemGroup. repeatKeyTooLarge
No
Is the item group already present?
Yes
Use that item group occurrence to import data
No
Create a new item grpup repeat and use that to import data

Below is a List of All the Error Codes and Their Descriptions:

- **errorCode.missingItemGroupOID:** The itemGroupOID is not present in the ItemGroupData tag.
- **errorCode.itemGroupOIDNotFound:** The item group OID supplied does not represent a valid item group OID.
- **errorCode.itemGroup.invalidRepeatKey:** The item group repeat provided is not found for the given form item group.

Item Data Tag

The ItemData tag contains information about the item for which the clinical data are provided.

```
Usage:  <ItemData ItemOID="F_ECRF1" OpenClinica:ItemName="Event Form1"
Value="item value">
```

This Tag has the Following Properties:

- **ItemOID** (Required)
- **OpenClinica:ItemName** (IGNORED)
- **Value** (Required) -- When not provided, the default form version is assigned.

ItemOID

- This property is REQUIRED and represents the unique identifier (OID) of the item for which the data will be imported.
- When this property is missing or no matching item is found, the system returns an appropriate error code and the data for this item is not imported.

OpenClinica:ItemName

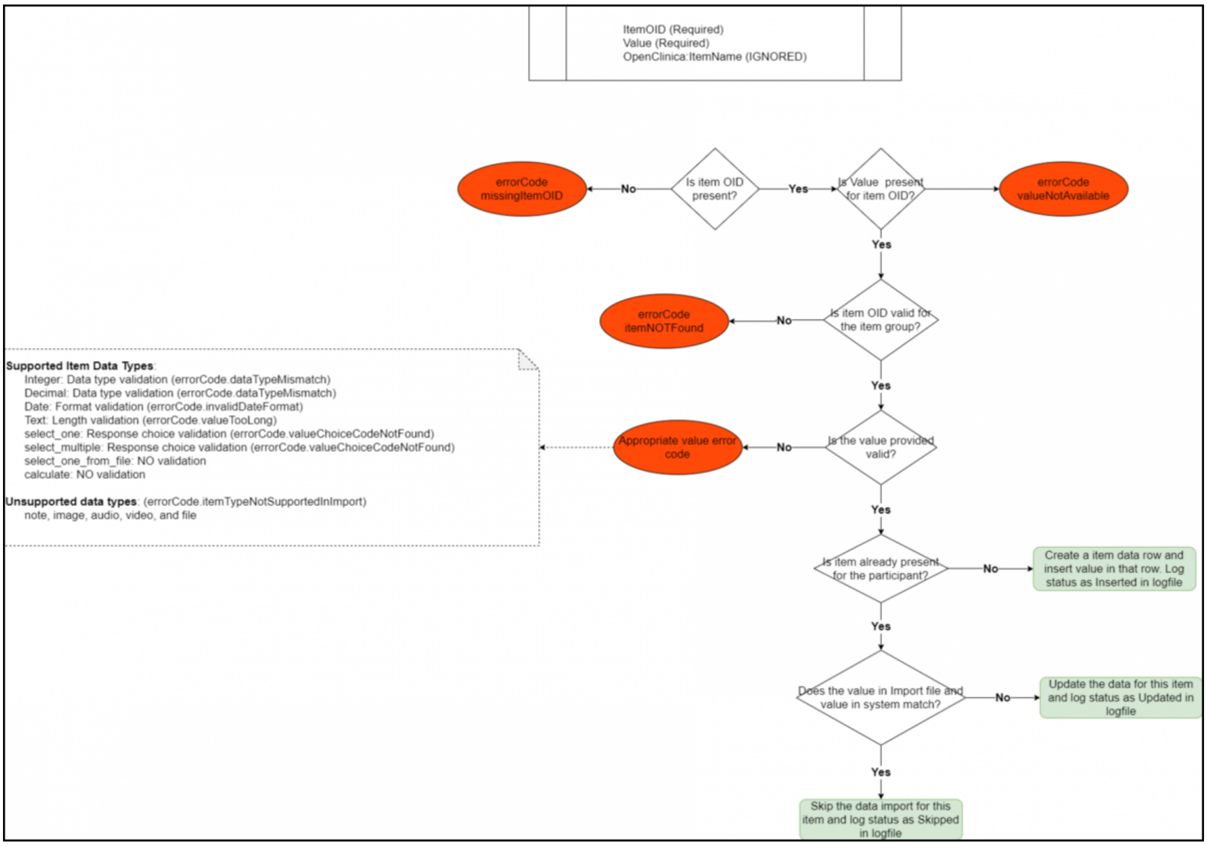
- This property represents the name of the form item.
- The value for this property is not used during the import process and has no impact on the import functionality.

Value

- This property is REQUIRED.
- This represents the value of the item to be imported.
- The value must be consistent with the item data type. If any issues are found, the system returns an appropriate error code and data are not imported in that item.
- The value must match the available response choices. If the given value is not found in the available response choices, the system returns an appropriate error code and data are not imported into that item.
- When an item data type is based on a list of response choices, the value must be the coded value of that response choice (name).
- The following list of item types are not supported as part of the import process and will be reported with an appropriate error code in the log file.
    - file
    - audio
    - video
    - image
    - note

During import, if the system discovers any issues with the value(s) provided, it will return appropriate error code in the import log file and data are not imported for that participant.

Below is an Image that Displays the SubjectData Tag workflow and Potential Error Messages:

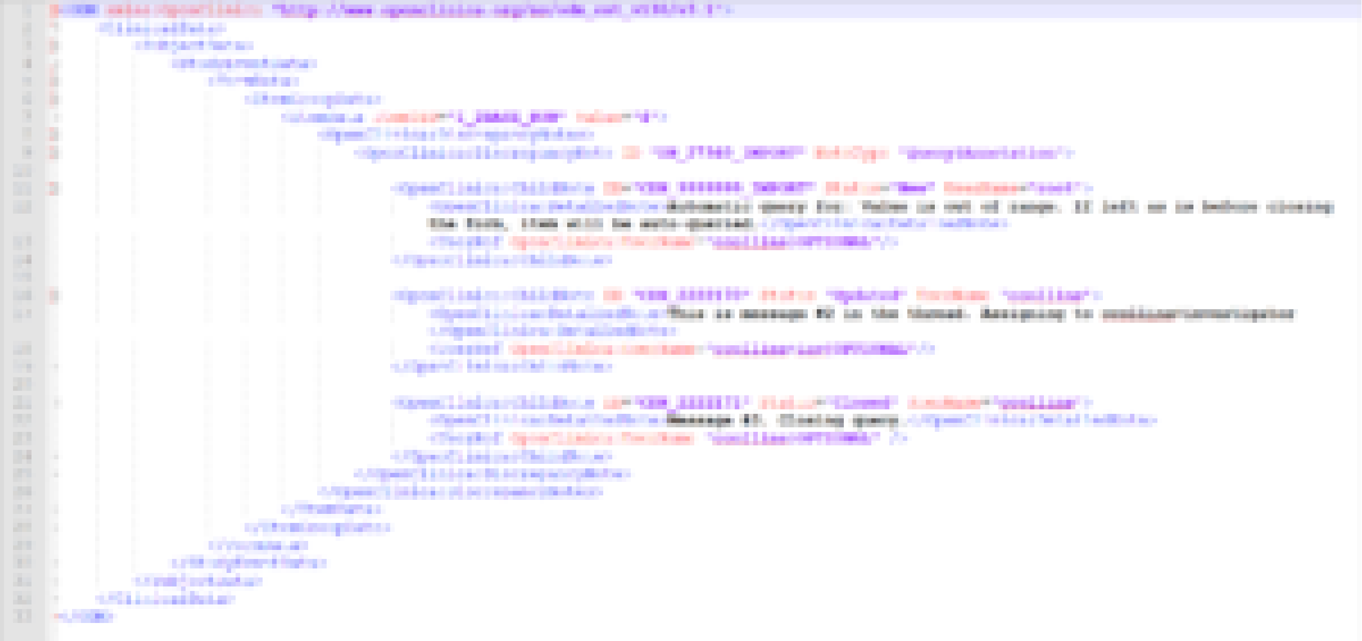

ItemOID (Required)
Value (Required)
OpenClinica:ItemName (IGNORED)
errorCode missingItemOID
No
Is item OID present?
Yes
Is Value present for item OID?
errorCode valueNotAvailable
Yes
errorCode itemNOTFound
No
Is item OID valid for the item group?
Yes
Supported Item Data Types:
Integer: Data type validation (errorCode.dataTypeMismatch)
Decimal: Data type validation (errorCode.dataTypeMismatch)
Date: Format validation (errorCode.invalidDateFormat)
Text: Length validation (errorCode.valueTooLong)
select_one: Response choice validation (errorCode.valueChoiceCodeNotFound)
select_multiple: Response choice validation (errorCode.valueChoiceCodeNotFound)
select_one_from_file: NO validation
calculate: NO validation
Unsupported data types: (errorCode.itemTypeNotSupportedInImport)
note, image, audio, video, and file
Appropriate value error code
No
Is the value provided valid?
Yes
Is item already present for the participant?
No
Create a item data row and insert value in that row. Log status as Inserted in logfile
Yes
Does the value in Import file and value in system match?
No
Update the data for this item and log status as Updated in logfile
Yes
Skip the data import for this item and log status as Skipped in logfile

Below is a List of All the Error Codes and Their Descriptions:

- **errorCode.missingItemOID:** The itemOID is not present in the ItemData tag.
- **errorCode.itemNotFound:** The given item OID does not represent a valid item OID.
- **errorCode.valueNotAvailable:** The Value property is missing in the ItemData tag.
- **errorCode.dataTypeMismatch:** The value provided for the item is inconsistent with the item's data type.
- **errorCode.invalidDateFormat:** The item data type is a date and the value provided is not a valid supported date format. Date must be provided in **yyyy-MM-dd** format.
- **errorCode.valueTooLong:** The item data type is text/string and the value provided is larger than 3999 characters.
- **errorCode.valueChoiceCodeNotFound:**
    - The item data type is a select list of response choices defined within the XLS Form and the value provided does not represent a valid response choice.
    - The item data type is a select list of response choices defined using a csv file in XLS Form.
- **errorCode.itemTypeNotSupportedInImport:** The item data type is not supported as part of XML import.

Query XML Import

Below is a screenshot of an example Query for import in XML format. These queries can be part of a normal clinical data import file.



XML Nodes and Attributes from the Sample Template Defined:

Queries will be imported as part of the ItemData node, an Item Data OID is required.

OpenClinica:DiscrepancyNotes: This node contains the full Query.

- **OpenClinca:DiscrepancyNote:** Parent note for Query, contains:
    - **NoteType:** Can be Query or Annotation.
    - **ID:** This is optional and contains the new Display ID for the note. If left blank, the system will populate this with a random, 9-digit ID. If provided, the user-specified ID will be used with a maximum of 32 characters.
    - **OpenClinca:ChildNote:** Child note within Query.
        - **ID:** This is optional and contains the new Display ID for the note. If left blank, the system will populate this with a random, 9-digit ID. If provided, the user-specified ID will be used with a maximum of 32 characters.
        - **UserName:** A valid OpenClinica UserName. The user who created the note.
        - **Status:** New, Updated, Closed, or Closed-Modified. The status of the Query as set by the Note.
    - OpenClinica:DetailedNote: The text of the note comment.
        - **UserRef:** User Name: Optional, provides a valid OpenClinica UserName for the user the assignee of the query.

Decription of Query Import Functionality

The ID attribute of both child and parent discrepancy notes in the import XML is optional. This will be a new, user-visible attribute that will be a unique value for each note that is distinct from the current internal database ID. If included in the import, the note ID will be imported into the system as is unless the ID is a duplicate within the system. In that case, the note will not be imported and this will be listed in the import log.

When the Display IDs are being automatically generated, if the system generates an ID that by chance is a duplicate of another ID, the system will retry the ID generation 10 times, and if all of these attempts fail, it will produce an error message: **errorCode.errorGeneratingDiscrepancyNoteId**.

The randomly-generated ID will also have **CDN_** or **DN_** appended to its beginning. In imports where these IDs are system-generated, we expect to always see those pre-fixes. In imports where the user specifies the ID, it will not add those prefixes, although we advise users to include them in their IDs if possible.

The import for queries will not consider any role-specific privileges associated with queries.

A user can perform an XML import of clinical data that includes queries within ItemData nodes as defined in the sample XML template. After an upload completes successfully, the Queries imported are visible in the Queries interface and their contents match the import.

When a user imports queries, and all discrepancy notes (child or parent) within a given query in the import have a specified ID that matches an existing note's ID, that query will not be imported.

If an imported query contains a mixture of notes with existing display IDs and notes that don't already exist, if the notes with new IDs are all after notes with existing IDs in their order within the query, the new notes will be added, and the existing notes will be skipped. If the query contains new notes that appear between existing notes, the query's import will fail with an error.

After a query has been successfully imported, the Name attribute as seen in the ODM will be populated with the first and last name of the user whose username appears in the associated UserName or UserRef UserName attribute as defined in their account information.

A query that has been created via import must behave exactly the same as a query created via the UI with regard to updating, closing, and viewing it.

When a user imports an XML file including an Item without the Item Data Value attribute, if a Query is included in the Item, the import successfully imports the Query, skipping the ItemData.

When importing Queries with the IDs explicitly specified, after the import completes successfully the display IDs in the ODM/downloads for those Queries should not have the **DN_** and **CDN_** prefixes added. For example, a user-specified ID of **12345** should appear in the Queries downloads as **12345**. A specified ID of **CDN_12345** should appear in the downloads as **CDN_12345**.

When a user imports Queries and attempts to add Query notes to an existing thread with the correct IDs, but has the thread under the wrong Item, the logs should contain the duplicate thread error.

When assigning users to queries existing requirements for including users based on study-level/site-level role and Participant's location still apply. This includes the user must exist in the site or study in order for the query to import correctly.

Error Messages

- **errorCode.noteIdAlreadyInUse** - This applies when a note in the import file has an ID specified that already belongs to an existing note, whether they are in the same item or not, and whether or not they are child or parent.
- **errorCode.newQueryInBetweenOldQueries** - Error message for new notes preceding existing notes.
- **errorCode.discrepancyNoteIdTooLong** - Error message for ID over character limit of 1000. This causes the import to exclude the query.
- **errorCode.detailedNoteMissing** - Missing Detailed Note.
- **errorCode.discrepancyNoteStatusNotValid** - Invalid query status or status does not follow

query workflow.
- **errorCode.missingDiscrepancyNoteStatus** - Missing query status.
- **errorCode.discrepancyNoteTypeNotValid** - Invalid note type.
- **errorCode.missingDiscrepancyNoteType** - Missing note type.
- **errorCode.userNotValid -** Error message for invalid UserName.
- **errorCode.assignedUserNotValid** - Error message invalid UserRef UserName. This causes the import to exclude the query.
- **errorCode.missingUserName -** Error message for missing UserName. This causes the import to exclude the query.
- **errorCode.noteTextTooLong** - Error message for DetailedNote over limit.
- **errorCode.errorGeneratingDiscrepancyNoteId -** Error message for random ID generation failure**.**
- **errorCode.existingNoteIdOnOtherItem** - Error message for each duplicate note.

Query Statuses for Importing

Possible statuses are: New, Updated, Closed, and Closed-Modified (Closed-Modified means the Query was closed and the data in the Item has since changed.)

Basic Logic:

- First Note must be New and only first Note can be New
- If current Note is New > next Note can be Updated, Closed
- If current Note is Updated > next Note can be Updated, Closed
- If current Note is Closed > next Note can be Updated or Closed-Modified
- If current Note is Closed-Modified, next Note can be Updated or Closed

Examples of valid note status orders in Queries:

- New (single note)
- New > Closed
- New > Updated
- New > Updated > Closed
- New > Updated > Closed > Updated > Closed
- New > Updated > Closed > Updated
- New > Updated > Closed > Closed-Modified
- New > Closed > Closed-Modified
- New > Updated > Updated > Updated

Examples of invalid note status orders in Queries:

- Any status other than New in first note
- New > Closed > Closed
- New > Closed-Modified > Closed-Modified
- Only Closed can transition to Closed-Modified
- New not in first note

Functional approval by Kate Lambert. Signed on 2024-11-07 10:39AM

Approved for publication by Paul Bowen. Signed on 2025-02-10 1:28PM

15.2 Importing Tabular Data

You can import data on the **Import Tabular CRF Data** screen. To do so, you need a Tabular Data file that contains your data and a Data Mapping file that maps your Tabular Data file columns to Form items.

When you import a flat file, each data row imports into the Form in a different Event occurrence for the specified Participant.

If you import into a Common Event, new event occurrences are created as needed. If you import into a Visit-Based Event, the data imports into existing scheduled Event occurrences.

Imports appear on the **Bulk Actions Log**. You can also download the log in a .zip file that contains 5 files.

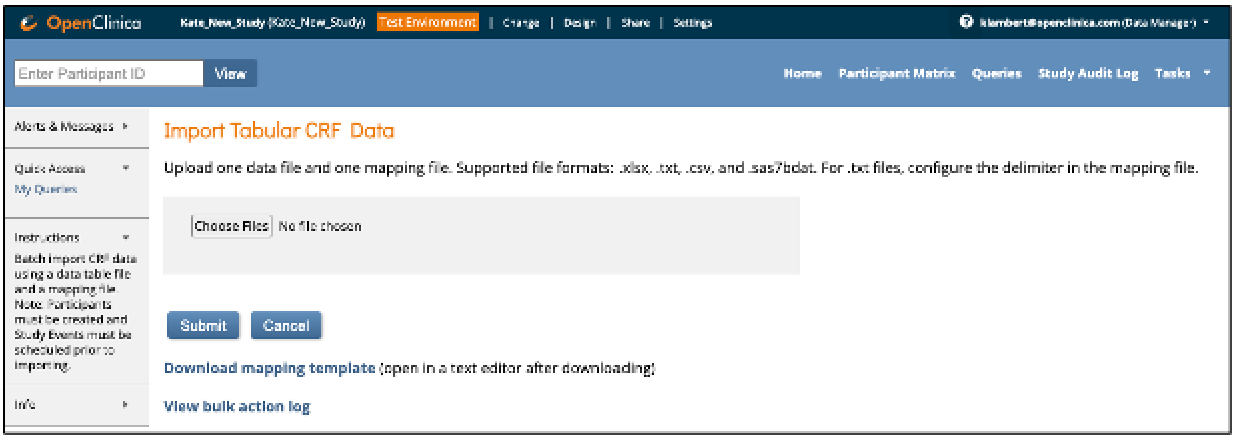
Note: *The flat file import feature does not accept Participant OIDs, it Participant IDs. These are the user-defined IDs that identify Participants in user-facing pages across the system, as opposed to the automatically generated OIDs which identify the Participants on the backend.*

- **This can be used to import any data except the following:**

- Not allowed: importing into forms in non-repeating common events containing multiple forms
- Not allowed: importing into items in repeating groups

To Import Data on the Import Tabular Data screen:

1. Download the mapping template.
2. Edit the template and save it.
3. Create a Tabular Data file.
4. Click the **Browse** button.
5. Select your Data Mapping file and Tabular Data file together by pressing the **Ctrl** key (Windows) or **Command** key (Mac) and clicking on both files.
6. Click the **Open** button.
7. Click **Submit**.



Import Tabular CRF Data

Upload one data file and one mapping file. Supported file formats: .xlsx, .txt, .csv, and .sas7bdat. For .txt files, configure the delimiter in the mapping file.

Batch import CRF data using a data table file and a mapping file. Note: Participants must be created and Study Events must be scheduled prior to importing.

Download mapping template (open in a text editor after downloading)

View bulk action log

You can click the **Tasks** button and select **Bulk Actions Log** to check the status of the job. You can view, download, or delete the import from there.

Tabular Data File

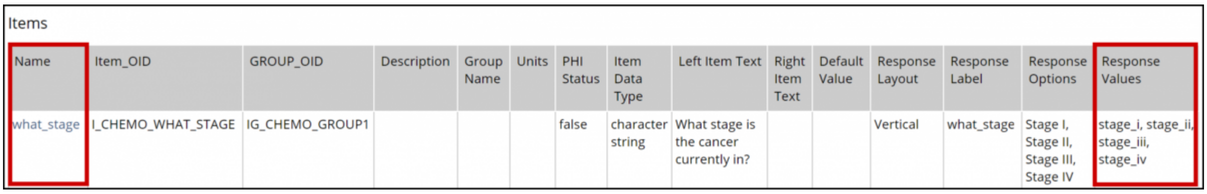
Use this file to specify Participant IDs for the Participants with data you want to import. You can also include other data items with OIDs.

Note: The following formats are supported: text (**.txt**), CSV (**.csv**), Excel (**.xlsx**), and SAS datasets (**.sas7bdat**). If you use a text (**.txt**) file, you can use a delimiter of your choice, for example a comma or tab. If no delimiter is specified, the import will use a pipe (**|**) by default. You can specify the delimiter in your Data Mapping File, for example: **#Delimiter=t**. Note that Excel automatically formats data in a useable format, but if your data comes from another source, you might need to put the data within each column in quotes if it contains commas or other delimiter values. For example, if the data is **a | b | c | d,e,f | g**, "d,e,f" would need to be in quotes due to the commas within the data.

To Make a Tabular Data File:

1. Create a tabular file as a text (**.txt**) document or use another supported format (see the note above). Alternately, download the sample file to edit.
2. Include a column for Participant IDs. If not configured differently, the heading must be **ParticipantID**.
3. Enter Participant IDs.
4. Create columns for associated data for each Participant, separated by pipe characters or, if you have created a text (**.txt**) file, another delimiter of your choice. Any spaces before or after item values are removed upon import, so the data entered in the system will not include spaces that are in the beginning or end of the file. Column headers should match those in the Data Mapping file.

Note: *Make sure to use the internal item name and response values. For example, in the image below, use **what_stage** for the item name and **stage_i**, **stage_ii**, **stage_iii**, and **stage_iv** for the responses.*



Items

Name	Item_OID	GROUP_OID	Description	Group Name	Units	PHI Status	Item Data Type	Left Item Text	Right Item Text	Default Value	Response Layout	Response Label	Response Options	Response Values
what_stage	I_CHEMO_WHAT_STAGE	IG_CHEMO_GROUP1				false	character string	What stage is the cancer currently in?			Vertical	what_stage	Stage I, Stage II, Stage III, Stage IV	stage_i, stage_ii, stage_iii, stage_iv

Data Mapping File

You can download the current Data Mapping file template on the Import Tabular CRF Data screen. The components are described below. There are also instructions included in the template in lines beginning with #.

Use this file to map each item to a column header from the Tabular Data file. To reference an item, use the format **itemgroupOID.itemOID**,

Required Configurations:

1. Under the heading, **# Study, Study Event and Form OIDs**, enter values in the following in

rows:

StudyOID=

StudyEventOID=

FormOID=

FormVersion=

Note: *You can only include one Form version in a Data Mapping file. To include another version in the import, you must create a new Data Mapping file and Tabular Data file. Each pair of files must be imported separately.*

2. Next to the first three rows, enter the appropriate OID.
3. Next to FormVersion, enter the name of the Form version.
4. Under the heading, **# Data file OIDs**, enter a row for the name of each column header in your Tabular Data file, followed by the corresponding Item Group OID and Item OID in the format of itemgroupOID.itemOID, where itemgroupOID and itemOID are the OIDs defined in the study for the item being mapped from the data file column. For example, to import an item under the column header, **Ethnicity**, enter **Ethnicity=IG_PHYSI_GRP3.I_PHYSI_PETHNIC**.

Optional Configurations:

You can specify which column delimiter is used in the Tabular Data File by setting the **Delimiter** value, for example, **Delimiter=|**.

You can use **Match Criteria** and **MatchAction** in the Data Mapping file to skip importing specific data or update specific data when the data in a Participant's Form matches the specified criteria. To use **Match Criteria**, you must have created Participants prior to the import.

If you want to use Skip/Match Criteria, enter **SkipMatchCriteria=itemgroupOID.itemOID**, where **itemgroupOID** and **itemOID** are the OIDs for the item defined in the study for the item used as part of the criteria. For example, to use **First Name** as criteria, enter **SkipMatchCriteria=IG_PHYSI_GRP2.I_PHYSI_FNAME**.

a. You can enter multiple items to use as your criteria by listing them separated by commas. For example, **SkipMatchCriteria=IG_PHYSI_GRP2.I_PHYSI_FNAME,IG_PHYSI_GRP2.I_PHYSI_LNAME.**
b. You can also use **ParticipantID** or **EventRepeatKey** as match criteria by including the header for that column you've specified in the comma-separated criteria list.

- If the Skip action has been specified and there is a match for all the criteria, that row is skipped.
- If the Update action has been specified and there is a match for all criteria, the existing record is updated to match what is in your data file.
- If there is no match or Skip/Match criteria are not used, the data for that row is imported.

You can elect to ignore columns in your Tabular Data File that are not referenced in your mapping file by setting the **IgnoreUnmappedColumns=yes**. If you do not use this option and your Tabular Data File has columns that are not mapped, your import will fail.

You can use the **FormWorkflowStatus** field to define a final status for your forms after import is finished. You can choose a status of **Initial Data Entry** or **Data Entry Complete**.

You can specify a **Reason for Change** in your import for any data changed due to Match Update. The default value is **Update via Import**.

After entering all configurations, save the file with the extension ".properties".

Note: *To make a* ***.properties*** *file, click* ***File****, and select* ***Save As****. Then change* ***Save as Type*** *to* ***All Files****, and add* ***.properties*** *to the file name. If you previously saved it as a text file, you might need to rename it to change the file type.*

The Log File

- Is generated with the name **datafilename_log.txt** (The system replaces "datafileneame" with the name of your Tabular Data file.)
- If a data file with the same name is used again, the existing log file from a previous upload is appended to the new log file.
- Contains the following:
    - **Row number:** The row number of the record in the Tabular Data file
    - **ParticipantID:** The value of the Participant ID in the Tabular Data file
    - **Status:** The outcome of the import indicated by the values **Completed**, **Failed**, or **Completed with Errors**
    - Message:
        - For a failed import, displays an error code.
        - For a successful import, displays the values imported or skipped.

Download the Log File:

Once the import is complete, click the **Download** button in the **Actions** column.

Common Errors

- Spaces in the Tabular Data or Data Mapping Files
- Invalid OID
- Items spelled differently in Data Test and Data Map File
- Mismatched number of columns in the Tabular Data File and entries in the Data Mapping File
- Items not in the correct format of itemgroupOID.itemOID
- Items listed by item name, not internal item name (variable)
- A status of **Completed with Errors** will be assigned if the job completed with any row failures. Rows that did not have errors will have imported successfully with this status.

Common Error Codes and Messages

Error Code	Message
errorCode.ValidationFailed	This is not the correct response set.
errorCode.ValidationFailed	You have a date value which is not in ISO 8601 format at the (Form OID) Check it and try again
errorCode.ValidationFailed	This field should be a number.
errorCode.ValidationFailed	The Event CRF you are trying to update does not have the correct status.

errorCode.ValidationFailed	Your CRF Version (Form OID) did not generate any results in the database. Check it and try again.
errorCode.ValidationFailed	Your Participant OID (Participant OID) does not reference an existing Participant in the study.
errorCode.ValidationFailed	You do not have a user role set up for (user role) in study (Study OID)
errorCode.ValidationFailed	Your Study Event (Event OID) for Participant OID (Participant OID) does not reference an existing Study Event in the Study.
errorCode.ValidationFailed	The Item Group (Item Group OID) did not generate any results in the database. Check it and try again.
errorCode.ValidationFailed	Your Form Layout OID (Form Layout OID) for Form OID (Form OID) does not reference a proper Form Layout in the Form.
errorCode.dataRowMissingPipe	Error-data file format not match header - less pipe than header
errorCode.Exception	Error processing data import request.

Functional approval by Kate Lambert. Signed on 2024-11-07 10:43AM

Approved for publication by Paul Bowen. Signed on 2025-02-10 1:20PM