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:

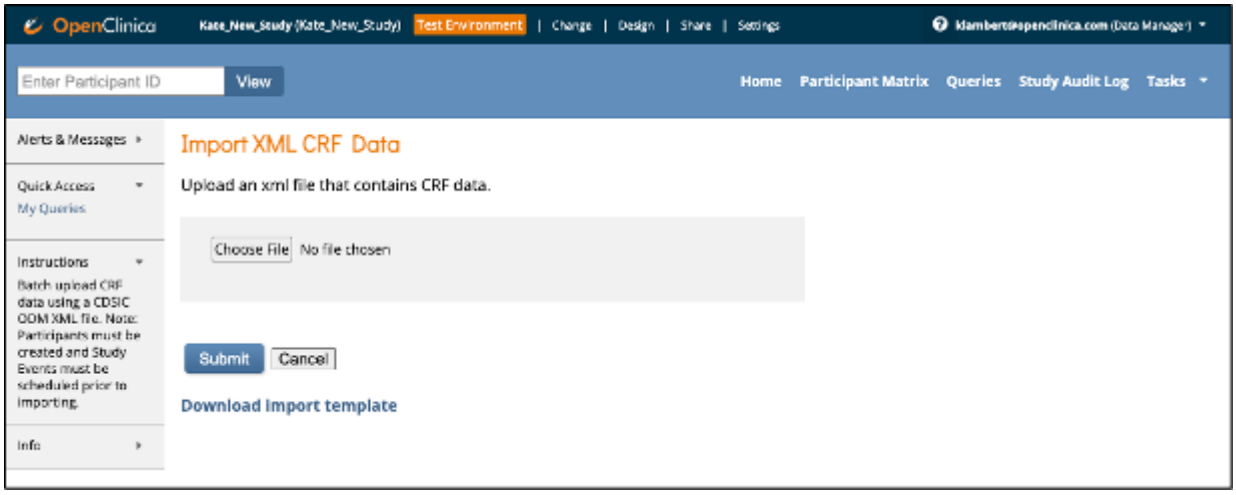
- Item data
- SDV Status
- Annotation
- Query
- Signature
- Form
- Study Event
- Participant
- Item Group

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.



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.

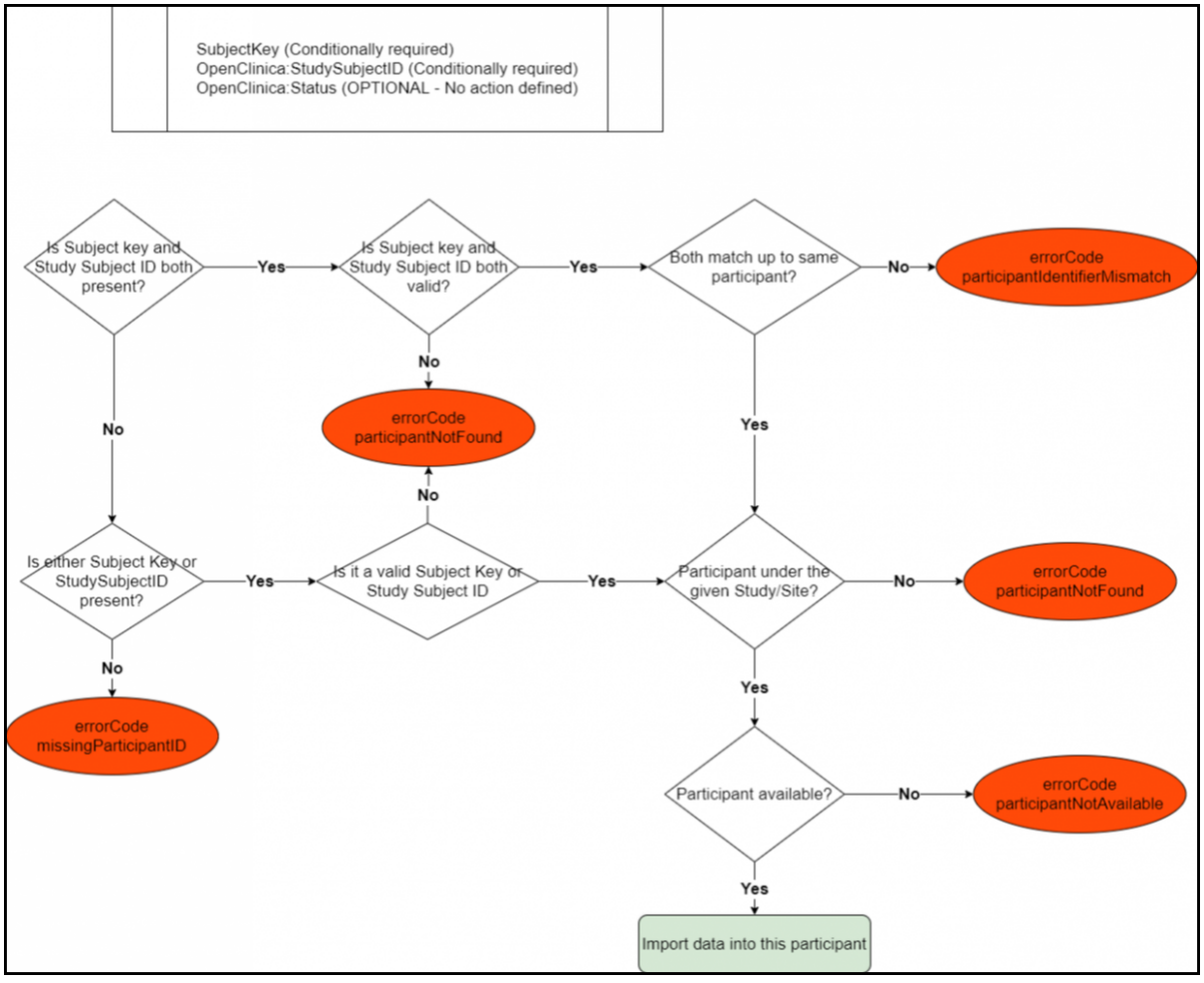
OpenClinica:StudySubjectID

- **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:



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

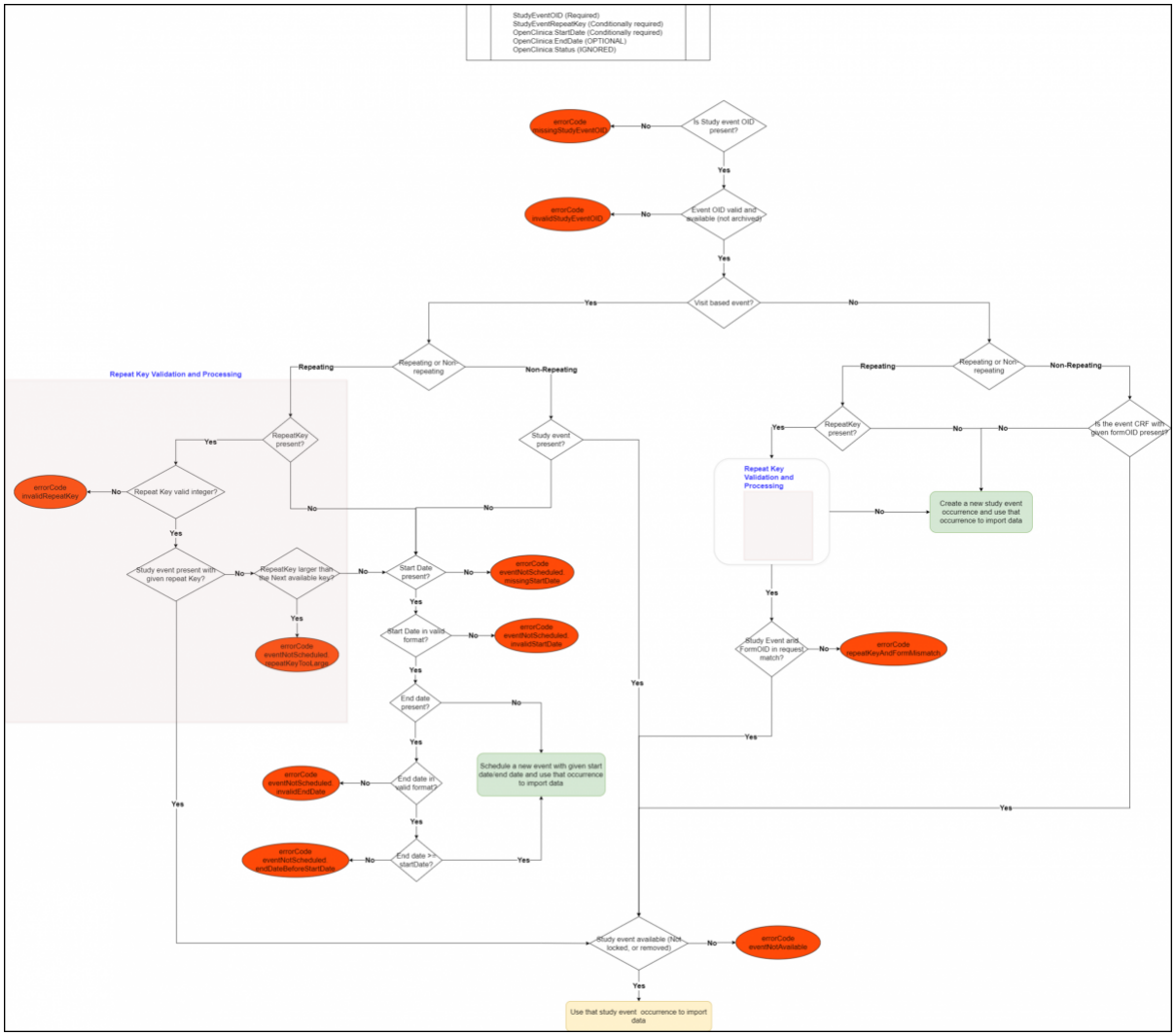
- This property represents the status of a study event.

- 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:



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

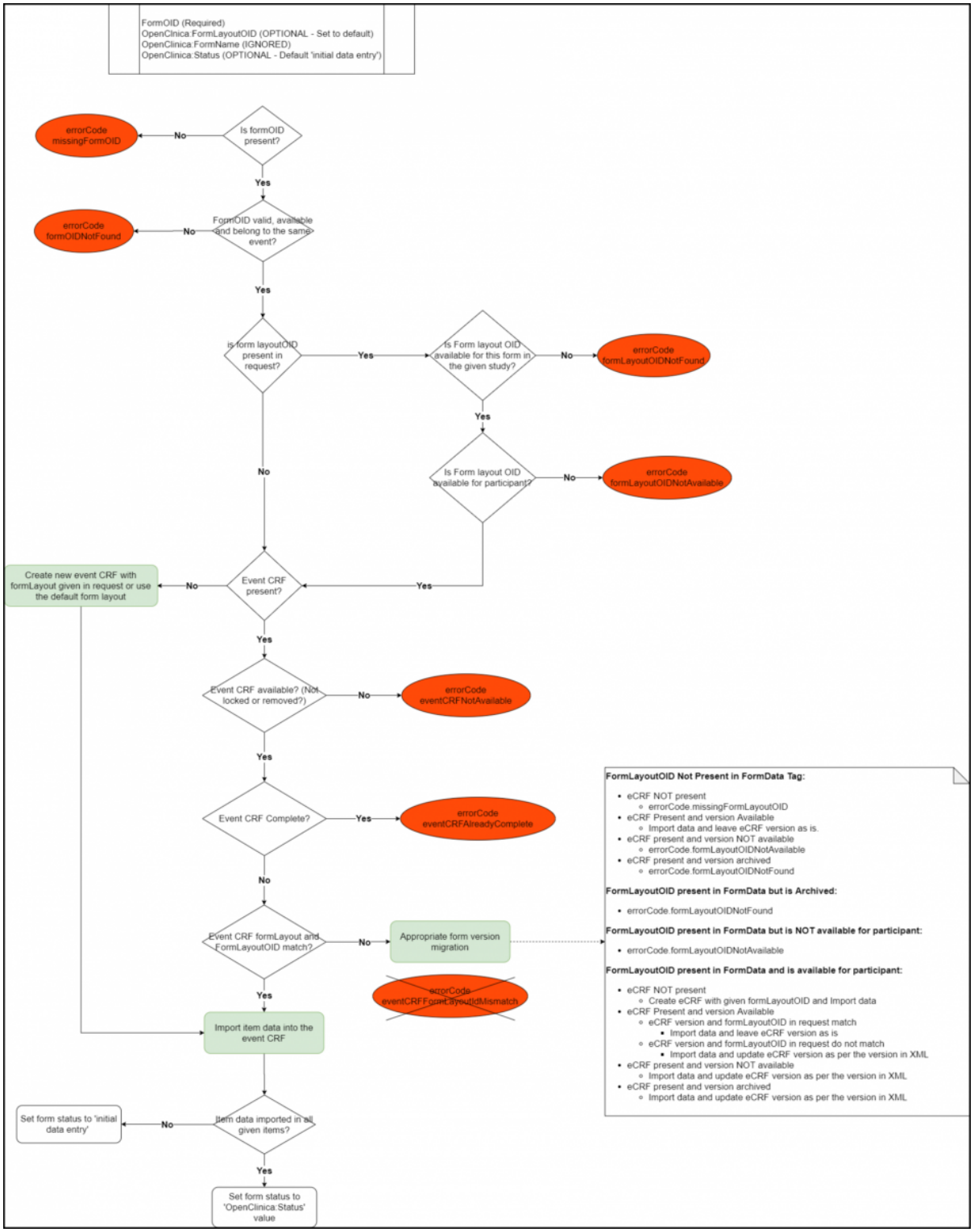
status is set as **initial data entry**; even if the provided status is **complete**.

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:



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.

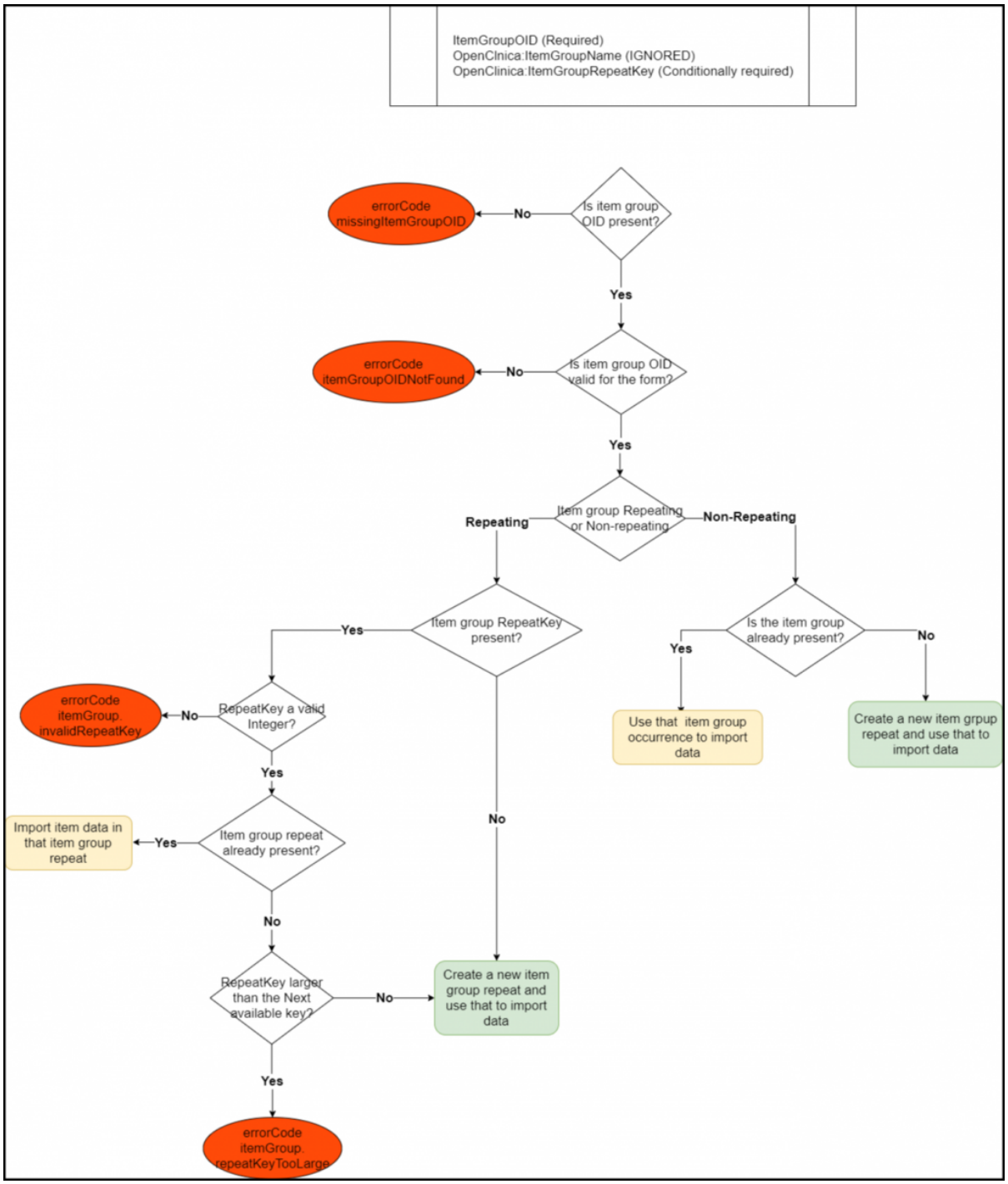
ItemGroupRepeatKey

- 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:



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

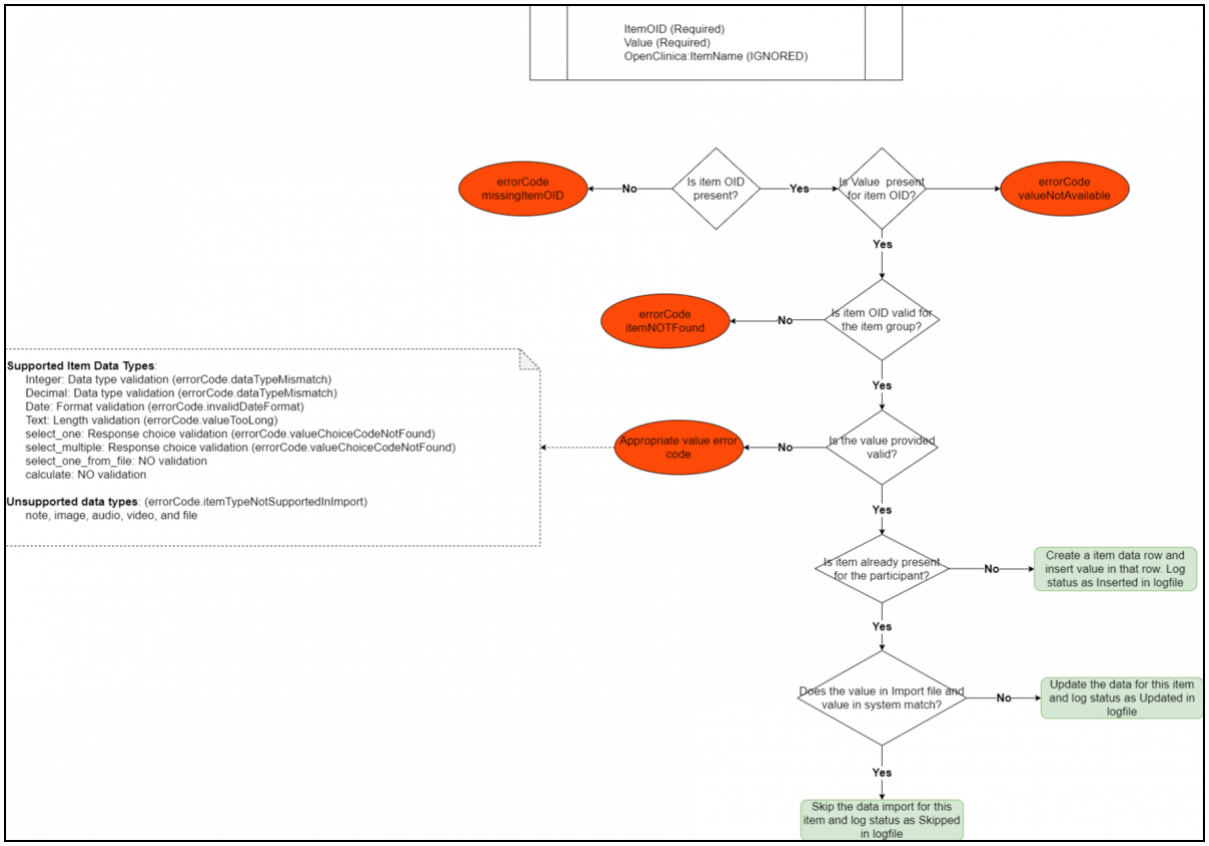
- This property represents the name of the form item.
- The value for this property is not used during the import process and has no impact on the import functionality.

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:



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

Not valid unless obtained from the OpenClinica document management system on the day of use.