



American College of Foot and Ankle Surgeons®

ACFAS NFAR Uploader Integration

Version 1.0.0

Last updated: 10/25/2024

Contents

ACFAS NFAR Uploader Integration	1
1. Introduction	3
1.1. Extracting	3
1.2. Formatting.....	3
1.3. Transmitting.....	3
2. File Types	3
2.1. The Basics – XLSX File Format	4
2.2. XLSX Samples.....	4
3. Data Types.....	4
3.1. TEXT Field Type	5
3.2. DATE and Date/Time Field Type	5

3.3.	BOOLEAN Field Type	6
3.4.	INTEGER Field Type	6
3.5.	DECIMAL Field Type	6
3.6.	DECIMALWITHUNIT Field Type	6
3.7.	SELECT Field Type.....	7
3.8.	MULTI-SELECT Field Type	7
4.	Blank Or Missing Values.....	7
5.	Queries and Form Control Logic	7
5.1.	Form Control Logic.....	7
5.2.	Edit Check (Query) Logic	8
6.	ACFAS forms, and their data elements	9
7.	Error Handling.....	10
7.1.	Hierarchy of Errors	10
7.2.	File-Level Error Checks	10
7.3.	File-Level Error Checks: General Data Content.....	11
	Change Log.....	13

1. Introduction

The goal of the Uploader integration is to provide a mechanism for American College of Foot and Ankle Surgery (ACFAS) sites to upload Uploader data to pre-populate forms in the National Foot and Ankle Registry platform. This solution will benefit sites by reducing the time required for a Registry Coordinator to manually enter data into the form and reducing the potential for data entry errors while providing a secure, consistent flow of data into the registry.

Uploader integration is the process of extracting, formatting, and transmitting data to the ACFAS NFAR Registry platform.

Uploader integration is achieved by following the Engage, Assess, Execute and Support implementation process for coordinating technical and workflow details between your site and NFAR Registry. This document provides a foundation of information about connectivity, file types, data requirements, architecture, and error handling.

1.1. Extracting

This first step of Uploader integration is the extracting of data is the responsibility of your medical center. The data elements, obtained from your current vendor or data collection system, will be uploaded in a manual process from an appropriately formatted file

1.2. Formatting

The ACFAS NFAR Registry Uploader only accepts properly formatted xlsx files. To use this time-saving functionality, we ask that your hospital create the files for uploading in a specified format provided by IQVIA in this coding manual. The ACFAS NFAR Registry Uploader will only pull those data elements specified in the file your site creates after which you may need to complete the forms manually.

1.3. Transmitting

To upload data your site must compile information in the format described in this document. It is expected that your current vendor/IS department will help you to generate the file. Transmitting data to the ACFAS NFAR Registry platform is done manually. Once the .xlsx file has been generated, you will need to log-in to the ACFAS NFAR Registry and navigate to the Upload tab.

2. File Types

Data submission methods for ACFAS NFAR Registry must be submitted in an xlsx file format. XLSX Sample files can be found in the ACFAS NFAR Uploader Samples folder of your Welcome Package.

Please refer to the ACFAS_Registry_Data_Dictionary spreadsheet in the Welcome Package for detailed specifications of the data elements and required formatting for each Encounter Report Form (CRF).

2.1. The Basics – XLSX File Format

- Excel (.xlsx) file format
- File name cannot contain any special characters. Only Alpha (A-Z), Numeric (0-9), and underscore (_) are allowed.
- Each .xlsx file cannot be more than 50MB in size
- Each .xlsx file will contain a heading row.
- The values in the heading row will be the element names separated a comma
- The order within the heading row and the data rows will match the order that is indicated for each CRF as per the XLSX samples in the Welcome package.
- Each data row will contain data for one record (a single patient, or diagnosis or encounter).
- The values in the data row will be the data value separated by a comma.

2.2. XLSX Samples

- There is a folder within the Welcome Package named 'ACFAS NFAR Uploader Samples'. Within that folder is 1 xlsx sample for every ACFAS NFAR Registry CRF available for upload. Additional module specific files may be provided as well.
- The xlsx files to be sent by sites will be in the same format as shown by the XLSX samples.
- Each of these XLSX samples contain all the CRF data elements corresponding for that form type.
- The headers in the file will indicate whether or not a field is Universal (no color), Level 1 (Green) or Level 2 (Orange).

3. Data Types

There are multiple field types that will be supported by the ACFAS Uploader. The supported field types are: BOOLEAN, DATE, DECIMAL, DECIMALWITHUNIT, INTEGER, MULTI-SELECT, SELECT and TEXT. There are also Repeat Group fields that are made up of a collection of these other types of data elements. Repeat Group information will be included in additional tabs in the file.

All data elements irrespective of data types can only accept values that conform to the constraints/formats defined for those elements within the registry configuration. *If a field value cannot be manually entered on the form due to formatting errors, then it will also be rejected in automatic uploads, and will cause the entire record to be rejected.*

Each of the data fields can be either populated with a value or left blank. If data is missing for a mandatory field of a CRF, then the entire record will be rejected. Additional information is contained in the Blank or Missing Values section below.

Below is how each of the supported field types will be handled by the UPLOADER Data Uploader.

3.1. TEXT Field Type

- Populated fields of text field type can contain any alphanumeric data.
- The maximum length of the text field is based on the maximum length of the field on the form. Any input value received that is greater than the max length allowed will be rejected.
- Please see Column F on the appropriate fields tab of the 'ACFAS_Registry_Data_Dictionary' file in the Welcome Package for max lengths.
- Data conversion is not done on the input data. For example: The text "LoWeRcAsE" is not converted by the ACFAS Uploader.
- Text values must be delimited with double quotes **IF** the value itself contains commas

3.2. DATE and Date/Time Field Type

- Dates and date/times must be provided in full MM/DD/YYYY or MM/DD/YYYY HH:MM format. MM/DD/YY format is not accepted. See table below for allowed values and system applied padding.

Precision	Allowed Date / Time Formats	System Applied Padding Outcome
Day	M/D/YYYY	0M/0D/YYYY
Day	M/DD/YYYY	0M/DD/YYYY
Day	MM/D/YYYY	MM/0D/YYYY
Year	YYYY	01/01/YYYY
Day, Minute	MM/DD/YYYY HH:MM	MM/DD/YYYY HH:MM
Day, Minute	MM/DD/YYYY HH:M	MM/DD/YYYY HH:0M
Day, Minute	MM/DD/YYYY H:MM	MM/DD/YYYY 0H:MM
Day, Minute	MM/DD/YYYY H:M	MM/DD/YYYY 0H:0M

- Leading zeros in the month and day field are optional. For example: '9/2/2012 11:45' is a valid date.
- The imported date value is not impacted by the time zone of the user that is running the EMR data Uploader or by the user that is viewing the data.
- Please see Columns E on the appropriate fields tabs of the 'ACFAS_Registry_Data_Dictionary' file included in the Welcome Package for the date value precisions allowed for each field.
- Imported date fields will use the same standard date range check that are used within the Encounter Report Form. For example: The "Date of Initial Encounter " field (diagnosisdt) on the Encounter form must be on or before the current date.

3.3. BOOLEAN Field Type

- Populated boolean fields should contain a value of TRUE (represents 'checked')

3.4. INTEGER Field Type

- Populated input integer fields will only contain integer numbers.
- Imported integer fields will use the same number range check that is used within the Encounter Report Form. For example: If we populated, the "Time to weight bearing (Weeks)" field on the Encounter form with "gg" you will see an error of "This field only accepts integer values."
- Please see Column G on the appropriate fields tab of the 'ACFAS_Registry_Data_Dictionary' file included in the Welcome Package for the numeric range allowed for each INTEGER field.

3.5. DECIMAL Field Type

- Populated input decimal fields will contain a decimal number or an integer number.
- Imported decimal fields will use the same number range and precision checks that are used within the Encounter Report Form. If we populated, the "HbA1C test result" (%) field on the Encounter form with "gg" you will see an error of "This field only accepts decimal values." Please see Columns G or (H) on the appropriate fields tab of the 'ACFAS_Registry_Data_Dictionary' file included in the Welcome Package for the numeric range and precision allowed for each DECIMAL field.
- Integer values and fields with less than the supported decimal precision are allowed in decimal fields by the EMR data Uploader. The additional precision is not added by the data Uploader. For example: If the input for " HbA1C test result" (%) " field is 5 the value will be set to 5 by the service. Additional precision is not added to convert the value to 5.00.
- Fields with more than the supported decimal precision are not allowed by the ACFAS Uploader.

3.6. DECIMALWITHUNIT Field Type

- Populated input decimalwithunit fields will contain a decimal number and a Unit measure associated with that number.
- The valid format in which data should be received is 'UnitOfMeasure@ValueInThatUnit' where the UnitOfMeasure is one of the acceptable units and ValueInThatUnit is the decimal value in that unit.
- Please see Columns G (or H) on the appropriate fields tab of the 'ACFAS_Registry_Data_Dictionary' file included in the Welcome Package for the numeric range and precision allowed for each DECIMALWITHUNIT field.
- Sample Data = 1@50

3.7. SELECT Field Type

- Populated fields with the select field type will contain the code values within the input file.
- The code values to be used for a SELECT field will be as per the registry accepted definitions.
- Please see the appropriate Codes tab of the 'ACFAS_Registry_Data_Dictionary' file included in the Welcome Package for acceptable code values (Column B) for every SELECT field, and their associated code texts (Column C) as seen on the registry. Column D shows the status of that particular code value – an inactive code value will be rejected.

3.8. MULTI-SELECT Field Type

- Populated multi-select fields can contain one or multiple code values.
- The semi-colon (;) character will be used for field separation of multiple code values.
- Please see the appropriate Codes tab of the 'ACFAS_Registry_Data_Dictionary' file included in the Welcome Package for acceptable code values (Column B) for every MULTI-SELECT field, and their associated code texts (Column C) as seen on the registry. Column D shows the status of that particular code value – an inactive code value will be rejected.
- Sample Data: for af_clinicalexamres = 1;2

4. Blank Or Missing Values

The absence of a value (1 or multiple spaces) between 2 delimiters signifies no response for that field. The field will be left unanswered if this happens. If the field for which a BLANK is received is a required field for that form, then an ERROR will be generated. If it is a non-required field, then no error or warning is generated.

5. Queries and Form Control Logic

5.1. Form Control Logic

- When a new form is being created, the form control logic will be applied against the corresponding input data values on the input data record.
- When an existing form is being updated, the form control logic will be applied against the corresponding input data values on the input data record. The data values on the existing form in ACFAS NFAR Registry are not considered when the form control logic is applied and may be overwritten (please see bullet 2 below for this scenario). The following form control logic will be applied only against the input data.

- If the input data field is populated and the corresponding input data is valid for the input data, the input values will be processed. See Example #1 related to Form Control on the Encounter form.
- If the input data field is populated and another corresponding input data is not valid, then the input value will be rejected. A form control warning will be generated if this action wipes out an existing value. See Example #2 related to Form Control on the Encounter form.

Table 1The following table shows examples of how form control logic is handled when an existing form is being updated.

Example #	Input 'Module' Value (acfas_module)	Input 'Date of surgery' Value (surgdt)	Comments
1	Ankle Arthritis (1)	09/16/2024	Inputs for acfas_module and surgdt values are updated on the existing Encounter form.
2	Ankle Fracture (2)	09/17/202424	Input acfas_module value is updated on the existing Encounter form. The input '09/17/202424' value is rejected and a WARNING will be generated if existing surgdt value is cleared.

5.2. Edit Check (Query) Logic

- When a new form is being created, the edit check logic will be applied against the corresponding input data field.
- When an existing form is being updated, the edit check logic will be applied against the corresponding input data field. The existing form in ACFAS NFAR Registry is not considered when the edit check logic is applied. The following edit check logic will be applied only against the input data.
 - If the input data is populated and the corresponding input data is valid for the input data, the input values will be processed. See Example #1 related to edit check on the Encounter form.
 - If the input data is valid and another corresponding input data is BLANK when a response is expected, then the valid input data is accepted, and an ERROR generated for the invalid MISSING (BLANK) value. See Example #2 related to edit check on the Encounter form.

Table 2 The following table shows examples of how edit check logic is handled when an existing form is being updated.

Example #	Input 'Module' Value (acfas_module)	Input 'Date of surgery' Value (surgdt)	Comments
1	Ankle Arthritis (1)	09/16/2024	Input acfas_module and surgdt values are updated on the existing Encounter form.
2	Ankle Fracture (2)	BLANK	Input acfas_module value is updated on the existing Encounter form. But because the surgdt value is missing, the edit check error (for query) will be generated.

6. ACFAS forms, and their data elements

- For a detailed list of all the data elements by form, their data types and the errors/warnings associated with each data field, please refer to the Data Dictionary folder attached in the Welcome Package:
 - This data dictionary folder has a separate document for demographics and each form
 - The Fields tab shows the data element name (Column A), the field name as it appears on the registry (Column B), and field type (Column C) for each field.
- For information on how the data elements need to be ordered on the XLSX record, please refer to the XLSX samples created (within the XLSX Samples folder on the Welcome Package). The data elements per form need to be in the exact order as shown on the XLSX samples.
 - Any missing or BLANK value for a field value on the data row needs to be shown as a space between 2 delimiters (,), so that the exact count of delimiters is always accurate for that particular form.
- There are mandatory data elements for every ACFAS form in scope. These mandatory elements, if missing or invalid, will cause the entire record to be rejected. Those fields are listed below:

Form Name	Field Name	Display Label	Field Type
Demographic	dob	Date of Birth	DATE
Encounter	diagnosisdt	Date of Initial Encounter	DATE

Follow Up	fu_date	Date of follow-up	DATE
-----------	---------	-------------------	------

7. Error Handling

7.1. Hierarchy of Errors

There are several levels of errors that can occur when the input file is processed.

- Some CRITICAL errors are severe enough to cause the entire file to be rejected, such as file format errors.
- Data level errors will not reject the entire record, but will reject those particular data values that contributed to the errors, and
- Warnings that are more informational in nature.

7.2. File-Level Error Checks

The Uploader will check for validity in certain general items in each .xlsx file. Errors found will reject the file and prevent it from Uploading.

Error Message	Meaning
"File size exceeds maximum limit."	<u>File size:</u> Each .xlsx file may not exceed 50 MB
"Error Uploading: We could not upload your file because it contains unsupported characters in the file name. Please convert special characters to alphabetic, numeric, or underscore characters."	<u>File Name Issues:</u> • Unsupported characters are in the file name. Remove any special characters
"Registry is not configured or allowed to accept <insert file type> file type"	<u>File Type Issue:</u> • Only .xlsx file types are allowed for upload
"Please upload a file with data"	<u>File Issue:</u> No data with .xlsx file

Error Message	Meaning
"File does not meet .xlsx formatting requirements."	<u>Valid .xlsx formatting:</u> • Each record is on a separate line • Fields are separated with commas • Leading and trailing whitespace are not allowed • Embedded commas – Fields must be delimited with double-quotes if the field itself contains commas • Embedded double-quotes – Fields with embedded double-quote characters must be enclosed within double-quote characters, and the embedded double-quote characters must be represented by a pair of double-quote characters. • Fields should not be double quote delimited when not required
The following column headings are duplicated in your file: <duplicate column heading name>. Your file has been discarded. Ensure column headings are not duplicated then retry the file upload if necessary.	<u>Column headers:</u> If the same column heading value is duplicated 1:1 where leading and trailing spaces are not considered, then this message will fire.
"Your account is not configured with any form groups for HF. Please contact the Help Desk for further assistance."	<u>Column headers:</u> If you do not have access to any of the form groups applicable to the study (HF) into which you are Uploading (and therefore the column headers are invalid).

7.3. File-Level Error Checks: General Data Content

The Uploader will check each data element value being Uploaded to see if they are valid entries. Errors found will reject the file and prevent it from Uploading.

Error Message	Meaning
“[Insert Patient ID]: Invalid value present for data element [insert Brief Description] ([insert column header text]).” <i>If any of the patient-level file errors to the right are found</i>	<u>Text variables:</u> The Uploader will verify that the rules for a Text variable present in the file have been met for each row of patient data that is not blank under that data element column. • Check for length, types of characters allowed (as defined in the Instructions). <u>Integer variables:</u> The Uploader will verify that the rules for an Integer variable present in the file have been met for each row of patient data that is not blank under that data element column. • Check for min/max values (as defined in the Instructions), numeric characters only. <u>Decimal variables:</u> The Uploader will verify that the rules for a Decimal variable present in the file have been met for each row of patient data that is not blank under that data element column. • Check for decimal precision, min/max values (as defined in the Instructions), numeric characters only. <u>Date variables:</u> The Uploader will verify that the rules for a Date variable present in the file have been met for each row of patient data that is not blank under that data element column. • Check for precisions allowed (as defined in the Instructions), MM/DD/YYYY or MM/DD/YYYY HH:MM formatting. • If a corresponding “precision” column does not exist for a Date data element, the precision should default to “5” (MM/DD/YYYY HH:MM). • Spaces are not allowed at the end of date or date/time values. <u>multi-select variables:</u> The Uploader will verify that the rules for a multi-select variable present in the file have been met for each row of patient data that is not blank under that data element column. • Check for valid values (as defined in the Instructions), proper formatting of “[number];[number]” (i.e. multiple values are separated by semi-colons, if more than one value is being Uploaded). <u>Select variables:</u> The Uploader will verify that the rules for a Single-Select variable present in the file I met for each row of patient data that is not blank under that data element column. • Check for proper formatting, valid values (as defined in the Instructions). <u>Boolean variables:</u> The Uploader will verify that the rules for a Boolean variable present in the file have been met for each row of patient data that is not blank under that data element column. • The only valid value is “1”

Change Log

Date of Change	Version #	Change
10/25/2024	1.0	Initial release of the document.