
Validation of Edit Checks

Sjouke Huisman
Thorin B.V.

Sjouke.Huisman@Thorin.nl



Introduction

Short Summary of XS2DataFax

Edit Check Usage & Definition

Edit Check Testing



XS2DataFax

Governs all DataFax related activities

From Definition to Testing

Holds over 50 Tables on:

- Studies, Client-specific Standards
- Plates, Visits, CRF pages, Conditional Maps
- Client Specifications
- DataFax Files (Schema, DFtips, etc)
- Edit Checks

Reads/Writes to DataFax

e.g. writes all Visit Map files, Reads DFschema



XS2DataFax: Design Objectives

Enhance Quality of Study Setups

- standardization
- avoiding 'simple' errors

Make life easier

- single point of Entry
- 'let the application do the work'

Allow thorough testing

- create test data
- automate & document test procedure

Save time

- automate manual tasks



XS2DataFAX: Two Examples

Annotated CRF: reads DFschema and DFtips and 'overlays' on .MIF file

SAS test data: XS2DataFAX creates test data with unique values for each data point



Edit Checks (ECs)

XS2DataFAX holds data on:

- Each Edit Check (Key: Check_ID)
- EC test data
- EC <-> SAS Variable links
- EC 'triggers' imported from DFschema

Generates:

- EC Framework (DFedits)
- QC & QCnotes
- Testdata
- DataFAX Worksheets



EC Usage (1)

Phased Approach

Step 1: record at minimum validation level

Step 2: variable has a value (separate EC:
'EC_REQ')

Step 3: variable has a legal value
(separate EC: 'EC_LGL')

Step 4: for cross-plate ECs: record exists
+ Step 1 – 3 + No Unresolved QC note

Step 5: execute EC



EC Usage (2)

Each EC has its own ID (e.g. "DM_AGE")

- Thus EC 'knows' if existing Query is for same or other problem
- EC 'waits' until first problem on Field is resolved, then instructs user to Edit Query for other problem



EC Usage (2)

QC Tracking

- Note field used for QC tracking
- Allows to overrule QC triggering (e.g. "DM_AGE=ALLOWED")

QC Resolution

- EC instructs user when problem is resolved
- Opens QC Edit window



EC Validation

Consists of three parts:

1. Proper Translation of Data Validation Plan into EC specification ('weight between 50 and 100 kg.')
1. Proper Translation of EC specification into DFedits
1. Proper Triggering



EC Validation: Proper Triggering

- ECs (e.g. 'CM_STARTD') are linked to SAS Variables ('STARTD')
- SAS Datasets (e.g. 'CM') are linked to DataFax Plates
- Thus, 'CM_STARTD' is linked to every occurrence of DataFax variable 'STARTD'
- XS2DataFax reads DFschema and compares actual DataFax specification with pre-defined SAS specification



EC Validation: Testing

- XS2DataFax allows to create testcases
 - User Friendly way to specify values for variables
 - Indicate if Test Data should trigger a QC note (or not)
- Test Data created when EC is specified
- Individual test values are combined into DataFax data-records



EC Validation: Test Procedure (1)

1. Testdata converted into DF records
1. DFimport imports Testdata into DF
1. DFprintdb prints Testdata on CRF
1. XS2DataFax adds Bookmarks to TestCRF (= Human Readable Input)



EC Validation: Test Procedure (2)

1. XS2DataFax creates individual Batchjobs per EC
1. Executes each Batchjob
1. Reads .XML output and checks, per EC, and per Testcase, if a Query was raised (or not)
1. Compares upfront specification (QC raised or not) with actual Batch EC output
1. Converts .XML output, per EC, into .HTML (= Human Readable Output)



EC Validation: Examples

- Report: Matching ECs to DataFax items
- Report: Validating ECs (Worksheet 24)
- HTML Output



Summary

XS2DataFax allows for a testing procedure which is:

- Documented
- Verifiable
- Repeatable
- Covers the whole process from DVP to actual Queries
- Time saving

