

%xuct

Overview

The **%xuct** macro validates variable values against controlled terminology defined in both the study specifications and CDISC SDTM Controlled Terminology. It checks if variables contain only the values specified in the Codelists tab of the specification and optionally verifies compliance with CDISC SDTM Controlled Terminology.

Version Information

- Version: 1.0
- Last Updated: 12AUG2022
- Author: Atorus Research

Dependencies

- SAS Version: SAS 9.4 V9
- Required Files:
 - Codelists metadata file (default: **SDTM_spec_Codelists.csv**)
 - CDISC SDTM Controlled Terminology dataset (**_ct.sas7bdat**) in the misc library
 - If **_ct.sas7bdat** is missing, CDISC SDTM Controlled Terminology CSV file (default: **SDTM Terminology.csv**)

Parameters

- **inds** (required): Name of the input dataset to check.
- **invar** (required): Name of the variable in the input dataset to be checked.
- **codelist** (required): Codelist name as specified in the ID column of the codelists metadata file.
- **subset** (optional): Logical condition to be used in the where clause when a variable may have multiple codelists. Default: **1=1** (all observations).
- **filename** (optional): Codelists metadata file name. Default: **SDTM_spec_Codelists.csv**.
- **filepath** (optional): Path to the codelists metadata file. Default: **[project path]/final/specs**.
- **ctfile** (optional): CDISC SDTM Controlled Terminology file name (CSV format). Default: **SDTM Terminology.csv**.
- **ctpath** (optional): Path to CDISC SDTM Controlled Terminology file. Default: **[project path]/final/raw/dict**.
- **checkct** (optional): Flag determining whether values should be checked for compliance with CDISC SDTM CT. Default: Y.
- **debug** (optional): Flag determining whether to delete temporary datasets. Default: N.

Return Values/Output

- Log messages indicating:
 - Values not found in the specification's Codelists tab
 - Values not compliant with CDISC SDTM Controlled Terminology (if checkct=Y)
 - Non-printable characters found in codelist values

- Error messages for missing or invalid parameters
- Creates `_ct.sas7bdat` in the misc library if not already present

Processing Details

1. Parameter validation:
 - Checks for required parameters
 - Verifies existence of codelist metadata file
 - Checks for CDISC CT dataset or source file
2. Codelist processing:
 - Handles domain-specific codelists (e.g., EX.UNIT, LB.UNIT)
 - Imports and processes codelist metadata
 - Validates data types for codelist values
3. Data validation:
 - Checks for non-printable characters in codelist values
 - Merges specification codelists with input data
 - If checkct=Y, validates against CDISC CT
4. Error handling:
 - Reports missing or invalid parameters
 - Warns about non-compliant values
 - Identifies non-printable characters

Examples

```
/* Basic usage - check CMCAT variable against CMCAT codelist */
%xuct(cm, CMCAT, CMCAT, checkct=N);

/* Check EXDOSU against domain-specific UNIT codelist with CT validation */
%xuct(ex, EXDOSU, EX.UNIT, checkct=Y);

/* Check LBORRESU with custom subset condition */
%xuct(lb, LBORRESU, LB.UNIT, subset=%str(LBCAT='CHEMISTRY'), checkct=Y);

/* IMPORTANT: Understanding `checkct` parameter behavior:
   - When checkct=N: Values are only validated against the Codelists tab in the
   specification (e.g., SDTM_spec_Codelists.csv)
   - When checkct=Y: Values are validated against both the Codelists tab AND CDISC
   controlled terminology
   - For prefixed codelists (e.g., EX.UNIT), the prefix is removed before CDISC CT
   validation */
%xuct(ex, EXDOSU, EX.UNIT, checkct=Y); /* Will check values against UNIT in CDISC
CT */
```

Common Issues and Solutions

1. Missing Codelist in Specification

- Error: "Codelist ID = [name] was not found in Codelists tab of the spec"

- Solution: Verify codelist name and ensure it exists in the specification

2. Missing CDISC CT Dataset

- Error: "misc_ct data set does not exist"
- Solution: Ensure `ctfile` parameter points to valid CDISC CT CSV file

3. Invalid Data Types

- Error: "Unknown Data Type for [codelist] codelist"
- Solution: Use valid data types in specification (text, integer, float, etc.)

Notes and Limitations

1. For domain-specific codelists (e.g., units used across multiple domains), the codelist ID must use DOMAIN.CODELIST format (e.g., EX.UNIT, LB.UNIT).
2. The CDISC CT CSV file must use "\$" as the delimiter.
3. Non-printable characters in codelist values are detected and reported but not automatically corrected.
4. The macro processes one variable/codelist combination at a time.

See Also

- `%xufmt`: Creates SAS formats from codelists
- `%xuloadcsv`: Imports and processes CSV files
- `%xuload`: Loads datasets with format handling

Change Log

Version 1.0 (12AUG2022)

- Initial release
- Basic codelist validation functionality
- CDISC CT compliance checking
- Non-printable character detection