PharmaSUG China 2025 - Paper XX-###

A Hands-On Guide: Efficiently Managing Special Characters with a Tool for Chinese Study

Dingpan Liu, HUTCHMED

ABSTRACT

Objective: This paper presents a comprehensive methodology for efficiently identifying, categorizing, and managing special characters in clinical datasets using a systematic tool approach tailored for Chinese studies.

Background: In pharmaceutical data management, special characters present significant challenges for data integrity, system compatibility, and regulatory compliance. Traditional ad-hoc approaches to character validation often result in inconsistent handling and missed edge cases, particularly in multilingual environments such as Chinese clinical studies where diverse character sets coexist.

Methods: We developed a four-component systematic approach: (1) Creation of a comprehensive Unicode reference table in Excel using automated functions to generate all Unicode characters with corresponding codes; (2) Development of a special character management framework categorizing characters into "Allowed," "Allowed with Warning," and "Error" classifications; (3) Implementation of a SAS-based automated detection program that scans specified datasets and generates standardized Excel reports; (4) Establishment of a continuous feedback loop for character classification refinement based on project-specific requirements.

Results: The tool successfully identified and categorized special characters across multiple datasets, generating actionable reports that enabled both preventive data entry corrections and post-processing character standardization. The iterative classification system demonstrated improved accuracy over successive implementations, with the comprehensive Unicode reference serving as a reliable foundation for character validation.

Conclusions: This systematic approach provides a scalable, maintainable solution for special character management in pharmaceutical data processing. The tool's modular design allows for continuous improvement and adaptation to evolving project requirements, making it particularly valuable for long-term data management strategies in international clinical studies.

Keywords: Special Characters, Unicode, Data Management, SAS Programming, Clinical Data, Chinese Studies

INTRODUCTION

The pharmaceutical industry operates in an increasingly globalized environment where clinical trials and regulatory submissions often involve multiple languages and character sets. In Chinese clinical studies, data managers frequently encounter challenges related to special character handling, ranging from traditional and simplified Chinese characters to various symbols, punctuation marks, and non-standard Unicode entries that may compromise data integrity and system interoperability.

Special characters in pharmaceutical datasets can originate from various sources: patient-reported outcomes containing native language text, investigator comments with mixed character sets, imported data from external systems, or inadvertent keyboard entries during data collection. These characters, while sometimes legitimate and necessary for accurate data representation, can cause significant downstream issues including database import failures, statistical analysis errors, regulatory submission rejections, and cross-system data transfer problems.

Current industry practices for managing special characters typically rely on reactive, case-by-case approaches that lack standardization and comprehensive coverage. Many organizations depend on manual review processes or basic pattern-matching techniques that fail to address the full scope of Unicode complexity. This ad-hoc methodology often results in inconsistent character handling across projects, missed validation opportunities, and substantial remediation efforts during critical project phases.

The Unicode standard, encompassing over 140,000 characters across multiple language scripts, presents both opportunities and challenges for pharmaceutical data management. While Unicode enables accurate representation of global languages essential for international studies, its vast character space also introduces potential data quality risks when uncontrolled or inappropriate characters enter clinical databases.

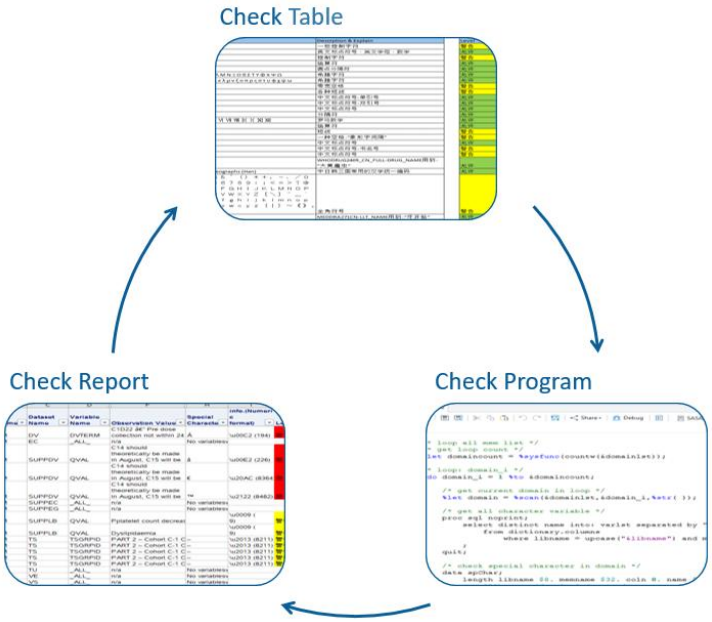
Previous literature has addressed general data cleaning methodologies and standard character validation techniques, but limited guidance exists for systematic special character management specifically tailored to pharmaceutical applications. Existing SAS-based solutions typically focus on basic ASCII validation or simple character substitution, lacking the comprehensive approach necessary for complex multilingual environments.

This paper addresses these gaps by presenting a systematic, tool-based methodology for special character management that combines comprehensive Unicode reference generation, flexible character classification frameworks, automated detection capabilities, and continuous improvement processes. Our approach recognizes that effective special character management requires not just identification and correction, but also systematic categorization, documentation, and knowledge accumulation across projects.

The methodology presented here was developed and refined through practical application in Chinese clinical studies, where the coexistence of Chinese characters, English text, and various symbols creates particularly complex validation scenarios. However, the underlying principles and technical implementation are broadly applicable to any pharmaceutical data management context requiring robust character validation and control.

By implementing this systematic approach, pharmaceutical organizations can achieve more consistent data quality outcomes, reduce remediation costs, improve regulatory compliance, and build institutional knowledge around special character management that scales across multiple projects and therapeutic areas.

PROCESS



The special character management system follows a systematic three-stage workflow designed to provide comprehensive character validation and continuous improvement capabilities. Figure illustrates the complete process flow.

SECTION 3.1: BUILDING THE CHARACTER CONTROL CHECKLIST

3.1.1 New a Unicode_Checklist.xlsx

3.1.2 New a excel sheet “UNICODE_num” as follows:

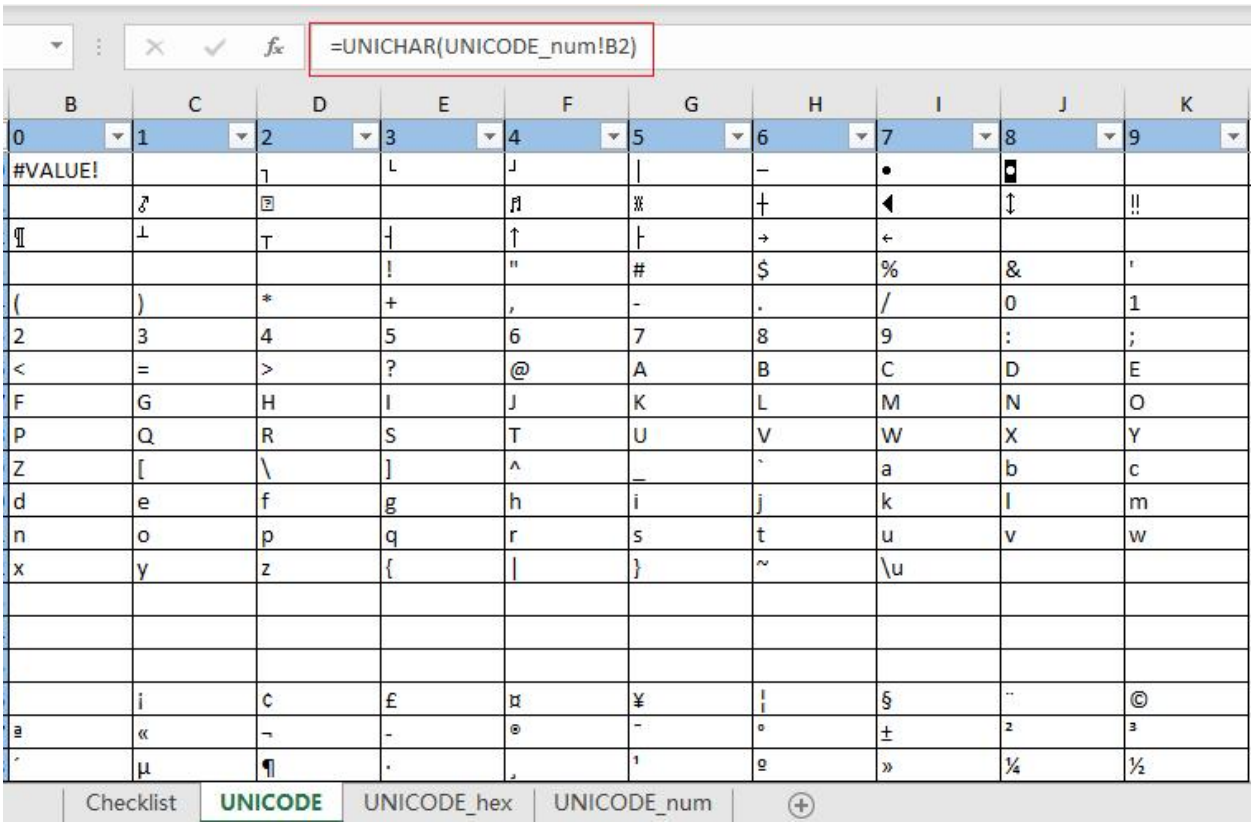
A	B	C	D	E	F	G	H	I	J	K
	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	10	11	12	13	14	15	16	17	18	19
2	20	21	22	23	24	25	26	27	28	29
3	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49
5	50	51	52	53	54	55	56	57	58	59
6	60	61	62	63	64	65	66	67	68	69
7	70	71	72	73	74	75	76	77	78	79
8	80	81	82	83	84	85	86	87	88	89
9	90	91	92	93	94	95	96	97	98	99
10	100	101	102	103	104	105	106	107	108	109
19154	191540	191541	191542	191543	191544	191545	191546	191547	191548	191549
19155	191550	191551	191552	191553	191554	191555	191556	191557	191558	191559
19156	191560	191561	191562	191563	191564	191565	191566	191567	191568	191569
19157	191570	191571	191572	191573	191574	191575	191576	191577	191578	191579
19158	191580	191581	191582	191583	191584	191585	191586	191587	191588	191589
19159	191590	191591	191592	191593	191594	191595	191596	191597	191598	191599
19160	191600	191601	191602	191603	191604	191605	191606	191607	191608	191609
19161	191610	191611	191612	191613	191614	191615	191616	191617	191618	191619

Checklist | UNICODE | UNICODE_hex | **UNICODE_num**

3.1.3 Use a function to map the values from the "UNICODE_num" to this sheet "UNICODE_hex":

B	C	D	E	F	G	H	I	J	K	L
0	1	2	3	4	5	6	7	8	9	
0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	
000A	000B	000C	000D	000E	000F	0010	0011	0012	0013	
0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	
001E	001F	0020	0021	0022	0023	0024	0025	0026	0027	
0028	0029	002A	002B	002C	002D	002E	002F	0030	0031	
0032	0033	0034	0035	0036	0037	0038	0039	003A	003B	
003C	003D	003E	003F	0040	0041	0042	0043	0044	0045	
0046	0047	0048	0049	004A	004B	004C	004D	004E	004F	
0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	
005A	005B	005C	005D	005E	005F	0060	0061	0062	0063	
0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	
006E	006F	0070	0071	0072	0073	0074	0075	0076	0077	
0078	0079	007A	007B	007C	007D	007E	007F	0080	0081	
0082	0083	0084	0085	0086	0087	0088	0089	008A	008B	
008C	008D	008E	008F	0090	0091	0092	0093	0094	0095	
0096	0097	0098	0099	009A	009B	009C	009D	009E	009F	
00A0	00A1	00A2	00A3	00A4	00A5	00A6	00A7	00A8	00A9	
00AA	00AB	00AC	00AD	00AE	00AF	00B0	00B1	00B2	00B3	
00B4	00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	

3.1.4 Use a function to map the values from the "UNICODE_num" to this sheet "UNICODE":



3.1.5 Finally, create a “Checklist” sheet to list all the characters you need to manage, just like the illustration:

A	B	C	D	E	F	G	H
ord	min	max	Unicode Character	Descrption & Explain		Level	modify
1	0	31	ASCII	一些控制字符		警告	dingpanL/2024-12-27
2	32	126	ASCII	英文标点符号、英文字母、数字		允许	dingpanL/2024-12-25
3	127	127	ASCII	控制字符		警告	dingpanL/2024-12-27
4	177	177	±	运算符		允许	dingpanL/2024-12-25
5	183	183	·	圆点分隔符		允许	dingpanL/2024-12-26
6	913	937	Α Β Γ Δ Ε Θ Ι Κ Λ Μ Ν Ξ Ο Π Σ Τ Υ Φ Χ Ψ Ω	希腊字符		允许	dingpanL/2024-12-26
7	945	969	α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ ς σ τ υ φ χ ψ ω	希腊字符		允许	dingpanL/2024-12-25
8	8203	8203		零宽空格		警告	dingpanL/2024-12-26
9	8208	8213	- - - - - —	各种短线		警告	dingpanL/2024-12-26
10	8216	8217	‘’	中文标点符号-单引号		允许	dingpanL/2024-12-25
11	8220	8221	“”	中文标点符号-双引号		允许	dingpanL/2024-12-25
12	8230	8230	…	中文标点符号		允许	dingpanL/2024-12-25
13	8231	8231	‧	分隔符		允许	dingpanL/2024-12-26
14	8544	8555	Ⅰ Ⅱ Ⅲ Ⅳ Ⅴ Ⅵ Ⅶ Ⅷ Ⅸ Ⅹ Ⅺ Ⅻ	罗马数字		允许	dingpanL/2024-12-26
15	8804	8805	≤ ≥	运算符		允许	dingpanL/2024-12-25
16	9472	9472	─	短线		警告	dingpanL/2024-12-26
17	12288	12288		一种空格-“象形字间隔”		警告	dingpanL/2024-12-26
18	12289	12290	、 。	中文标点符号		允许	dingpanL/2024-12-25
19	12298	12299	《》	中文标点符号-书名号		警告	dingpanL/2024-12-25
20	12308	12309	〔 〕	中文标点符号		警告	dingpanL/2024-12-26
21	17898	17898	蠱	WHODRUG2409_CN_FULL-DRUG_NAME用到-”大黄蠱虫“		允许	dingpanL/2024-12-26
22	19968	40959	CJK Unified Ideographs (Han)	中日韩三国常用的汉字统一编码		允许	dingpanL/2024-12-25
23	65281	65381	！ ” # $ % & ’ （） ＊ ＋ ， － ． ／ 0 1 2 3 4 5 6 7 8 9 ： ； ＜ ＝ ＞ ？ @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ［ ＼ ］ ＾ ＿ ｀ a b c d e f g h i j k l m n o p q r s t u v w x y z ｛ ｜ ｝ ～ 《 》 。「 」 、 ・	全角符号		警告	dingpanL/2024-12-26
24	181015	181015	骆	MEDDRA271CN-LLT_NAME用到-“牙开骆“		允许	dingpanL/2024-12-26

J	K	L
允许	警告	拒绝
		默认会拒绝除警告/允许之外的UNICODE字符

SECTION 3.2: IMPLEMENTING THE AUTOMATED DETECTION PROGRAM

3.2.1 Create table a SAS code CheckSpecialChar.sas

Please refer to the appendix for the specific code.

SECTION 3.3: GENERATING AND UTILIZING VALIDATION REPORTS

This section describes the development and deployment of the SAS-based automated scanning program. The program systematically processes target datasets, applies character encoding detection algorithms, performs classification matching against the control checklist, and prepares structured output for report generation.

Obs	Libname	Dataset Name	Variable Name	Observation Reco	Observation Value	Position in OBS Ind	Special Character	Unicode info.(Numeric format)	Level
148	SDTM	SUPPDM	_ALL_		n/a		No variableswith specialcharacter		
149	SDTM	SUPPDS	_ALL_		n/a		No variableswith specialcharacter		
150	SDTM	SUPPDV	QVAL	523	3 visit on 10JUL24 (the C14 should theoretically be made in August, C15 will be done in September after patientâ€™s holidays)	112	â	\u00E2 (226)	拒絶
151	SDTM	SUPPDV	QVAL	523	3 visit on 10JUL24 (the C14 should theoretically be made in August, C15 will be done in September after patientâ€™s holidays)	113	€	\u20AC (8364)	拒絶
152	SDTM	SUPPDV	QVAL	523	3 visit on 10JUL24 (the C14 should theoretically be made in August, C15 will be done in September after patientâ€™s holidays)	114	™	\u2122 (8482)	拒絶
153	SDTM	SUPPEC	_ALL_		n/a		No variableswith specialcharacter		
154	SDTM	SUPPEG	_ALL_		n/a		No variableswith specialcharacter		
155	SDTM	SUPPLB	QVAL	8290	Pplatelet count decreased	26		\u0009 (9)	拒絶
156	SDTM	SUPPLB	QVAL	15153	Dyslipidaemia	14		\u0009 (9)	拒絶
157	SDTM	SUPPMB	_ALL_		n/a		No variableswith specialcharacter		
158	SDTM	SUPPMH	_ALL_		n/a		No variableswith specialcharacter		
173	SDTM	TI	_ALL_		n/a		No variableswith specialcharacter		
174	SDTM	TR	_ALL_		n/a		No variableswith specialcharacter		
175	SDTM	TS	TSGRPID	39	PART 2 – Cohort C-1 Only	8	–	\u2013 (8211)	
176	SDTM	TS	TSGRPID	43	PART 2 – Cohort C-1 Only	8	–	\u2013 (8211)	
177	SDTM	TS	TSGRPID	49	PART 2 – Cohort C-1 Only	8	–	\u2013 (8211)	
178	SDTM	TS	TSGRPID	61	PART 2 – Cohort C-1 Only	8	–	\u2013 (8211)	
179	SDTM	TS	TSGRPID	62	PART 2 – Cohort C-1 Only	8	–	\u2013 (8211)	
180	SDTM	TS	TSVAL	50	Response assessed using Response Evaluation Criteria in Solid Tumors (RECIST V1.1). Response is evaluated with the use of MRI or CT scan [Time frame: baseline, every 8 weeks ± 7 days	175	±	\u00B1 (177)	警告
181	SDTM	TS	TSVAL1	50	during the first 24 weeks, then every 12 weeks ± 7 days thereafter, up until disease progression]	48	±	\u00B1 (177)	警告
182	SDTM	TU	_ALL_		n/a		No variableswith specialcharacter		
183	SDTM	TV	TVSTRL	67	Every 12 weeks (+/-14 days) from EOT visit …	45	…	\u2026 (8230)	警告
184	SDTM	TV	TVSTRL	68	Every 12 weeks (+/-14 days) for a maximum of 2 years from EOT visit …	69	…	\u2026 (8230)	警告
185	SDTM	VE	_ALL_		n/a		No variableswith specialcharacter		
186	SDTM	VS	_ALL_		n/a		No variableswith specialcharacter		

CONTINUOUS IMPROVEMENT AND FEEDBACK LOOP

3.4.1 Iterative Refinement Process

The special character management system is designed as a self-improving framework that evolves through successive implementation cycles. Each validation exercise provides valuable insights that enhance the accuracy and relevance of the character control checklist, creating a progressively more sophisticated validation capability.

The feedback loop operates on multiple levels: immediate tactical adjustments based on individual project findings, strategic refinements informed by cross-project patterns, and methodological improvements derived from user experience and system performance analysis.

3.4.2 Classification Optimization

Character classification accuracy improves through systematic review of validation results. Characters initially flagged as errors may be reclassified as acceptable based on legitimate use cases discovered during data review. Conversely, characters previously considered acceptable may be moved to warning or error categories when found to cause downstream processing issues.

This dynamic classification approach recognizes that character acceptability is context-dependent and may vary based on data destination systems, regulatory requirements, and organizational standards. The system maintains classification history to support decision traceability and enable rollback capabilities when needed.

3.4.3 Knowledge Accumulation Strategy

Each project implementation contributes to an expanding organizational knowledge base regarding special character patterns, common data quality issues, and effective remediation strategies. This accumulated knowledge is systematically captured through:

Pattern Documentation: Recording recurring character combinations and their typical sources

Resolution Libraries: Maintaining proven remediation approaches for common character issues

Stakeholder Feedback: Incorporating insights from data managers, statisticians, and regulatory affairs teams

System Compatibility Matrices: Documenting character behavior across different platforms and applications

3.4.4 Scalability and Adaptation

The modular design enables adaptation to evolving project requirements and organizational standards. New character categories can be added to accommodate emerging data types, regulatory changes, or system upgrades. The classification framework scales effectively across therapeutic areas, geographic regions, and study phases while maintaining consistency in core validation principles.

Cross-project standardization is achieved through centralized checklist management, while project-specific adaptations are supported through configurable classification overlays that extend rather than replace the core framework.

3.4.5 Performance Metrics and Quality Indicators

System effectiveness is monitored through key performance indicators including detection accuracy rates, false positive reduction over time, user acceptance metrics, and processing efficiency improvements. These metrics guide system refinements and demonstrate value delivery to organizational stakeholders.

Regular performance reviews identify optimization opportunities in both technical implementation and operational procedures, ensuring the system continues to meet evolving organizational needs while maintaining high standards of data quality assurance.

CONCLUSION

This paper presents a systematic methodology for managing special characters in pharmaceutical datasets through a three-stage workflow: character control checklist creation, automated detection programming, and validation report generation. The approach addresses critical data quality challenges in multilingual environments, particularly Chinese clinical studies where diverse character sets coexist.

The key innovation lies in the continuous feedback loop that enables organizations to build institutional knowledge and develop increasingly sophisticated validation capabilities over time. The SAS-based automated detection program ensures consistent validation across large datasets while reducing manual effort, and the structured reporting format facilitates both immediate remediation and strategic planning.

Implementation results demonstrate effectiveness in identifying character-related issues before they impact downstream processes. The modular design ensures scalability across different therapeutic areas and organizational contexts while maintaining validation consistency.

This methodology provides a practical solution for special character validation, enabling organizations to achieve improved data quality outcomes, reduced remediation costs, and enhanced regulatory compliance across clinical development programs.

APPENDIX

```
%macro CheckSpecialChar(libname=,memname=,checklist=,output=);
    /* defensive for parameter */
    /* %if - %return or %about;*/
    /* ... */
    /* pass and goon */

    /* import checklist*/
    proc import
        datafile="&checklist."
        out=unicode_checklist
        dbms=xlsx
        replace;

        sheet="Checklist";
    run;

    /* get checklist */
    %let passRange = -1;
    proc sql noprint;
       select catx(":", min, max) into: passRange separated by " " from unicode_checklist
where level='允许';
       select catx(":", min, max) into: warnRange separated by " " from unicode_checklist
where level='警告';
       select catx(":", min, max) into: erroRange separated by " " from unicode_checklist
where level='拒绝';
    quit;
```

```
    /* when domainlst = _all_ then get current libname all dataset */
    %if (&memname = _all_)
    %then %do;

        proc sql noprint;
            select distinct memname into: domainlst separated by " "
                from dictionary.columns
                    where libname = upcase("&libname")
            ;
        quit;

    %end;
    %else %do;
        %let domainlst = &memname;
    %end;

    /* set variable & label for output */
    data resChar;
        length libname $8. memname $32. coln 8. name $32. record 8. value $5000. index 8.
info infouni $5000. level $8.;
        label libname = "Libname"
            memname = "Dataset Name"
            index = "Position in OBS Index"
            info = "Special Character"
            infouni = "Unicode info.(Numeric format)"
            record = "Observation Record"
            name = "Variable Name"
            value = "Observation Value"
            level = "Level"
        ;
        call missing(of _all_);
        stop;
    run;

    /* loop all mem list */
    /* get loop count */
    %let domaincount = %sysfunc(countw(&domainlst));

    /* loop: domain_i */
    %do domain_i = 1 %to &domaincount;

        /* get current domain in loop */
        %let domain = %scan(&domainlst,&domain_i,%str( ));

        /* get all character variable */
        proc sql noprint;
            select distinct name into: varlst separated by " "
                from dictionary.columns
                    where libname = upcase("&libname") and memname = upcase("&domain.") and
prxmatch("/char|c/i",type)
            ;
        quit;

        /* check special character in domain */
        data spChar;
            length libname $8. memname $32. coln 8. name $32. record 8. value $5000. index
8. info infouni $5000. level $8.;
            set &libname..&domain indsname=dssource end=eof;

            /* loop1.check all variable in domain */
```

```
            array mychar &varlst.;

            do iii=1 to dim(mychar);
                call missing(libname, memname, coln, name, record, value, info, infouni,
index, level);

                if klength(mychar{iii}) ^= 0
                    then do;
                    libname=upcase(scan(dssource,1,'.'));
                    memname=upcase(scan(dssource,2,'.'));
                    name=upcase(vname(mychar{iii}));
                    coln=put(iii, best. -l);
                    value=mychar{iii};
                    record=_n_;

                    /* loop2.check all character in variable */
                    do i = 1 to klength(mychar{iii});
                        tmp = ksubstr(mychar{iii},i,1);

                        /* check for ASCII */
                        if ^prxmatch("/\\u[0123456789ABCDEF]+/", unicodec(tmp)) or
unicodec(tmp) = '\u007F' then do;
                            if rank(tmp) not in (&passRange.) then do;
                                info = catt(tmp);
                                index = put(i,best. -l);
                                infouni = catt("\u", put(rank(tmp), hex4.), " (",
put(rank(tmp),best.), ")");

                                if rank(tmp) in (&warnRange.) then
                                    level = '警告';
                                if rank(tmp) in (&erroRange.) then
                                    level = '拒绝';

                                output;
                            end;
                        end;

                        /* check for UNICODE */
                        else do;
                                                    if
input(compress(unicodec(tmp,"ncr"),"&#;"),best.) not in (&passRange.) then
                                                    do;
                                                            info = catt(tmp);
                                                            index = put(i,best. -l);
                                                            infouni = catt(unicodec(tmp), "
(", compress(unicodec(tmp,"ncr"),"&#;"), ")");

                                if rank(tmp) in (&warnRange.) then
                                    level = '警告';
                                if rank(tmp) in (&erroRange.) then
                                    level = '拒绝';

                                                            output;
                                                    end;
                        end;
                    end;
                end;
            end;

            keep libname memname coln name record value info index infouni;
```

```
        run;

        %let nobs = 0;

        proc sql noprint;
            select count(*) into: nobs
                from spChar
            ;
        quit;

        %if &nobs.=0 %then %do;

            data spChar;
                length libname $8. memname $32. coln 8. name $32. record 8. value $5000.
index 8. info infouni $5000.;
                libname=upcase("&libname.");
                memname=upcase("&domain.");
                name = "_ALL_";
                value = "n/a";
                info = 'No variables with special character';
                output;
            run;

        %end;

        data resChar;
            set resChar spChar;
        run;

    %end;

    ***loop end;
    proc sort data = resChar;
        by libname memname coln name record index;
    run;

    ods excel file="&output."
        options(frozen_headers='on' autofilter='all' sheet_name="sheet1");

    proc print data = resChar(drop=coln) label;
    run;

    ods excel close;

    /*proc export data=resChar(drop=coln) outfile="&output." dbms=xlsx replace;*/
    /*sheet="sheet1";*/
    /*quit;*/
    proc sql;
        drop table
            spChar, resChar, unicode_checklist;
    quit;

%mend CheckSpecialChar;

***parameter: memname
1._all_    (all data)
2.dm       (single data)
3.dm se sv (multiple data sepearated by spece)
***;
%let exlList = %str(\\10.13.23.101\biometrics\Project\306\2020-306-
GLOB2\Prog\Final\Dev\Documents\Specs\SDTM\Compliance\UNICODE_Checklist.xlsx);;
```

```
%let outFile = %str(\\10.13.23.101\biometrics\Project\306\2020-306-
GLOB2\Prog\Final\Dev\Documents\Specs\SDTM\Compliance\rawdata_check20250520.xlsx);

%CheckSpecialChar(libname=rawdata,
    memname=_all_,
    checklist=&exlList.,
    output=&outFile.;
    );
```

ACKNOWLEDGMENTS

The author wishes to express sincere gratitude to the HUTCHMED statistical programming team for their valuable insights and practical feedback during the development and testing of this methodology. Their expertise in Chinese clinical trial data processing was instrumental in refining the character classification framework and validation approaches.

Special appreciation is extended to team members who contributed real-world datasets and use cases that enhanced the robustness of the detection algorithms. Their constructive suggestions during the iterative refinement process were invaluable.

Thanks are due to the PharmaSUG China organizing committee for providing a platform to share this methodology with the broader pharmaceutical programming community, and to the open-source communities whose work in Unicode standardization and SAS programming techniques provided essential technical foundations for this project.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dingpan Liu
Hutch-med(Shanghai) Co., Ltd
+86 15651732206
dingpanl@hutch-med.com
HUTCHMED 和记黄埔医药（上海）医药有限公司 (hutch-med.com)