PharmaSUG China 2021 - Paper AD-032

A Submission Package Translation Toolkit by SAS and Excel VBA

Shijia Wang, Tigermed Co., Ltd.

ABSTRACT

Chinese regulatory agencies (National Medical Products Administration, NMPA) released the Guideline on the Submission of Clinical Trial Data in 2020, which clarified specific requirements of data submission in China. Many studies in non-Chinese language need to be translated based on this requirement.

In the translation process, we often meet some difficulties, such as garbled code, consistency check between multiple documents in a project, and consistency implementation in multiple studies. With more than 40 translation experience projects, we summarized a set of execution procedures, and developed an effective toolkit to implement the studies by SAS and Excel VBA. This paper will introduce our implementation of submission package translation codes and shared the key parts of SAS and VBA.

INTRODUCTION

In July 2020, NMPA released Guideline on the Submission of Clinical Trial Data, which was officially implemented on October 1, 2020. This guideline specified the detailed requirements for data standards in the submission, the contents in the submission package and the requirements that each sponsor should follow, especially for the submission package in foreign language. Once the guideline published, we began to discuss and develop tools and processes for translate submission package for NMPA and we have completed more than 40 translate projects until now. This paper describes the basic requirements for foreign submission package, summarizes common problems in hands-on projects and tools and processes we used to overcome them.

NMPA'S REQUIREMENT FOR SUBMISSION PACKAGE

In NMPA’s guideline, specific requirements are listed. The essential documents and requirements are as follow:

- aCRF: Annotated Case Report Form, specific description of the mapping relationship between the information unit (i.e., field information) of collected subject data (electronic or paper) and the corresponding variable or variable value in the submitted original data set on the basis of blank CRF. For foreign language study, the description of the question designed for data collection; the values or codelist for the questions involving the efficacy indicators should be in Chinese.
- Original/Analysis Datasets: SDTM/ADaM can be considered as the Original/Analysis Datasets if sponsor submits data according to CDISC standards. For foreign language study, dataset label and variable label, name of adverse event, name of concomitant drug and name of medical history in the CSR should be Chinese.
- Define.xml: It shall at least include the name, label and basic structure description of each dataset in the submitted database as well as the name, label, type, source or derivation process of each variable in each dataset. For foreign language study, description of each data set, label and derivation process of each variable, code list of efficacy indicators should be in Chinese.
- Reviewer’s Guide: including but not limited to: description on the use of study data, relationship between CSR and data, some key information in the study documents, description on the use of submitted program code, encoding used in the dataset and description on other special circumstances. This document should be in Chinese.

CHALLENGES FOR SUBMISSION PACKAGE TRANSLATION

ORGANIZATION OF MULTIPLE DOCUMENTS AND TRANSLATE SOURCE

There are various types of translation documents required in the NMPA's guideline. The aCRF and RG are PDF files, the original/analysis database is xpt file, and the define is xml file. The translation method is not consistent for each type of file. Especially for xpt and xml files, their translation methods are very different from usually document translation. Especially for define.xml, the translation comes from multiple sources, for example, the dataset label and variable label should from SDTM/ADaM IG, and be consistent with SAS data sets, the comments and derivation should be translated manually, etc. So, how to manager these contents and keep consistency is the first challenge we meet.

SIMILAR NUMEROUS COMPLEX ALGORITHM

The derivations in define.xml need translate to Chinese per NMPA's guideline. In hands-on projects, these derivations usually in larger number and have complex contents. While lots of them have some degree of similarity, one small mistake can make a big difference. It is necessary to combine the protocol, SAP, and CRF to ensure the accuracy. For translators with no programming experience or CDISC experience, this process has great challenges and risks.

GARBLED CHARACTERS IN CN DATASET

For cross-language data process, garbled character is an important issue which affect the final package quality. For pdf files, xml files and SAS codes, it is relatively easy to avoid the generation of scrambled code by transcoding. But for the SAS data set, the target encoding selection and transcoding process has a lot of details to pay attention to. The commonly used Chinese encoding include GB2312(EUC-CN) and UTF-8, both of them have advantages and disadvantages. Therefore, it is necessary to pay attention to which target encoding should be selected and whether the transcoding process is carried out successfully.

INCONSISTENCIES BETWEEN THE DOCUMENTS/DATASETS

In the submission package, xpt files or SAS data sets play the central role, all other documents are used to describe or explain the data. However, both reading and translate data in xpt format require relatively complex operations, especially the large number of variable labels, which require a high degree of consistency, both between datasets and between datasets and documents. So, we need to find a better entry point to solve this question.

HARD TO CHECK AND REVIEW

As mentioned in previous, there are many file types and translation sources in translation studies, so high requirements are also put forward to review. How review works should be performed, how they should be recorded, and how they should be tracked become difficulties that need to be overcome for high-quality delivery.

WAYS TO RESOLVE

FIND THE ENTRY POINT OF THE TRANSLATION WORK

After dozens of hands-on projects experience, we believe that define.xml is the most appropriate entry point for translation work. Define.xml is a necessary document for the FDA, PMDA, and NMPA submission, and is one of the most important documents for the Agency to accurately understand the content of the data submitted for review. The requirements for the translation of define.xml in the guideline have a very large overlap with other files, especially the SAS data sets. That is to say, the translation of define.xml is also a significant part of the translation of other files.

Define.xml is a machine-readable document which is highly structed. The contents in define.xml can be conveniently grasped and retrieved through existing software, such as Pinnacle 21 C/E, Define Editor, etc. which provides a feasible scheme for our translation of define.xml. The step of extracting the contents from define.xml is done through Pinnacle 21 C in our process, however, it is optional rather than essential.

Process of Define.xml Translation

Generally speaking, the question “how to translate define.xml” is similar to "how to put elephants in refrigerator", it can be divided into three steps: the first step, to extract the content in define.xml (to open the refrigerator door), the second step, to translate the extracted contents (to put the elephant into the refrigerator), and the third step, to put the translated content back into define.xml (to close the refrigerator door).

However, in the actual operation process, we will meet lots of difficulties. The define spec provided by Pinnacle21 has a simply structure and highly depends on the structure of define.xml. It is easy to use this spec to complete step 1 and step 3, while for step 2, this spec is not used friendly enough. The difficulties are mainly reflected in the following aspects:

- Difficult to track. After the completion of translation, it is difficult to compare the original text with the translation, and the structure of the document cannot be changed, which puts forward high requirements for the reviewer of translation and poses a penitential risk to the quality of the final delivery.
- Difficult to collaborate. The content to be translated is scattered in various positions of the document, which puts forward high requirements for division of labor, cooperation and communication.
- Difficult for version control. Translate, review and update are throughout the whole lifecycle of study, once the contents need rollback, especially when the contents affect SAS data sets, the operation will be extremely complex.

Based on the above difficulties, we made some changes to the define spec structure and generate a new file: define spec config. All contents need translate are put into this config file flatly. With this config file, we can achieve the target that manager all translation in one file and easy to review, track and update.

The overall flow of define.xml translations is shown in the figure 1.

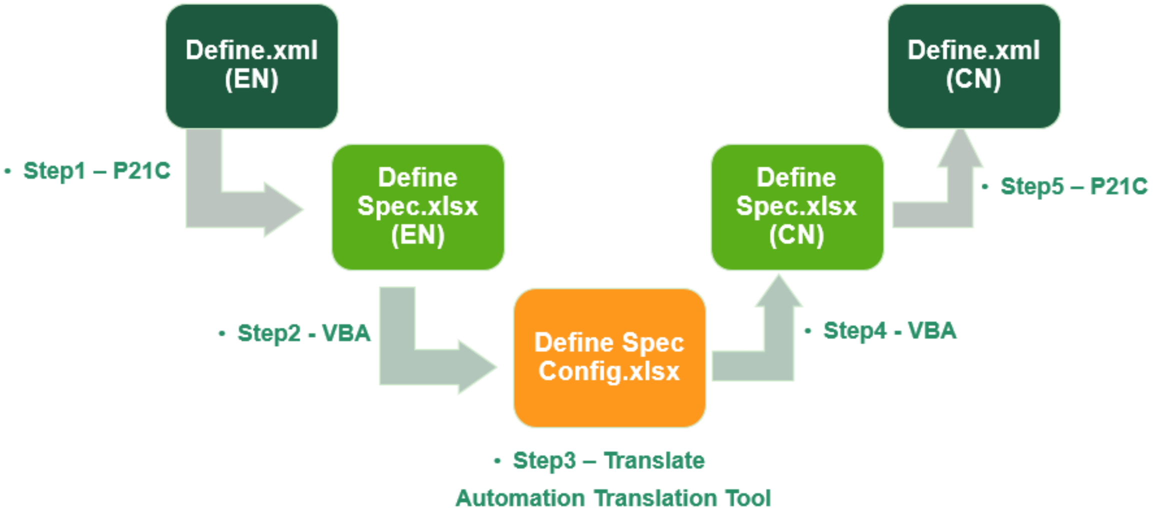
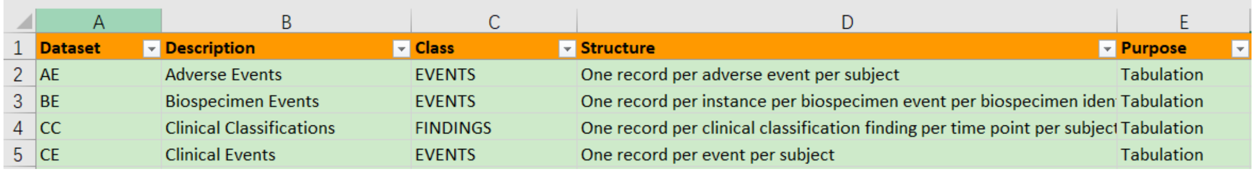


Figure 1 Overall flow of define.xml translations

Figure 2 and Figure 3 show the structure of define spec generated by Pinnacle21.



	A	B	C	D	E
1	Dataset	Description	Class	Structure	Purpose
2	AE	Adverse Events	EVENTS	One record per adverse event per subject	Tabulation
3	BE	Biospecimen Events	EVENTS	One record per instance per biospecimen event per biospecimen iden	Tabulation
4	CC	Clinical Classifications	FINDINGS	One record per clinical classification finding per time point per subject	Tabulation
5	CE	Clinical Events	EVENTS	One record per event per subject	Tabulation

Figure 2 Structure for Dataset Metadata

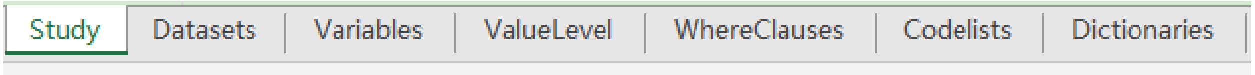
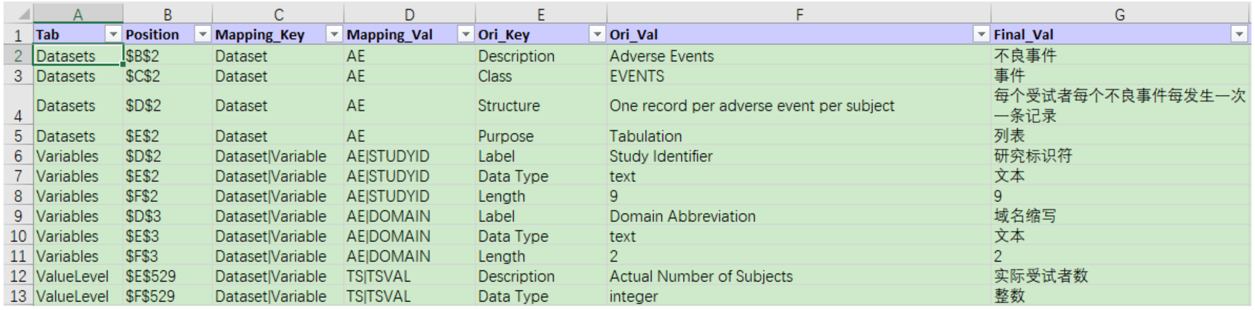


Figure 3 Tabs in define spec

Figure 4 shows the structure of define spec config.



	A	B	C	D	E	F	G
1	**Tab**	**Position**	**Mapping_Key**	**Mapping_Val**	**Ori_Key**	**Ori_Val**	**Final_Val**
2	Datasets	B2	Dataset	AE	Description	Adverse Events	不良事件
3	Datasets	C2	Dataset	AE	Class	EVENTS	事件
4	Datasets	D2	Dataset	AE	Structure	One record per adverse event per subject	每个受试者每个不良事件每发生一次一条记录
5	Datasets	E2	Dataset	AE	Purpose	Tabulation	列表
6	Variables	D2	Dataset|Variable	AE|STUDYID	Label	Study Identifier	研究标识符
7	Variables	E2	Dataset|Variable	AE|STUDYID	Data Type	text	文本
8	Variables	F2	Dataset|Variable	AE|STUDYID	Length	9	9
9	Variables	D3	Dataset|Variable	AE|DOMAIN	Label	Domain Abbreviation	域名缩写
10	Variables	E3	Dataset|Variable	AE|DOMAIN	Data Type	text	文本
11	Variables	F3	Dataset|Variable	AE|DOMAIN	Length	2	2
12	ValueLevel	E529	Dataset|Variable	TS|TSVAL	Description	Actual Number of Subjects	实际受试者数
13	ValueLevel	F529	Dataset|Variable	TS|TSVAL	Data Type	integer	整数

Figure 4 structure of define spec config

Ways to translate define spec config

There are three ways to translate define spec config:

- Mapping to guideline and IG, it will apply to Document Reference, Dataset Definitions, Variable Definitions, CT Definitions in define.xml.
- Mapping to company library: it will apply to Dataset Definitions, Variable Definitions, Value Definitions, CT Definitions which not covered by guideline and IG.
- Machine translation + Manual Review, it will apply to complex computational method definitions and comment definitions.

For variable labels not in the guideline and IG, we use VBA to map the records to our company library, which includes calculating the edit distance between variable names and calculating the cosine distance between variable labels. VBA codes can be referenced in the appendix.

For the translation of complex derivations, we use machine translation + manual update. The machine translation content is integrated into define spec config through machine translation API and used as reference, and the translate work will completed by SAS Programmers. This process requires a combination of protocol, SAP and CRF to ensure the accuracy and efficiency.

Once the define spec config translation completed, this file can be used as source to keep consistency in other files. The contents in define spec config can be easily to update into SAS data sets by SAS utility, and this config can also conveniently be used as a reference for aCRF and RG translation to keep consistency.

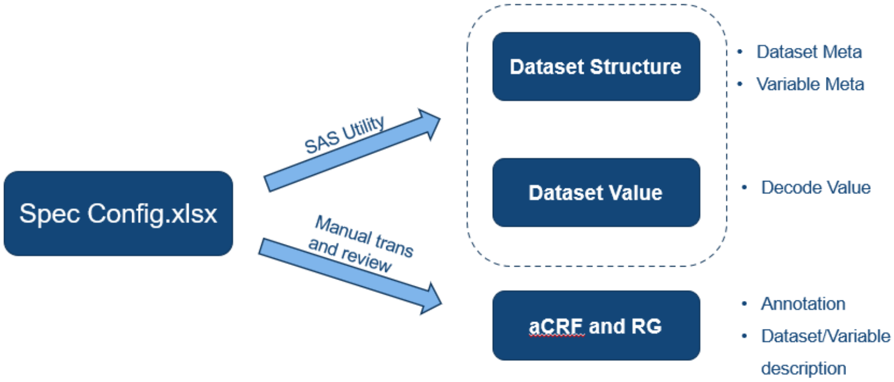


Figure 5 Relationship between Define spec config and other files

TRANSCODING AVOIDS TO GARBLED CHARACTERS

The garbled characters problem is very common during cross-language data transmission, but this problem is not unavoidable as long as we prepare properly.

Selection of target code and SAS version

Before transcoding, we need to first determine the target encoding. The commonly used Chinese encoding include UTF-8 and GB2312(EUC-CN), which are superior to each other. UTF-8 is more applicable, with little risk of garbled characters. However, this kind of coding requires more storage space, which means that the number of Chinese words can be stored in the data set is less. EUC-CN requires relatively low storage space, but it cannot ensure all characters are transcoded correctly, especially for non-English records in MRCT. So, the use of EUC-CN still has the risk of garbled. The actual choice of which encoding should be used in a project needs to be based on the actual situation, but in general we think that UTF-8 is a better choice.

UTF8 is the default encoding for Unicode SAS, so we recommend that all relevant operations on SAS data sets should be completed under Unicode SAS.

When to transcode

We believe that the earlier the transcoding of data sets is carried out, the less the impact on subsequent processing and the smaller the potential risk of garbled characters. Therefore, we recommend that the first step after getting the data should be transcoding it. The transcoding process is generally shown in the following figure.

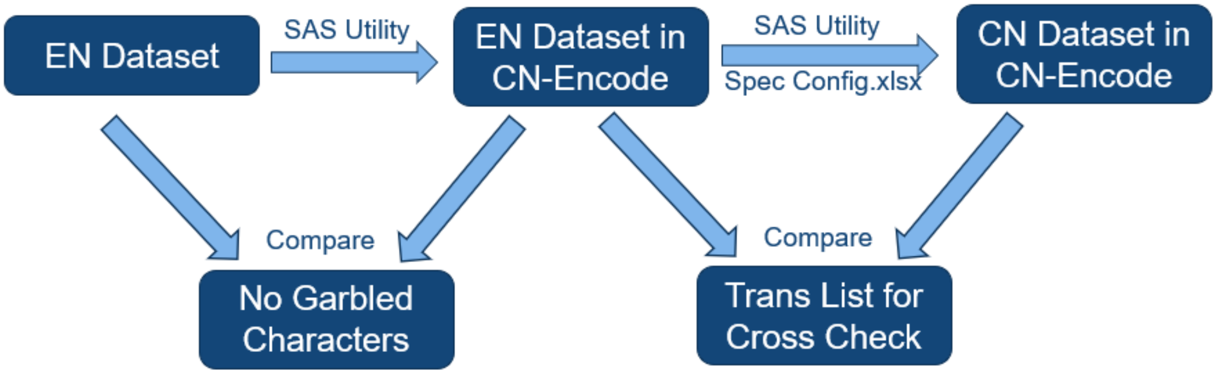


Figure 6 General process of transcoding

CHECK AND REVIEW

High-quality submission is inseparable from check and review. For translated submission packages, we summaries three main ways to achieve well reviewed.

- Run Pinnacle21 in NMPA engine. Check the data for general consistency issues with Pinnacle21.
- Independent Validation: Check the consistency among English xpt file, Chinese xpt file and config file through independent programming.
- Senior Review: Manual review through the summarized check list to ensure the overall quality of the package. The check list is as follow.

Type	#	Description
General	1	The folder sturcture follow eCTD m5 requirements
aCRF	1	Chinese Version aCRF exists and the contents translated correctly.

Type	#	Description
xpt files	1	The xpt files located in correct position, and the number of xpt files is equal to source data.
	2	All xpt files must have lower case names and consistant with define.xml and source data.
	3	Verify all xpt files can be opened without errors and each xpt file has the same number of observations as the corresponding source dataset.
	4	Verify each dataset label is translated correctly and is consistant with define.xml.
	5	Verify variable labels for each dataset are translated correctly and are consistant with define.xml.
	6	Verify the MedDRA version is correct and the decoded values are translated correctly.
	7	Verify the WHODrug version is correct and the decoded values are translated correctly.
define.xml	1	Define.xml located in correct folder and named as define.xml
	2	Verify whether correct stylesheet file(.xsl) is used.
	3	Verify study name in define.xml and the overall structure of the define.xml are consistant with source file.
	4	Verify the order of datasets in define.xml is correct.
	5	Make sure all xpt files and other supportive documents are included in the define.xml and the hyperlinks worked well.
	6	Verify Dataset Description, Class, Structure, Purpose are translated correctly.
	7	Verify Variable Label, Datatype, Origin, Derivation, Comments are translated correctly.
	8	Verify references to Meddra and WhoDrug dictionaries listed in External Dictionaries under the Controlled Terminology section are the same as listed in the Reviewer's Guide.
	9	Spot check Value Level Metadata (VLM) hyperlink worked well and translated correctly.
	10	Spot check efficacy codelist encoded value translated correctly.
	11	Spot check CRF page numbers are correct.
Reviewer's Guide	1	The file located in correct position and named correctly (csdrg for SDTM package and adrg for ADaM package).
	2	Verify the dictionary version is correct.
	3	Verify the encoding of translated datasets is included in RG.
	4	Verify the document's sturcture is constant with source file.
	5	Verify the protocol related contents consistant with Chinese version protocol.
	6	Verify the CSR related contents consistant with Chinese version CSR.
	7	Verify all contents are translated smoothly and correctly.

Table 1 Checklist for Senior Review

CONCLUSION

By extracting the content from define.xml into define spec config and using config as the basis for subsequent translations, the difficulty of managing translation documents is solved and consistency is ensured. For complex derivation and comments, use machine translation as reference and complete the translation work by SAS Programmers. Through early transcoding and develop SAS Utility to avoid the appearance of garbled in SAS data sets. Control the overall quality of submission package through compliance check tool, independent validation and senior review.

REFERENCES

NMPA 2020. “Guideline on the Submission of Clinical Trial Data”. https://www.nmpa.gov.cn/directory/web/nmpa/images/obbSqc7vwdm0ssrU0enKb7dtd29u9a4tbzUrdTyo6jK1NDQo6mhty5wZGY=.pdf

CDISC Define-XML Team, “CDISC Define-XML Specification.”, Version 2.1, 2019, CDISC. https://www.cdisc.org/standards/foundational/define-xml/define-xml-v2-1

Michael, S and Lavanya, P. 2018. UTF What? A Guide for Handling SAS Transcoding Errors with UTF-8 Encoded Data: https://www.pharmasug.org/proceedings/2018/BB/PharmaSUG-2018-BB08.pdf

Pinky, D. 2018. Generating Define.xml from Pinnacle 21 Community: https://www.lexjansen.com/pharmasug/2018/AD/PharmaSUG-2018-AD29.pdf

Richard D. Langston, SAS Institute Inc. 2018, A Macro for Ensuring Data Integrity When Converting SAS® Data Sets, Available at： https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1778-2018.pdf

CONTACT INFORMATION

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

Shijia Wang
Tigermed Co., Ltd., Shanghai, China
(+86)21-50276030
Shijia.wang@tigermedgrp.com

APPENDIX

VBA CODE FOR CALCULATE COSINE SIMILARITY

```
Function CosineSimilarity(strA As String, strB As String) As Double
    Dim objDic_All As Object, objDic_1 As Object, objDic_2 As Object
    Dim lngID As Long, strKey As String
    Dim arrKey As Variant, arrResult As Variant
    Dim dblSum As Double, dblVal_A As Double, dblVal_B As Double

    If strA = "" Or strB = "" Then
        CosineSimilarity = 0
        Exit Function
    End If

    Set objDic_All = CreateObject("Scripting.Dictionary")
    Set objDic_1 = CreateObject("Scripting.Dictionary")
    Set objDic_2 = CreateObject("Scripting.Dictionary")
```

```
      For lngID = 1 To Len(strA)
          strKey = Trim(Mid(strA, lngID, 1))
          If strKey <> "" Then
              objDic_All(strKey) = ""
              objDic_1(strKey) = Val(objDic_1(strKey)) + 1
          End If
      Next
      For lngID = 1 To Len(strB)
          strKey = Trim(Mid(strB, lngID, 1))
          If strKey <> "" Then
              objDic_All(strKey) = ""
              objDic_2(strKey) = Val(objDic_2(strKey)) + 1
          End If
      Next


      arrKey = objDic_All.Keys
      ReDim arrResult(LBound(arrKey) To UBound(arrKey), 1 To 3)

      For lngID = LBound(arrKey) To UBound(arrKey)
          arrResult(lngID, 1) = arrKey(lngID)
          arrResult(lngID, 2) = objDic_1(arrKey(lngID)) + 0
          arrResult(lngID, 3) = objDic_2(arrKey(lngID)) + 0
      Next

      Set objDic_All = Nothing
      Set objDic_1 = Nothing
      Set objDic_2 = Nothing

      For lngID = LBound(arrResult) To UBound(arrResult)
          dblSum = dblSum + arrResult(lngID, 2) * arrResult(lngID, 3)
          dblVal_A = dblVal_A + arrResult(lngID, 2) ^ 2
          dblVal_B = dblVal_B + arrResult(lngID, 3) ^ 2
      Next

      CosineSimilarity = dblSum / (Sqr(dblVal_A) * Sqr(dblVal_B))
End Function
```

VBA CODE FOR CALCULATE EDIT DISTANCE

```
  Private Function min(one As Integer, two As Integer, three As Integer)
      min = one
      If (two < min) Then
       min = two
      End If
      If (three < min) Then
       min = three
      End If
  End Function

  Private Function ld(str1 As String, str2 As String)
  Dim n, m, i, j As Integer
  Dim ch1, ch2 As String
      n = Len(str1)
      m = Len(str2)
      Dim temp As Integer
```

```
    If (n = 0) Then
        ld = m
    End If
    If (m = 0) Then
        ld = n
    End If
Dim d As Variant
ReDim d(n + 1, m + 1) As Variant
    For i = 0 To n
        d(i, 0) = i
    Next i
    For j = 0 To m
        d(0, j) = j
    Next j
    For i = 1 To n
        ch1 = Mid(str1, i, 1)
        For j = 1 To m
            ch2 = Mid(str2, j, 1)
            If (ch1 = ch2) Then
            temp = 0
            Else
                temp = 1
            End If
            d(i, j) = min(d(i - 1, j) + 1, d(i, j - 1) + 1, d(i - 1, j - 1)
+ temp)
        Next j
    Next i
    ld = d(n, m)
End Function

Public Function Sim(str1 As String, str2 As String) As Double
    Dim ldint As Integer
    ldint = ld(LCase(str1), LCase(str2))
    Dim strlen As Integer
    If (Len(str1) >= Len(str2)) Then
        strlen = Len(str1)
    Else
        strlen = Len(str2)
    End If
    Sim = 1 - ldint / strlen
End Function
```

VBA CODE FOR CALLING GOOGLE TRANSLATE API

```
Public Function GoogleTranslate(sourceText As String, sourceLanguage As
String, targetLanguage As String, apiKey As String) As String
    Dim googleApi As Object
    Dim url As String
    Dim TransStr As String
    Set googleApi = CreateObject("MSXML2.XMLHTTP")
    url = "https://www.googleapis.com/language/translate/v2?key=" & apiKey
    url = url & "&source=" & sourceLanguage
    url = url & "&target=" & targetLanguage
    url = url & "&q=" & sourceText
    googleApi.Open "GET", url, False
    googleApi.send
    ObjJson = googleApi.responseText
    Dim oJSON As Object
    Set oJSON = JsonConverter.ParseJson(ObjJson)
    TransStr = oJSON("data")("translations")(1)("translatedText")
    GoogleTranslate = TransStr
End Function
```