



# Khmer Word Segmentation Encoding API Technical Report

**Cambodia Country Component**  
**PAN Localization project**  
**PAN Localization Cambodia (PLC) of IDRC**

**Tuesday, June 30, 2009**  
**Prepared by: Mr. Tith Sakal**

## Khmer Word Segmentation API documentation (Monday, July 06, 2009)

1.	Introduction to Khmer Word Segmentation API .....	4
2.	Class Description .....	4
2.1.	BigramBased .....	4
2.2.	CEGenerator .....	5
2.3.	CEWord .....	5
2.4.	CharacterRecognizer .....	6
2.5.	Consonant .....	6
2.6.	ConsonantRule .....	7
2.7.	ConsonantShifter .....	7
2.8.	Delimiter .....	8
2.9.	EastSubscript .....	8
2.10.	EastVowel .....	8
2.11.	EditDistance .....	9
2.12.	ErrorWordDetector .....	9
2.13.	IndependentScript .....	9
2.14.	IndependentVowel .....	10
2.15.	KCCBigramBased .....	10
2.16.	KCCBigramModel .....	10
2.17.	KCCBuffer .....	11
2.18.	KCCBufferInfo .....	11
2.19.	KCCDetector .....	12
2.20.	KhmerCharacterSet .....	13
2.21.	KhmerDictionary .....	13
2.22.	KhmerDictionaryBased .....	14
2.23.	KhmerScript .....	15
2.24.	KhmerTypeSet .....	15
2.25.	KhmrCharComparer .....	16
2.26.	KhmrCharSplitter .....	16
2.27.	MMAModel .....	17
2.28.	NorthVowel .....	17
2.29.	OtherCompound .....	17
2.30.	PLCComponents .....	18
2.31.	PLCCorpus .....	18
2.32.	PLCDictionary .....	19
2.33.	PLCSearchEngine .....	19

2.34.	ReaderEngine .....	20
2.35.	ROBAT .....	20
2.36.	Rule.....	20
2.37.	RuleConsInfo .....	21
2.38.	RuleEngine .....	22
2.39.	SegWord.....	22
2.40.	SouthSubscript .....	23
2.41.	SouthVowel .....	23
2.42.	State .....	23
2.43.	StateCollection .....	24
2.44.	Subscript.....	24
2.45.	SubscriptRule.....	24
2.46.	TerminatorGraphInfo .....	25
2.47.	VariousSign .....	26
2.48.	Vowel.....	26
2.49.	VowelRule .....	26
2.50.	WestSubscript.....	27
2.51.	WestVowel .....	27
2.52.	Word .....	28
2.53.	WordBigramBased.....	28
2.54.	WordBigramModel.....	28
2.55.	WordInfo .....	29
2.56.	ZWJ .....	29
2.57.	ZWNJ.....	29
3.	Data Files .....	30
4.	Usage .....	31
4.1.	Segmentation.....	31
4.2.	Spell Checking .....	32

## 1. Introduction to Khmer Word Segmentation API

This report is not detail discuss on Khmer Word Segmentation technique, we only guide developer what is inside Khmer Word Segmentation API and how to use it. If you developer or researcher plan to get deep information about Khmer Word Segmentation please refer to Khmer Word Segmentation Project report made by PAN Localization Cambodia of IDRC.

Khmer Word Segmentation is a basic research for Natural Language Processing. The purpose of this research is to split Khmer words from a sentence, Khmer Words are written continuously without any delimiter like space. Space is inserted into a sentence in some circumstances to break in reading. Space is not used to break between word and word in a sentence for Khmer Language. According to this problem to develop some application base on word for our language such as Spell Checker, Text To Speech is impossible.

Khmer Word Segmentation API is an engine to hole the process for the problem mentioned above. The tool for developing the API is Visual Studio Dot Net 2005 (VS.Net 2005) and apply Visual Basic Dot Net as a programming Language. As the result we got DLL (Dynamic Link Library) file for Khmer Word Segmentation API.

The previous version of the API has been separated into four main modules, KHMER WORD SEGMENTATION MODULE, KHMER COMON EXPRESSION GENERATOR MODULE, KHMER CHARACTER CLUSTER MODULE, and DATA SEARCH MODULE. Those modules are independent API; however, they link to each other during the process. By this time we combine all the modules into one API to provide an easy way for developer using our engine.

## 2. Class Description

### 2.1. *BigramBased*

Class Name	BigramBased	
<b>Description</b>	This class inherits from the PLCSearchEngine. It provides all the functionalities to access to the word Bigram trained corpus.	
<b>Attributes</b>	corpBi	An object to access directly to bigram corpus data files.
<b>Operations</b>	getBigramType()	Return the number of different Bigrams that has the given word as the first word of the pair.
	getBigramToken()	Return the number of all the Bigrams that has the given word as the first word of the pair.
	getAllBigramType()	Return the number of all the different Bigrams in the corpus.

	getAllBigramToken()	Return the number of all the Bigrams in the corpus.
	getVocabulary()	Return the number of vocabulary in the whole corpus.

## 2.2. CEGenerator

Class Name	CEGenerator	
Description	The main class for the generation of Khmer Common Expression.	
Attributes	kccGen	KCC generator
	ruleSet	All the rule for the generation of the KCE
Operations	generateCE()	Return the CE string for a string or Khmer script array given.
	getSubscriptRulesNumber()	Return the subscript rule number that match the current KCCBuffer
	getVowelRulesNumber()	Return the vowel rule number that match the current KCCBuffer
	fillBuffer()	Replace the current KCC Buffer into KCE correspondent the matched rules.

## 2.3. CEWord

Class Name	CEWord	
Description	The class which represents the pair of word and its KCE	
Attributes	key	The KCE string
	iniWord	The initial word string

#### 2.4. CharacterRecognizer

Class Name	CharacterRecognizer	
Description	It provides all the functionalities to access to all the Khmer character and information.	
Attributes	reader	Initializer of Khmer character and its information from a file given.
	FILENAME	The name of the file that contains the character information.
	isKhmerCharacter()	Test if the character given is Khmer character or not. Return True if it is Khmer character and False if it is not.
	getCharacter()	Return the information of the Khmer character given.
	convertStrToKhmerscripts()	Return array of Khmerscript (Khmer character code and its information) from a given sentence.
Operations		

#### 2.5. Consonant

Class Name	Consonant	
Description	This class inherits from Khmer script. It represents a Khmer consonant.	
Attributes	VowelSound	The sound of the character A or O.
	Group	The group of the character in Khmer language. The group is from 1 to 6.
	Tamroutable	The character is TAMROUTABLE or not.
	SoundShiftable	The character is Soundshiftable or not.
	SubPosition	The position of the consonant; it could be west, south or east.
	CoChar	The character with the opposite sound of the current character. For example, if the current character is ក then the CoChar is គ.

<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

## 2.6. ConsonantRule

Class Name	ConsonantRule	
<b>Description</b>	Inherited from the class Rule. It represents the rule in the consonant, ROBAT part.	
<b>Operations</b>	matchRule()	Return if the KCC buffer given match to a certain rule or not. It must be override by its subclass
	matchConsonant()	Return if the KCC buffer matches the rule in the consonant part or not.
	matchSubscript()	Return if the KCC buffer matches the rule in the subscript part or not.
	matchConsonantShifter()	Return if the KCC buffer matches the rule in the consonant shifter part or not.
	matchVowel()	Return if the KCC buffer matches the rule in the Vowel part or not.
	matchVS()	Return if the KCC buffer matches the rule in the various sign part or not.

## 2.7. ConsonantShifter

Class Name	ConsonantShifter	
<b>Description</b>	This class inherits from Khmer script. It represents a Khmer consonant shifter.	
<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

### 2.8. Delimiter

Class Name	Delimiter	
<b>Description</b>	This class inherits from Khmer script. It represents a purely delimiter of Khmer the KCC such as English character, Khmer number etc...	
<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

### 2.9. EastSubscript

Class Name	EastSubscript	
<b>Description</b>	This class inherits from subscript. It represents subscript that is found in the east of the consonant.	
<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

### 2.10. EastVowel

Class Name	EastVowel	
<b>Description</b>	This class inherits from Vowel. It represents vowel that is found in the east of the consonant.	
<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.



### 2.11. EditDistance

Class Name	EditDistance	
Description	This class contains the function to measure the amount of difference between two Khmer sequences.	
Attributes	objKhmrCharSpliter	Return as KhmrCharSpliter
	objKhmrCharComparer	Return as KhmrCharComparer
Operations	KhmrCharSpliter()	Property to access to KhmrCharSpliter attribute
	KhmrCharComparer()	Property to access to KhmrCharComparer attribute
	FindDifferentialPercent()	This function is base from a method named "Levenshtein Distance". The function is for measuring the amount of difference between two sequences.
	Minimum()	Get minumum of three given value

### 2.12. ErrorWordDetector

Class Name	ErrorWordDetector	
Description	The main class of the module.	
Attributes	ceGen	KCE generator assigned for the process of the generation of CE.
Operations	getBestSegmentation()	Return the best word segmentation from the string given. The method must be override.
	getSegments()	Return all the possible segmentations from the sentence given.

### 2.13. IndependentScript

Class Name	IndependentScript	
Description	This class inherits from Khmerscript. It represents the Khmer character that can stand alone in writing. It is also the terminator of the KCC also.	

<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

#### ***2.14. IndependentVowel***

<b>Class Name</b>	<b>IndependentVowel</b>	
<b>Description</b>	This class inherits from Khmerscript. It represents the Khmer independent vowel.	
<b>Attributes</b>	SimSound	A string that represents the similar writing of the character
	VowelSound	The vowel sound of the character A or O.
<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

#### ***2.15. KCCBigramBased***

<b>Class Name</b>	<b>KCCBigramBased</b>	
<b>Description</b>	This class inherits from the BigramBased class. Therefore, it gets all the functionalities in its parent class.	
<b>Attributes</b>	Path	The path of the file that contains the KCC bigram data.

#### ***2.16. KCCBigramModel***

<b>Class Name</b>	<b>KCCBigramModel</b>	
<b>Description</b>	This class inherits from the ErrorWordDetector. It provides the functionality to segment Khmer sentence using Khmer Character Cluster (KCC or Orthographic syllable) Bigram Model.	
<b>Attributes</b>	kccBigram	An attribute used for retrieving of data of the

		KCC Bigram trained corpus.
<b>Operations</b>	getBestSegmentation()	Return the best word segmentation from the string given. The method overrides its parent class.
	KCCBigramMatching()	Return the best segmentation from a set of segmentation candidates that the sentence given produced based on KCC Bigram probabilities.
	wittenBellLogProbability()	Return the logarithm of the probability of a KCC sequence according to the KCC Bigram corpus.
	wittenBellProbability()	Return the probability of a KCC sequence according to the KCC Bigram corpus.

### 2.17. KCCBuffer

Class Name	KCCBuffer	
Description	The based class that represents a KCE rule.	
<b>Attributes</b>	KhmerScriptArray	KCC buffer, the array of Khmer scripts
	kccSound	The sound of the current KCC A or O
<b>Operations</b>	getKCCSound()	Return the sound of the current KCC Buffer.

### 2.18. KCCBufferInfo

Class Name	KCCBufferInfo	
Description	The definition of script type	
<b>Attributes</b>	SIZE	11
	CONSONANT	0
	INDEPENDENT_VOWEL	0
	SUB_SCRIPT1	1

	SUB_SCRIPT2	2
	CONSONANT_SHIFTER	3
	VOWEL	4
	VARIOUS_SIGN_1	5
	VARIOUS_SIGN_2	6
	OTHERS	7
	OTHER_SUBSCRIPT	8
	ROBAT	9
	ZWJ	10 (Zero With Joiner)
<b>Operations</b>	There is no operation for this class	

### 2.19. KCCDetector

Class Name	KCCDetector	
Description	The main class for the detection of Khmer KCC.	
<b>Attributes</b>	terminatorGraph	Terminator graph information.
	chrRecognizer	Khmer character information and functionalities.
	terminatorFile	A file that contains terminator graph information.
<b>Operations</b>	initialTerminatorGraph()	Initialize terminator graph information.
	convertToKCCBufferArrays()	Return the array which contains all the buffer of the KCC in the form of KCC buffer format from a Khmer string given.
	normalize()	Return the normalized string from a Khmer string given.
	reformKCC()	Return the normalized string from the buffer array given.

	detectKCC()	Detect KCC from the Khmer script array given.
--	-------------	---

### 2.20. *KhmerCharacterSet*

Class Name	KhmerCharacterSet	
Description	Some Khmer Characters. All attributes are Char type	
Attributes	COEUNG	\U17D2
	THO	\U1792
	PO	\U1796
	QU	\U17A7
	QUK	\U17A8
	QAI	\U17B0
	TRIISAP	\U17CA
	SIGN_KHAN	\U17D4
	LO	\U179B
	LAK	\U17D8
Operations	There is no operation for this class	

### 2.21. *KhmerDictionary*

Class Name	KhmerDictionary	
Description	It provides all the functionalities to access to the Khmer dictionary.	
Attributes	khDictionary	A list of all the KCE and its corresponding words.
	PlaceNames	A list of the places.
	OrgNames	A list of the name of the organization

	RomanNames	A list of all the Romanization words.
<b>Operations</b>	initialDictionary()	Initialize the dictionary from the file to the khDictionary.
	getWordsFromChounnat()	Return the list of the word from the CHUON NATH that has the same KCE as given by the user.
	existWordInChounnat()	Test if the word string given exists in the dictionary or not.
	existCEInChounnat()	Test if the KCE given exists in CHUON NATH dictionary or not.
	existWordsInChAndOther()	Test if the word given exists in the CHUON NATH dictionary or is a place, Romanization, organization or not.
	existWordsInAllDictsAndOther()	Test if the word given exists in the CHUON NATH dictionary or Other dictionary or is a place, Romanization, organization or not.
	getWordsFromAllDicts()	Return the list of the words from the all the dictionary by providing the KCE.
	existWordInAllDicts()	Test if the word given exists in the entire dictionary or not.
	existCEInAllDicts()	Test if the KCE given exists in the entire dictionary or not.
	existPlace()	Test if the word given is a place or not.
	existOrganization()	Test if the word given is the name of an organization or not.
	existRomans()	Test if the word given is the Romanization word or not.

## 2.22. *KhmerDictionaryBased*

<b>Class Name</b>	<b>KhmerDictionaryBased</b>
<b>Description</b>	This class inherits from the PLCSearchEngine. It provides all the

	functionality to access to the Khmer dictionary.
--	--

### 2.23. *KhmerScript*

Class Name	KhmerScript	
Description	An abstract class for a Khmer character and its related information.	
Attributes	Code	The code of the Khmer character.
Operations	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given. It must be override by its sub classes.

### 2.24. *KhmerTypeSet*

Class Name	KhmerCharacterSet	
Description	Type of some Khmer characters. All attributes are in Byte. The right side is the value assigning to each type set.	
Attributes	C_SOUTH_SUBSCRIPT	1
	C_EAST_SUBSCRIPT	2
	C_WEST_SUBSCRIPT	3
	INDEPENDENT_VOWEL	4
	WEST_VOWEL	5
	SOUTH_VOWEL	6
	NORTH_VOWEL	7
	EAST_VOWEL	8
	VARIOUS_SIGN	9
	CONSONANT_SHIFTER	10
	INDENPENDENT_SCRIPT	11
	DELIMITER	12

	ROBAT	13
	ZWNJ	14
	ZWJ	15
	VOWELSOUND_A	0
	VOWELSOUND_O	1
<b>Operations</b>	There is no operation for this class	

### 2.25. *KhmrCharComparer*

Class Name	KhmrCharComparer	
<b>Description</b>	This class implements from IKhmrCharComparer interface	
<b>Attributes</b>	ComparRule	Store the comparesionRule table from <i>ComparesionRule.PLC</i> data file.
<b>Operations</b>	FindDifferential()	Compare two given strings and return the value as Sigle. The value could be 0, 1.0 and the value from the <i>comparesionRule.plc</i> according to the given String.
	New()	Loading the comparesionRule table into ComparRule attribute.

### 2.26. *KhmrCharSpliter*

Class Name	KhmrCharSpliter	
<b>Description</b>	This class implements from IKhmrCharSpliter Interface.	
<b>Attributes</b>	KhmerConsonants	This attribute has type of Hashtable (collection provide by .Net technology)
<b>Operations</b>	Split()	Split the Khmer Text given in to character and return as a collection (ArrayList)
	IsKhmerConsonant()	Check if the character given is Khmer Consonant or not



**2.27. MMAModel**

Class Name	MMAModel	
Description	This class inherits from the ErrorWordDetector. It provides the functionality to segment Khmer sentence using Maximum Matching Algorithm.	
Attributes	mma	An attribute used for the looking up for data in the dictionary.
Operations	getBestSegmentation()	Return the best word segmentation from the string given. The method overrides its parent class.
	maximumMatching()	Return the best segmentation from a set of segmentation candidates that the sentence given produced based on maximum matching algorithm.

**2.28. NorthVowel**

Class Name	NorthVowel	
Description	This class inherits from Vowel. It represents vowel that is found the north of the consonant.	
Operations	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

**2.29. OtherCompound**

Class Name	OtherCompound	
Description	This class is used to read a given PLC file	
Attribute	list	Hash table to store the data
Operations	checkHeader()	Check if the given file is PLC file or not.
	New()	Constructor to read the given PLC file and put the data into <i>list</i> attributes.

### 2.30. *PLCComponents*

PLC Components is a module containing two interfaces, *IKhmrCharComparer* and *IKhmrCharSplitter*. *IKhmrCharComparer* contains one function, *FindDifferential*, in type of Sigle and has two parameters in String type. *IKhmrCharSplitter* is an interface contains one function, *Split*, in type of ArrayList and has one parameter in type of String.

### 2.31. *PLCCorpus*

Class Name	PLCCorpus	
Description	It provides all the functionalities to access to the bigram data.	
Attributes	indexTable	A table that contains all the indexes of all the words in the corpus.
	ABType	Number of all the different Bigram pair in the corpus.
	ABToken	Number of all the Bigram pair in the corpus provided.
	AVocabulary	Number of all the vocabulary in the text corpus.
	NFilesInFolder	Number of file in each folder of Bigram data.
Operations	getBigramType()	Return the number of different Bigrams that has the given word as the first word of the pair.
	getBigramToken()	Return the number of all the Bigrams that has the given word as the first word of the pair.
	getAllBigramType()	Return the number of all the different Bigrams in the corpus.
	getAllBigramToken()	Return the number of all the Bigrams in the corpus.
	getVocabulary()	Return the number of vocabulary in the whole corpus.
	getBigramFileName()	Return the name of the file that contains the pair of Bigram given information.

	getIndexWord()	Return the index of the word given.
	sequenceSearch()	Search for the bigram information in a file using sequence search algorithm
	dichoSearch()	Search for the bigram information in a file using binary search algorithm

### 2.32. *PLCDictionary*

Class Name	PLCDictionary	
Description	Contains dictionary data and some functionality related to word in dictionary	
Attributes	TableDic	Hashtable to store word list
	FilePath	Path data file in hard drive
	objEditDistance	Object of EditDistance class
	TableMappedKey	Hash table to store MappedKey data
Operations	LoadMappedKey()	Load the map key data file located in <i>\Dictionary\Mappekey.PLC</i>
	LoadWordList()	Load the word list to memory
	EditDistanceEngine()	Property to return or set to edit distance attribute
	ExistSimilarWord()	Check if the given word exist or similar to the word list in dictionary or not
	SearchSimilarWords()	Return the similar word given, it could be more than one word

### 2.33. *PLCSearchEngine*

Class Name	PLCSearchEngine
Description	The main class of the module.

<b>Attributes</b>	KhDict	An object used to provides all the functionalities to access to the Khmer dictionary word list.
-------------------	--------	---

### 2.34. ReaderEngine

<b>Class Name</b>	<b>ReaderEngine</b>	
<b>Description</b>	Provide the functionality to initialize the Khmer character information into the file.	
<b>Attributes</b>	dico	A table to store the information of the Khmer character.
<b>Operations</b>	initialTerminatorGraph()	Initialize the Khmer character information from the file given into the local variable.

### 2.35. ROBAT

<b>Class Name</b>	<b>ROBAT</b>	
<b>Description</b>	This class inherits from Khmerscript. It represents the Khmer ROBAT sign.	
<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

### 2.36. Rule

<b>Class Name</b>	<b>Rule</b>	
<b>Description</b>	The based class that represents a KCE rule.	
<b>Attributes</b>	Rule	The matching rule part
	Mapping	The mapping of the matching rule part
<b>Operations</b>	matchRule()	Return if the KCC buffer given match to a certain rule or not. It must be override by its subclass

**2.37. RuleConsInfo**

Class Name	RuleConsInfo	
Description	This class is a custom define value.	
Attributes	CONSONANT	Type of this attribute is Byte and assign to 0
	SUBSCRIPT	Type of this attribute is Byte and assign to 1
	CONSONANT_SHIFTER	Type of this attribute is Byte and assign to 2
	VOWEL	Type of this attribute is Byte and assign to 3
	VARIOUS_SIGN	Type of this attribute is Byte and assign to 4
	SIZE	Type of this attribute is Byte and assign to 5. To identify there is 5 Khmer script types in this class.
	KCC_SOUND_A	Type of this attribute is Char and assign to 0
	KCC_SOUND_O	Type of this attribute is Char and assign to 1
	KCC_EITHER_SOUND	Type of this attribute is Char and assign to 2
	OTHER_NOT_EXIST	Type of this attribute is Char and assign to 0
	OTHER_EXIST	Type of this attribute is Char and assign to 1
	OTHER_OPTIONAL	Type of this attribute is Char and assign to 2
	RIGHT_SAME	Type of this attribute is Char and assign to 0

<b>Operations</b>	RIGHT_NOT_EXIST	Type of this attribute is Char and assign to 1
	RIGHT_ALTERNATESOUND	Type of this attribute is Char and assign to 2
	There is no Operation for this class.	

### 2.38. RuleEngine

Class Name	RuleEngine	
<b>Description</b>	It represents all the rules for the generation of KCE.	
<b>Attributes</b>	consonantRules	All the consonant rules
	subscriptRules	All the subscript rules
	vowelRules	All the vowel rules
<b>Operations</b>	getConsonantRulesNumber()	Return the consonant rule number that match the current KCCBuffer
	getSubscriptRulesNumber()	Return the subscript rule number that match the current KCCBuffer
	getVowelRulesNumber()	Return the vowel rule number that match the current KCCBuffer
	fillBuffer()	Replace the current KCC Buffer into KCE correspondent the matched rules.

### 2.39. SegWord

Class Name	SegWord	
<b>Description</b>	This class describes the necessary information of a segmented word.	
<b>Attributes</b>	word	The segmented word string.
	state	The state of the segmented word whether it is a correct word or error word.
	ce	The Khmer common expression of the segmented word

	tag	Additional information of the word whether it is a name or place or something else.
--	-----	---

#### 2.40. *SouthSubscript*

Class Name	SouthSubscript	
Description	This class inherits from subscript. It represents the subscript that is found in the south of the consonant.	
Operations	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

#### 2.41. *SouthVowel*

Class Name	South vowel	
Description	This class inherits from Vowel. It represents vowel that is found in the south of the consonant.	
Operations	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

#### 2.42. *State*

Class Name	State	
Description	This class describes the necessary information of a possible a state. State might be a KCC or combination of KCC. It is used necessary when segmenting the sentence into all possible segmentations.	
Attributes	iniWord	The segmented state string
	ce	The Khmer common expression of the segmented state

	initial	The start position of the state in the sentence.
	final	The last position of the state in the sentence.
	found	The status of the state whether it exists as word or not.

#### 2.43. *StateCollection*

Class Name	StateCollection	
Description	This class inherits from CollectionBase (a type of collection). This class is to store a State Object as a collection.	
Operations	Item	Return the item from the collection
	Add	Add new Item to the collection
	IndexOf	Return the index of the given State Object
	Insert	Insert the item into a collection next to the given index
	Remove	Remove the given item from the collection
	Contains	Check if the given item exists or not

#### 2.44. *Subscript*

Class Name	Subscript	
Description	This class inherits from Khmer script. It represents the Khmer subscript.	
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given. The method must be overridden by its subclass.

#### 2.45. *SubscriptRule*

Class Name	SubscriptRule	
Description	Inherited from the class Rule. It represents the rule in the subscript and consonant shifter part.	



<b>Operations</b>	matchRule()	Return if the KCC buffer given match to a certain rule or not. It must be override by its subclass
	matchConsonant()	Return if the KCC buffer matches the rule in the consonant part or not.
	matchSubscript()	Return if the KCC buffer matches the rule in the subscript part or not.
	matchConsonantShifter()	Return if the KCC buffer matches the rule in the consonant shifter part or not.
	matchVowel()	Return if the KCC buffer matches the rule in the Vowel part or not.
	matchVS()	Return if the KCC buffer matches the rule in the various sign part or not.

#### 2.46. TerminatorGraphInfo

Class Name	TerminatorGraphInfo	
Description	This class is custom define value. All the attributes are in type of Byte.	
<b>Attributes</b>	SIZE	14
	CONSONANT	0
	CONSONANT_SHIFTER	1
	WEST_VOWEL	2
	NORTH_VOWEL	3
	SOUTH_VOWEL	4
	EAST_VOWEL	5
	WEST_SUBSCRIPT	6
	EAST_SUBSCRIPT	7
	SOUTH_SUBSCRIPT	8
	VARIOUS_SIGN	9

	INDEPENDENT_VOWEL	10
	ZWNJ	11
	ZWJ	12
	ROBAT	13
	TERM	14
<b>Operations</b>	There is no operation for this class	

#### 2.47. VariousSign

Class Name	VariousSign	
<b>Description</b>	This class inherits from Khmerscript. It represents the Khmer various sign.	
<b>Attributes</b>	Omittable	The value if the sign can be omittable or not.
<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

#### 2.48. Vowel

Class Name	Vowel	
<b>Description</b>	This class inherits from Khmer script. It represents the Khmer vowel.	
<b>Operations</b>	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given. The method must be overridden by its subclass.

#### 2.49. VowelRule

Class Name	VowelRule	
<b>Description</b>	Inherited from the class Rule. It represents the rule in the vowel, various sign, ZWJ and ZWNJ part.	

<b>Operations</b>	matchRule()	Return if the KCC buffer given match to a certain rule or not. It must be override by its subclass
	matchConsonant()	Return if the KCC buffer matches the rule in the consonant part or not.
	matchSubscript()	Return if the KCC buffer matches the rule in the subscript part or not.
	matchConsonantShifter()	Return if the KCC buffer matches the rule in the consonant shifter part or not.
	matchVowel()	Return if the KCC buffer matches the rule in the Vowel part or not.
	matchVS()	Return if the KCC buffer matches the rule in the various sign part or not.

### 2.50. *WestSubscript*

Class Name	EastSubscript	
<b>Description</b>	This class inherits from subscript. It represents subscript that is found in the west of the consonant.	
<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

### 2.51. *WestVowel*

Class Name	WestVowel	
<b>Description</b>	This class inherits from Vowel. It represents vowel that is found in the west of the consonant.	
<b>Operations</b>	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

### 2.52. Word

Class Name	Word	
Description	This class is used for storing a Khmer word and the information if the word exist in Chhoun Nat Dictionary or not	
Attribute	Word	This attribute has type of String to store given word
	Ch	This attribute has type of Boolean, 1 means Chhoun Nat dictionary word and 0 means Non Chhoun Nat dictionary word.
Operations	New()	Constructor for this class

### 2.53. WordBigramBased

Class Name	WordBigramBased	
Description	This class inherits from the BigramBased class. Therefore, it gets all the functionalities in its parent class.	
Attributes	Path	The path of the file that contains the word bigram data.

### 2.54. WordBigramModel

Class Name	WordBigramModel	
Description	This class inherits from the ErrorWordDetector. It provides the functionality to segment Khmer sentence using Word Bigram Model.	
Attributes	wrdBigram	An attribute used for retrieving of data of the word Bigram trained corpus.
Operations	getBestSegmentation()	Return the best word segmentation from the string given. The method overrides its parent class.
	BigramMatching()	Return the best segmentation from a set of segmentation candidates that the sentence given produced based on Bigram probabilities.

	wittenBellLogProbability()	Return the logarithm of the probability of a word sequence according to the word Bigram corpus.
	wittenBellProbability()	Return the probability of a word sequence according to the word Bigram corpus.

### 2.55. WordInfo

Class Name	ZWJ	
Description	This class is used to identify the data in <i>Index.plc</i> in <i>WordBigram</i> data directory	
Attributes	Index	Index of word in <i>Index.plc</i> file for a specific word
	BigramToken	BigramToken in <i>Index.plc</i> file for a specific word
	BigramType	BigramType in <i>Index.plc</i> file for a specific word
Operations	There is no operation for this class.	

### 2.56. ZWJ

Class Name	ZWJ	
Description	This class inherits from Khmerscript. It represents the Zero Width Joiner.	
Operations	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.
	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.

### 2.57. ZWNJ

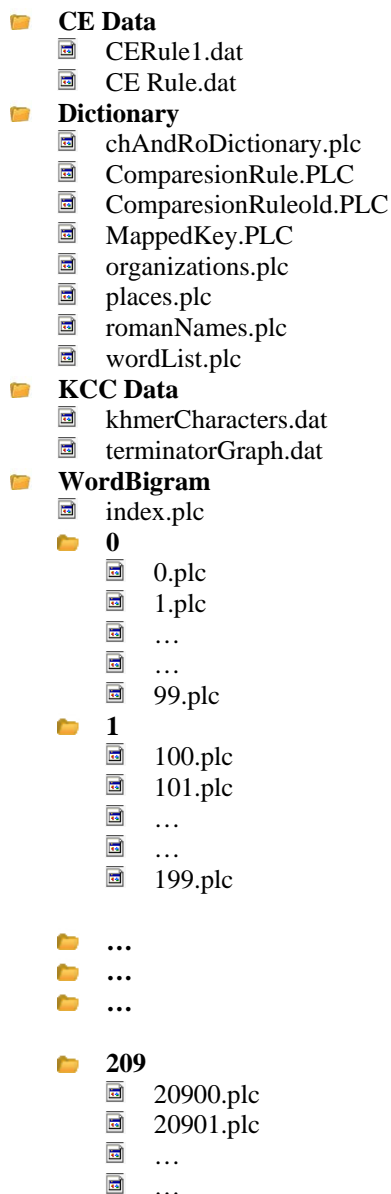
Class Name	ZWNJ	
Description	This class inherits from Khmerscript. It represents the Zero Width Non Joiner.	
Operations	isTerminator()	Return if the current character is the terminator of the KCC in the sentence or not.

	detectTerminator()	Detect if the current character is the terminator of the KCC in the sentence given.
--	--------------------	---

### 3. Data Files

This report is not detail discuss on Khmer Word Segmentation data structure, we only mention how many data files required for Khmer Word Segmentation API. If you need to get more information on that please refer to Khmer Word Segmentation Project Report by PAN Localization Cambodia of IDRC. All the data files must store in the same location with API library file.

#### Khmer Word Segmentation API data files





**Note:** In WordBigram data folder contains a file name *index.plc* and sub folders named 0 to 210. Each sub folder contains Bigram pair information file. The structure of the folder that contains the Bigram data as follow:

1. The index file
2. The folders that contain the Bigram pair information.
  - The folder is named as 0 to N
  - Each folder contains M Bigram data pair files.

## 4. Usage

### 4.1. Segmentation

There are two Modules to segment Khmer words from a sentence, MMAModel (Maximum Matching Algorithm Model), Word Bigram Model. We can apply one of them to segment Khmer words from a sentence.

## Objective

Suppose you have a text file named *KhmerSentence.txt* contains Khmer string and located in drive [C] on your hard drive (ទោះយ៉ាងណាក៏ដោយក៏សំខាន់ណាស់ដែលត្រូវបញ្ជាក់ថាលោកមិនបានដឹងអំពីជោគវាសនារបស់ល្ខើយឆឹករៀតណាមទាំងនោះទេក្រោយពីការសួរចម្លើយព្រោះលោកមានតួនាទីត្រឹមតែជាអ្នកសួរចម្លើយប៉ុណ្ណោះ). You need to split the sentence into Khmer words and put it back to another text file by writing one word per line.

## Coding in VB.Net

```
'WBM_Result is a collection object to get the result from Word BigramModel.
Dim WBM_Result As Collection

'WBM_Result is a collection object to get the result from Maximum Matching
Model.
Dim MMA_Result As Collection

'Khmer Sentence to be segmented
Dim KhmerSentence As String

'Object Word Bigram Model
Dim WordBigramMethod As New KhmerWordSegmentation.WordBigramModel

'Object Maximum Matching Model
Dim MaximumMachingMethod As New KhmerWordSegmentation.MMAModel

'Object Segment Word to get words after segmenting from any model
```

```
Dim KhmerWord As KhmerWordSegmentation.SegWord

'Split Khmer sentence into Khmer words by using WordBigram Method
WBM_Result = WordBigramMethod.getBestSegmentation(KhmerSentence, _
KhmerWordSegmentation.KhDictionary.MIX_DICTIONARY)

'Split Khmer sentence into Khmer words by using Maximum Maching Method
MMA_Result = MaximumMachingMethod.getBestSegmentation(KhmerSentence, _
KhmerWordSegmentation.KhDictionary.MIX_DICTIONARY)

'Stream Writer object to write every words after segmenting by using Word
'Bigram Model algorithm to a text file.
Dim WBM_Writer As New System.IO.StreamWriter("C:\WBM_Result.txt")

'Write khmer words in the collection to WBM_Result.txt file, each words for
one line.
For i As Integer = 1 To WBM_Result.Count
    KhmerWord = WBM_Result(i)
    WBM_Writer.Write(KhmerWord.Word)
    WBM_Writer.Write(VbCrLf)
Next
WBM_Writer.Close()

'Stream Writer object to write every words after segmenting by using
'Maximum Matching algorithm to a text file.
Dim MMA_Writer As New System.IO.StreamWriter("C:\MMA_Result.txt")

'Write khmer words in the collection to MMA_Result.txt file, each words for
one line.
For i As Integer = 1 To MMA_Result.Count
    KhmerWord = MMA_Result(i)
    MMA_Writer.Write(KhmerWord.Word)
    MMA_Writer.Write(VbCrLf)
Next
MMA_Writer.Close()
```

After running the sample code above you will get two output text files named *WBM\_Result.txt* and *MMA\_Result.txt* which contain Khmer words already segment from the given string.

## 4.2. Spell Checking

To check the spelling of Khmer sentence we need to follow up two steps, first is to segment the sentence into array of words, second is to compare the segment words to the word list in the dictionary. As mention in segmentation usage [4.1] we can use any model to segment Khmer sentence, Word Bigram Model or Maximum Matching Model.

We have two dictionaries word list; one is CHOUN NAT and Mix Dictionary (CHOUN NAT + Royal Academic Dictionary).



## Objective

Suppose you have a text file named *KhmerSentence.txt* contains Khmer string and located in the same path of your application. (ទោះយ៉ាងណាណាកម្មវិធីបានបញ្ជាក់ថាណាកម្មវិធីបានដឹងអំពីជោគវាសនារបស់ល្ខើយសិក្សាព្រឹត្តិណាមទាំងនោះទេក្រោយពីការសួរឆ្លើយព្រោះណាកម្មវិធីនាទីត្រឹមតែជាអ្នកសួរឆ្លើយប៉ុណ្ណោះ). You need to check the spelling of the sentence and give the suggested word to the word which does not exist in the Mix Dictionary.

## Coding in VB.Net

```
Dim WordBigramModelObject As New KhmerWordSegmentation.WordBigramModel
Dim WordAfterSegmented As New Collection
Dim WordToBeChecked As KhmerWordSegmentation.SegWord
Dim ErrorWordFile As System.IO.StreamWriter
Dim SuggestedWords() As String
Dim KhmerSentence As String
Dim KhmerSentenceFile As StreamReader

KhmerSentenceFile = New StreamReader(My.Application.Info.DirectoryPath & _
"\KhmerSentence.txt")
KhmerSentence = KhmerSentenceFile.ReadToEnd
KhmerSentenceFile.Close()

'Start to segment Khmer sentence in to collection of Khmer words
WordAfterSegmented =
WordBigramModelObject.getBestSegmentation(KhmerSentence,KhmerWordSegmentatio
n.KhDictionary.MIX_DICTIONARY,KhmerWordSegmentation.SEG_METHOD.WITH_EDIT_DIS
TANCE)

'i is used to name error word file
Dim i As Integer = 0

'Start check spelling
For Each WordToBeChecked In WordAfterSegmented

'False, does not exist in dictionary word list
'True, exist in dictionary word list
If Not WordToBeChecked.state Then

    'found error word in list; file name of error word is started from 1
    i += 1
    ErrorWordFile = New StreamWriter(My.Application.Info.DirectoryPath &
"\ " & i.ToString & ".txt", False, System.Text.Encoding.UTF8)

    'Write the error word in to the first line of the error word file
    ErrorWordFile.WriteLine(WordToBeChecked.Word)
    ErrorWordFile.WriteLine("-----")

    'To get sugested word
    SuggestedWords =
    WordBigramModelObject.wrdBigram.khmerDict.getWordsFromAllDictsWithED(
    WordToBeChecked.ce,WordToBeChecked.Word)

    'write the sugested words in to the error word file
    For j As Integer = 0 To SuggestedWords.Length - 1
        ErrorWordFile.WriteLine(SuggestedWords(j))
    Next
```

```
        'close error word file  
        ErrorWordFile.Close()  
        ErrorWordFile = Nothing  
    End If  
  
Next
```

After running the code if there are words which have been segmented from the sentence do not exist in the dictionary word list you will get the file name *1.txt*, *2.txt* .... *n.txt*. If your sentence has four words that do not exist in the dictionary you will get four text files (*1.txt*, *2.txt*, *3.txt* and *4.txt*); each file contains the error word in the first line and the suggested word in the next line.