

Asheninka User Documentation

H. Andrew Black

SIL International

andy_black@sil.org

21 November 2016

Version: 0.3.0 Alpha

Contents

1	Introduction	2
2	General user interface items	2
2.1	Creating a new project	2
2.2	Opening an existing project	3
2.3	Saving a project	3
2.4	Save a project as a new project	3
2.5	Setting fonts for vernacular and analysis language display	3
2.6	User interface language	3
2.7	Help menu items	3
3	Approaches to syllabification	4
3.1	CV pattern approach	4
3.1.1	CV segment inventory	5
3.1.2	CV natural classes	5
3.1.3	CV syllable patterns	6
3.1.4	CV words	6
3.1.5	Predicted vs. correct CV words	7
3.2	Sonority hierarchy approach	7
3.3	Onset-nucleus-coda approach	7
3.4	Moraic approach	7
3.5	Nuclear projection approach	8
3.6	Optimality theory approach	8
4	Converting predicted syllabification to correct syllabification	8
5	Project management	8
6	Importing and exporting words	8
6.1	Import words	9
6.2	Export words	9
7	Other tools	10
7.1	Find a word	10
7.2	Remove all words	10

7.3 Compare two implementations	10
References	12
Index	??

1 Introduction



The **Asheninka** program is a linguistic tool with two goals:

1. Explore various syllabification algorithms.
2. Provide a principled way to insert discretionary hyphens in a list of words (which can then be used for typesetting text).

The first is clearly a linguistic research goal. The second is a practical application of the first goal. While one probably will want to use IPA (IPA 2016) for the first, the second should be done using the practical orthography.

2 General user interface items


This section notes some common items found in various places throughout the user interface. It also discusses some of the more basic **File**, **Language**, **View**, and **Help** menu items.

While editing data, you add a new item by using the **Edit / New item** menu item or clicking on the  button. To delete an item, use the **Edit / Remove item** menu item or click on the  button.


A number of these have a field called “Active”. If the checkbox is checked, then this item will be used when syllabifying. If it is not checked, then it will not be used. This lets you experiment with various possibilities.

When the view contains a pane with rows and columns, you can click on a column header to sort by that column (it currently only sorts by Unicode code point). You can even hold the Shift key down and click on another column header to get a secondary sort (or on a third column to get a tertiary sort). There are some tables which cannot be sorted because the user controls the order manually. In such cases (such as **CV Syllable Patterns**), the sorting is disabled.


2.1 Creating a new project

When you want to create a new project, use **File / New** menu item or click on the  toolbar button. This brings up a standard “Save as...” dialog box. The expected file extension to use with all **Asheninka** files is “.ashedata” (for **Asheninka** data). Key the file name you want and press OK.

2.2 Opening an existing project

When you already have an existing **Asheninka** project and you want to open it into **Asheninka**, use the **File / Open...** menu item or click on the  toolbar button. This brings up a standard “File Open...” dialog box. Find the file you want and click on OK.

2.3 Saving a project

While **Asheninka** will automatically save your work about every 30 seconds, you can also use the **File / Save** menu item or click on the  button in the toolbar.

2.4 Save a project as a new project

When you decide to save the current project with a different name, use the **File / Save As...** menu item.

2.5 Setting fonts for vernacular and analysis language display

Use the **Language** menu item to set the font information for displaying the vernacular and/or the analysis languages. Currently, only font family, font size, and font style can be set. We do plan to include color and sorting information in a later version.

2.6 User interface language

You can set the user interface language by using the **View / Change the interface language** menu item. This brings up a dialog box showing the current interface language in a drop down chooser. Click on the chooser's drop down button to see other interface language choices. The choices given use the name of the language in the current interface language (so if the current interface language is English, then it will show “Spanish” as an option; if the current interface language is Spanish, it will show English as “inglés”).

The current version has English and a rough, most likely often inaccurate version of Spanish. Any corrections to the Spanish are welcome.

2.7 **Help** menu items

Currently, there are three **Help** menu items:

1. **User Documentation** which shows this document.
2. **Introduction to syllabification** which has some general discussion of syllabification issues.
3. **About Asheninka** which has some standard information about the current version of **Asheninka**.

3 Approaches to syllabification


Asheninka will eventually offer six different syllabification algorithms or approaches. Only one is implemented in the current version.

The various approaches or algorithms are explained and illustrated in the “Overview” document. You can read this document by clicking [here](#) or by using the **Help / Introduction to syllabification** menu item.

3.1 CV pattern approach

In the CV pattern approach, you need to define the following items:

1. segments (section 3.1.1)
2. natural classes (section 3.1.2)
3. syllable patterns (section 3.1.3)

After that, you import a word list (see section 6) or enter a list of words by hand. Then you use the **Tools / Syllabify Words** menu item or click on the  tool bar button. This will apply the algorithm of the CV pattern approach to all the words. You can then see the results in the **CV Words** view. See section 3.1.4 for ideas on how to check the results, among other things.

The main “game” to play with the CV pattern approach in **Asheninka** is to adjust the segment inventory, natural classes, and syllable patterns so that one gets most, if not all, words to syllabify correctly.

The main algorithm is as follows:

1. It does a left-to-right sweep of the word, trying to find the sequence of segments which covers the whole word. It uses the graphemes defined in the segment information to determine a match with a segment. If it cannot find a sequence that covers the whole word, it reports an error of “**Failure: could not parse into segments beginning at 'some characters'**” and quits. The 'some characters' part indicates the place in the word where it could not find a character or character sequence that matched any graphemes in any of the segments. Most likely, either there was a typo in the word or a grapheme is missing from some segment.
2. If it succeeded in finding a sequence of segments, it then performs a left-to-right sweep of the segments it posited, trying to find a sequence of natural classes that covers the entire sequence of segments. It uses the set of segments or natural (sub-)classes defined within a natural class to determine a match with a natural class. If it cannot find a sequence of natural classes that covers the entire sequence of posited segments, it reports an error of “**Failure: could not parse into natural classes; did find classes 'some classes' covering graphemes 'some graphemes'**” and quits. The 'some classes' part indicates the sequence of natural classes that were found and the 'some graphemes' part indicates which graphemes were included in those natural classes. Hopefully this will help you figure out why other natural classes were not found for this word.

3. If it succeeded in finding a sequence of natural classes, it performs a left-to-right sweep of the sequence of the natural classes it posited, trying to find a sequence of syllable patterns that covers the entire sequence. It tries the syllable patterns in the order they are arranged in the **CV Syllable Patterns** view (see section 3.1.3 below). If it cannot find a sequence of syllable patterns that covers the entire sequence of posited natural classes, it reports an error of “**Failure: could not parse natural classes into syllables**” and quits.
4. If it succeeded in finding a sequence of syllable patterns, it outputs the syllabification into the **Predicted Syllable Breaks** field and reports “**Success**”.

We now give more information on the various views available in the user interface for the CV pattern approach.

3.1.1 CV segment inventory

Make sure that what is in the **CV Segment Inventory** view covers all the segments you have in the orthography. Currently, **Asheninka** uses a “longest match” algorithm for figuring out which segment a letter (or sequence of letters) belongs to. This could be a problem in some cases where a language has both codas and onsets and a given sequence of consonants might be ambiguous between various possibilities. So if you have a **k**, a **kl**, an **l**, and an **ll**, then something like **aklli** could be ambiguous between **ak.lli** and **akl.li**. The current version will only try **akl.li** because **kl** is the longest match in a left-to-right sweep. We do plan to add environments so one can limit certain graphemes to specified environments.

In the “Segment” field, key a letter or letters that will help you remember this segment. For example, a low central unrounded vowel might be keyed as **a** while a voiceless alveopalatal affricate might be keyed as **tʃ** or as **ch**.

The “Description” field is for your benefit. Key whatever helps you (and anyone looking at your data) know what the segment is. For example, you could key its phonetic or phonological description.


In the “Graphemes” field, key all the ways the particular segment appears in your orthography, including any upper case forms. Separate each one by a space.

3.1.2 CV natural classes

Make sure that in the **CV Natural Classes** view, every segment is in a natural class. With the current version, a given segment should only be in one class. We do plan to allow a given segment to be in more than one class in a later version.

The “Name” field is for you to define how this natural class will appear in syllable patterns. We suggest using something short and if there are more than one character in it, make the first one be capitalized and the rest be in lower case.¹ The “Description” field is for your use to document anything special about this natural class.


¹This is not required, it is just a suggestion.

The “Segment or Natural Class” field has a chooser button on the far right which looks like . Click on it to bring up a dialog box which lets you select segments and/or natural classes which belong to this natural class. Note that it is possible to insert one entire natural class within another natural class. For example, if you have a class of nasal consonants, you can include that class within another class that has all consonants.

3.1.3 CV syllable patterns

Use the **CV Syllable Patterns** view to create CV syllable patterns that cover the kinds of syllables you think the language has. Be sure to allow for vowel-initial syllables if there are vowel-initial syllables in the language.

The “Name” field is for you to define a short name for this syllable pattern. We suggest using something short. The “Description” field is for your use to document anything special about this syllable pattern.

The “Natural Classes” field shows the sequence of any natural classes this pattern consists of. It has a chooser button on the far right which looks like . Click on it to bring up a dialog box which lets you select natural classes which constitute this syllable pattern. This chooser consists of one or more drop-down boxes. Click on the drop-down arrow in the box. It will show you the list of possible natural classes to choose from. In addition, if this is the very first drop-down box or the last one currently being shown, it will also include “Word boundary”. When you choose “Word boundary”, the pattern shown above the drop-down box(es), will show “#” to indicate a word boundary.

Finally, the order in which the syllable patterns occur is important, as mentioned in section 3.1. Recall that with the CV pattern approach, as **Asheninka** parses a word, it tries to match the patterns in the order in which they appear. You control the order of the syllable patterns by clicking on a pattern in the middle pane and then using the up and down arrows to change its order. Note that the normal way of sorting by a column via clicking on a column header is disabled for this view.

3.1.4 CV words

In the **CV Words** view, the middle pane of the display shows all the words with three other columns:

1. **Predicted Syllable Breaks**
2. **Correct Syllable Breaks**
3. **Parser Result**


When you click on a word in the middle pane, the same fields are shown on the right where you can edit them.

The **Predicted Syllable Breaks** column/field contains the result of the last time you ran the **Syllabify Words** tool. When it is empty, it means either that the parser has never been invoked or that the parser failed to produce a result. If the parser

failed to produce a result, the **Parser Result** column/field should have some kind of explanation for where the parser ran into a problem (this will be in red).

When the **Predicted Syllable Breaks** column/field contains a result, it shows how this word was syllabified the last time you ran the **Syllabify Words** tool, given the state of the segments, natural classes, and syllable patterns when it was invoked. When there is a result in this column/field, then the **Parser Result** column/field should have “Success” (in green).

Remember that you can click on a column header to sort the contents of the column. If you click on the **Parser Result** header, then any error messages should show at the top. You might find this useful when figuring out how to set the set of segments, natural classes, and syllable patterns to get a desired result.

The **Correct Syllable Breaks** column/field contains what you have indicated to be the correct syllabification for this word. It may be filled in if you imported words from a **ParaText** hyphenatedWords.txt file and that file indicated that this word's hyphenation had been approved. Otherwise, this column/field is expected to be blank until you either manually enter the correct syllabification or run the **Convert Predicted to Correct Syllabification** tool by using **Tools / Convert predicted to correct syllabification** menu item or clicking on the  tool bar button.

When you run the **Convert Predicted to Correct Syllabification** tool, it brings up a dialog box listing all the words which have a non-empty **Predicted Syllable Breaks** field. See section 4 for more on this.

3.1.5 Predicted vs. correct CV words

The **Predicted vs. Correct CV Words** view shows any words which have both a predicted value and a correct value and, in addition, the two values differ. This is intended to give you a way to quickly see how the predictions of the current set of segments, natural classes, and syllable patterns differ from the expected results. In this view, by the way, the predicted and correct words are aligned in pairs with the predicted syllabification immediately above the correct syllabification. This is an attempt to make it easier to see the differences between the two.

3.2 Sonority hierarchy approach

This approach has not been implemented yet.

3.3 Onset-nucleus-coda approach

This approach has not been implemented yet.

3.4 Moraic approach

This approach has not been implemented yet.

3.5 Nuclear projection approach

This approach has not been implemented yet.

3.6 Optimality theory approach

This approach has not been implemented yet.

4 Converting predicted syllabification to correct syllabification

When you run the **Convert Predicted to Correct Syllabification** tool, it brings up a dialog box listing all the words which have a non-empty **Predicted Syllable Breaks** field. You can select which words should be converted by manually checking the box before them² or by clicking on the checkbox in the header row and choosing to “Select All”, “Clear All”, or “Toggle”:

1. “Select All” checks every word.
2. “Clear All” unchecks every word.
3. “Toggle” makes every word that is checked be unchecked and every word that is unchecked be checked.

When you click on the OK button, **Asheninka** will copy the value in the **Predicted Syllable Breaks** field of every word that is checked to the corresponding **Correct Syllable Breaks** field.

5 Project management

Feel free to make frequent (labeled) backups via **File / Project Management / Back up this project** menu item. You can give it a descriptive label. You won't see the label on the file name, though. Rather, you'll see it when you go to **File / Project Management / Restore a project** menu item.

You can make a backup, try something else (e.g., add a segment or class; or reorder the syllable patterns or add a new one), and see how it goes. If it's worse, just restore to the previous state. If it's better, make a backup of that and go on.

6 Importing and exporting words

When using **Asheninka**, of course, you need words to parse. Once they are parsed and you have set the set of correct syllabification for them (see section 4), you may want to export the set of words to be used by a typesetting tool. Such a tool can use the syllabified words as a list of words with discretionary hyphens so that it can know when to hyphenate long words at ends of lines.

²To check the box, either click in it or click in the row and press the space bar.

6.1 Import words

Asheninka allows you to import a list of words in the following forms:

1. A text file with one word per line
2. A word list exported from **ParaText**
3. The hyphenatedWords.txt file from **ParaText**
4. A word list exported from **FieldWorks Language Explorer** (aka FLEEx) (in tab-delimited form)

You can use any or all of these to add words to **Asheninka**'s list of words. A given word form will only be inserted once. By “word form” we mean that word pairs such as **achto** and **Achto** will be considered different since their capitalization is distinct.

To import a word list, use **File / Import Words** menu item and then choose the type of import to use. In each case, a standard file open dialog will appear. Find the file that has the format you need, choose it, and click on “Open”. Depending on the number of words in the imported file, this may take a while to complete.

If you are working on a word list found in **ParaText**, we recommend using the hyphenatedWords.txt file method. This file includes both upper and lower case forms of words and also will include any hyphenation the **ParaText** user has manually approved. At least with some versions of **ParaText**, this file can be found in the “My Paratext Projects” folder and then in the folder of your project name.

Newly added words will appear in the **CV Words** view with a **Parser Result** showing “**Untested**”. This means that the parser has not yet been run on the word.

6.2 Export words

When you want to export the word list in **Asheninka** to some kind of typesetting tool, use the **File / Export Hyphenated Words** menu item. The list of hyphenated words can be exported in three formats:

1. Export for **InDesign** (simple list)
2. Export for **ParaText** (hyphenatedWords.txt)
3. Export for **XLingPaper** (hyphenations exception file)

Asheninka exports hyphenated words in the following way:


1. When the word has a value in **Correct Syllable Breaks**, that value is used.
2. If **Correct Syllable Breaks** is empty, then if there is a value in **Predicted Syllable Breaks**, it uses that value.
3. Otherwise, it does not export the word.

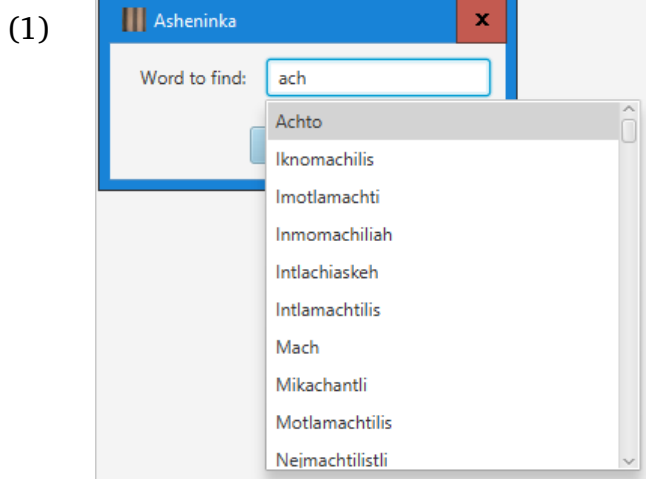
It also will replace the periods used to demarcate syllables with an equals sign for both **ParaText** and **InDesign**. It replaces the periods with a hyphen for the **XLingPaper** output.

7 Other tools

Asheninka also offers three other tools not previously covered.

7.1 Find a word

Use the **Tools / Find Word** menu item or key **Ctrl F** or click on the  button on the toolbar to find a particular word. This brings up a dialog box which allows you to start typing the word you are looking for. As you type, it has a drop down area showing all words which contain the sequence of characters you have typed. Note that it shows all words which have this sequence anywhere in them, not just at the beginning. For example, with one project when I typed “**ach**”, what is in example (1) showed.



This can be very useful not only for finding a particular word, but also for looking for particular sequences of characters.

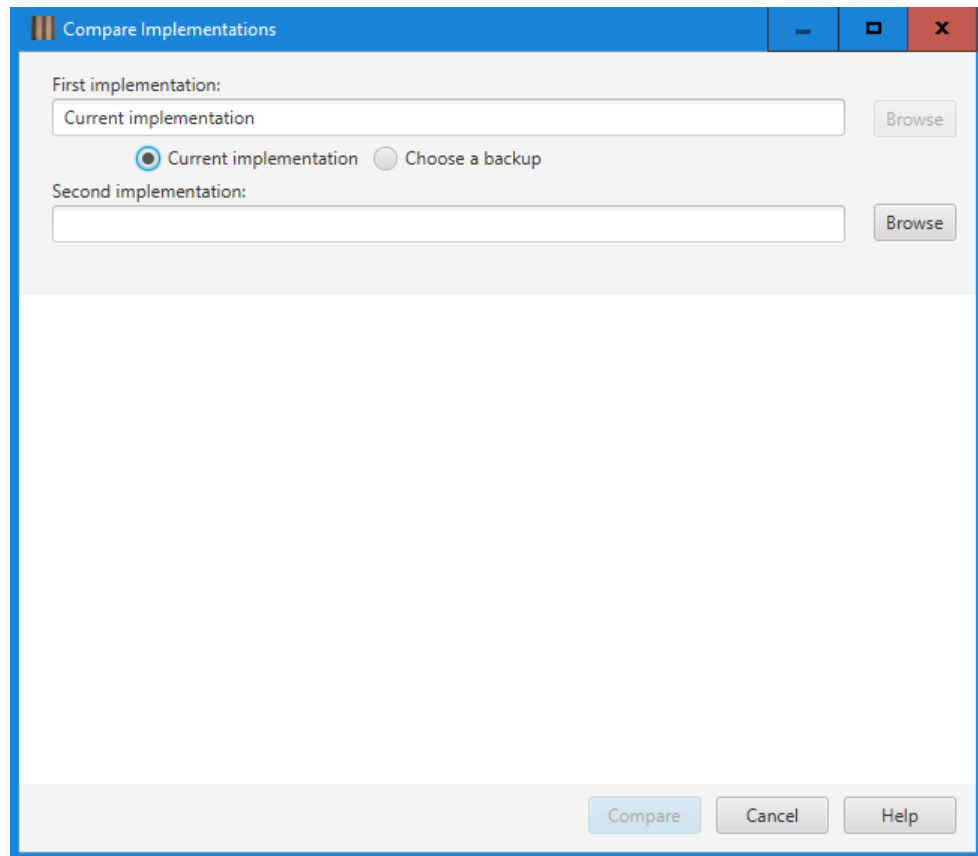
7.2 Remove all words

Use the **Tools / Remove all words** menu item to completely clear the list of words in the **CV Words** view. When you invoke this menu item, it will ask you if you are sure you want to do this before performing it.

7.3 Compare two implementations

Asheninka offers a way to more easily see what some changes to an approach have on how the words are syllabified. Use the **Tools / Compare Implementations** menu item to do this. It brings up a dialog box which looks like what is in example (2).

(2)



You can compare the current implementation (i.e., what is showing in the user interface right now) with what is in a backup file (see section 5) or you can compare two different backups. As you can see in example (2), the current implementation is set by default. To compare two backups, click on the “Choose a backup” radio button.

Whenever you need to select a backup file, click on the “Browse” button. This will bring up the “Restore a Project” dialog. Choose the backup you want and click OK.

When you have chosen the two implementations to compare, click on the “Compare” button. Depending on how many differences there are, the comparison process can take quite a long time so please be patient. When it is done, a report is shown in the bottom part of the dialog. When I did this for one project, I got what is shown in example (3).

(3)

Compare Implementations

First implementation:

☒ Current implementation ☐ Choose a backup

Second implementation:

CV Approach Comparison between:

1. Current implementation
2. C:\Users\Andy Black\Documents\My Asheninka\Nahuatl de la Sierra Negra\Backups\Nahuatl de la Sierra Negra\220160802-151207.ashebackup (After adding kua, kue, kui as V (work-around for kw))

Performed on November 11, 2016 11:38 AM

Segment Inventory

Both have the same segment inventory.

Natural Classes

The following natural classes differ between the two.

First	Second
K (kua, kue, kui)	
V (a, aa, e, ee, i, ii, o, oo, u)	V (a, aa, e, ee, i, ii, kua, kue, kui, o, oo, u)

Syllable Patterns

The following syllable patterns differ between the two.

First	Second
K (K)	
KC (K, C)	

The order of the syllable patterns is shown below.

First	Second
K (K)	CV (C, V)
CV (C, V)	CVC (C, V, C)

The dialog box will stay visible until you close it (e.g., by clicking on the “Cancel” button). This lets you examine the report while still being able to make changes to the data in the user interface.

References

IPA. 2016. International Phonetic Association.
<https://www.internationalphoneticassociation.org/> (accessed 9 November 2016)

Index

- Active field, 2
- Add data item, 2
- Analysis language, *see* Language, Analysis.
- Compare two implementations, ??
- CV pattern approach, 4
 - Algorithm, 4
 - Natural classes, ??
 - Predicted vs. correct syllabification, ??
 - Segments, ??
 - Syllable Patterns, ??
 - Words, ??
- Delete data item, 2
- Failure messages
 - CV Pattern approach
 - Parse into natural classes, 4
 - Parse into segments, 4
 - Parse into syllables, ??
- Fonts, *see* Language.
- Help, 3
 - Introduction to syllabification, 3, 4
 - User documentation, 3
- Insert data item, *see* Add data item.
- Language
 - Analysis, 3
 - User interface, 3
 - Vernacular, 3
- Moraic approach, ??
- New project, *see* Project, New.
- Nuclear projection approach, ??
- Onset-nucleus-coda approach, ??
- Open a project, *see* Project, Open.
- Optimality Theory approach, ??
- Ordering data in display, *see* Sorting data in display.
- Parsing
 - Characters into segments, ??
 - CV algorithm, *see* CV pattern approach, Algorithm.
 - Parsing result failure messages, 4
- Predicted syllabification
 - Comparing with correct, ??
 - Converting to correct, ??
- Project
 - Management
 - Backup, ??
 - Restore, ??
 - New, 2
 - Open, 3
 - Save, 3
 - Save project as a new project, 3
- Remove data item, *see* Delete data item.
- Save a project, *see* Project, Save.
- Save a project as a new project, *see* Project, Save project as a new project.
- Sonority hierarchy approach, ??

Sorting data in display, 2

Syllabification, *see* Help, Introduction to syllabification.

User documentation, *see* Help, User documentation.

User interface language, *see* Language, User interface.

Vernacular language, *see* Language, Vernacular.

Words

Export, ??

 To **ParaText**, ??

 To a text file (plain text), ??

 To **InDesign**, ??

 To **XLingPaper**, ??

Find a word, ??

Import, ??

 From **ParaText**, ??

 From a text file (plain text), ??

 From **FLEx**, ??

Remove all words, ??