

# **TraDisc Manual**

**Christoph Malisi**

---

# TraDisc Manual

Christoph Malisi

---

---

## Table of Contents

<a href="#">Introduction</a>	v
<a href="#">I. TraDisc</a>	6
<a href="#">1. The XML Corpus Format</a>	7
<a href="#">Specifying the XML Input Format of the Corpus</a>	7
<a href="#">The TraDisc Standard Format</a>	8
<a href="#">Saving and Loading an XML Input Format</a>	8
<a href="#">2. The Schema</a>	10
<a href="#">The Schema Editor</a>	10
<a href="#">The Cell Editor</a>	11
<a href="#">The Markable Editor</a>	11
<a href="#">The Schema Overview</a>	13
<a href="#">Saving and Loading a Schema</a>	14
<a href="#">Beginning a New Schema</a>	14
<a href="#">Merging one Schema with Another</a>	14
<a href="#">3. Annotations</a>	15
<a href="#">Beginning a New Annotation</a>	15
<a href="#">The Annotation Text Window</a>	15
<a href="#">The Annotation Control Window</a>	15
<a href="#">Navigation</a>	16
<a href="#">Actually Annotating: Selecting Tags</a>	17
<a href="#">Saving and Loading Annotations</a>	17
<a href="#">Splitting and Merging Annotations</a>	18
<a href="#">Printing Annotations</a>	18
<a href="#">4. Evaluating and Analyzing Annotations</a>	19
<a href="#">The Evaluation Table</a>	19
<a href="#">Normalizing the Evaluation Values</a>	19
<a href="#">Selecting the Markables to Evaluate</a>	20
<a href="#">Exporting Evaluation Data</a>	21
<a href="#">Combining the Evaluation Table with Loaded Evaluation Files</a>	21
<a href="#">Evaluating a Tag</a>	21
<a href="#">Overview of the Schema Usage</a>	22
<a href="#">The Total Number of Annotated Markables</a>	22
<a href="#">The Number of All Annotated Markables in the Entire Corpus</a>	23
<a href="#">Distribution of Annotated Markables over Parts of the Corpus</a>	23
<a href="#">Calculating the Complexity Score of an Annotation</a>	24
<a href="#">The Complexity Score Table</a>	24
<a href="#">Calculating the Complexity Score for an Entire Annotation</a>	25
<a href="#">Distributing the Complexity to Parts of the Corpus</a>	25
<a href="#">Creating Junctograms</a>	26
<a href="#">II. Tokenizer</a>	28
<a href="#">5. Using Tokenizer</a>	29
<a href="#">Tokenizing the Text</a>	30

---

<a href="#">Creating the XML File</a> .....	31
---	----

---

# Introduction

TraDisc is a program designed to annotate linguistic corpora in an XML format. (If you would like to process a text that is not yet in an XML format, you can use the program [Tokenizer](#) to attain a simple XML format.)

TraDisc was initially developed in order to identify and annotate junctors (sentence connectives) in a corpus, however it can also annotate any other feature in a corpus.

---

# Part I. TraDisc

TraDisc permits the user to annotate tokens of a corpus with functions or features of his choice. Such features can be either one- or two-dimensional. This handbook discusses an example in which the nominative personal pronouns of a text are to be annotated. The feature to be annotated consists of the dimensions *person* and *number*. The personal pronoun *he*, for example, could be annotated with the feature *3rd person-singular*.

The dimensions of the features form the columns and rows of a table, the [TraDisc schema](#). Our example thus has three columns, *1st person*, *2nd person* and *3rd person*, and two rows, *singular* and *plural*.

Tokens (mostly words) can be entered into the fields of this table. TraDisc offers help in actually annotating the tokens and offers various possibilities of moving through the text in order to easily find markables. (Tokens listed in the schema are called markables. In our example, these are the potential personal pronouns *I*, *you*, *he*, *she*, *it*, *we*, *you*.) Furthermore, TraDisc offers various [instruments for analysis and evaluation](#), which evaluate the text according to the chosen criteria of annotation, e.g. the number and type of markables.

---

---

# Chapter 1. The XML Corpus Format

## Note

If you would like to use TraDisc to process a text already in the TraDisc standard XML format, i.e., if the text has been put in XML format by the Tokenizer, then knowledge of this chapter is not absolutely necessary. The settings for this XML format are set to default.

## Specifying the XML Input Format of the Corpus

In order to be able to work with a corpus in TraDisc, it must be available in an XML format. This format must be entered into TraDisc in order for it to recognize the text. In the menu item "Input Format → Edit corpus input format" the corresponding input dialog appears. It consists of two major cards: **tokens** and **other elements to display**.

The XML element containing the tokens of the corpus must be specified under the card **Tokens**. The XML tag of the element must be noted (in the field **XML tag name**), as well as the name of the XML attribute containing the specific word that is to be displayed in the TraDisc text field (in the input field **XML attribute to display**). Both of these specifications must always be given. Further XML attributes and attribute values always to be present in the XML element of a token in the XML format of the corpus can be set. To do this, enter the name and value of the attribute with the button **Add**. Entries to the list of required attributes can be erased: select an item and click on **remove**. By clicking on **Apply changes**, the changes in the input format of the corpus are updated, and the new format appears in the preview window **Current XML input format**.

The card **Other elements to display** specifies XML elements that don't contain any tokens, but are still to be shown in the TraDisc text window. Usually these are the XML elements that contain the punctuation of the text being annotated. As with the XML elements for the tokens, the XML tag and the attribute to be represented (which contains the punctuation mark) must be specified. There is also the option of entering further attributes here if needed. The entry fields are equivalent to those of the card **tokens**.

The input format dialog box with the entered values of the TraDisc standard format.

## The TraDisc Standard Format

An XML format has been pre-set for starting TraDisc, the TraDisc standard format. The tag of the XML elements containing tokens is `token`, the attribute containing the actual token (the word), is `f`. There are no further required attributes of the token XML element. Punctuation marks are saved in XML elements with the tag `other`. As these are to be displayed, the tag `other` is entered in **Other elements to display**.

If the corpus to be processed is available in the TraDisc standard format, nothing needs to be changed in the menu "Input Format".

**Example of a corpus in the TraDisc standard format.** The corpus text is: *Welcome to TraDisc!* The corresponding XML corpus looks as follows:

```
<corpus>
  <token f="Welcome"/>
  <token f="to"/>
  <token f="TraDisc"/>
  <other f="!"/>
</corpus>
```

## Saving and Loading an XML Input Format



---

In order to save the current input format, select "Input Format → Save input format specifications..." in the menu. A file saving dialog box will open. In order to load a saved input format, select "Input Format → Load input format specifications...", e.g. if multiple corpora are to be processed with the same input format.

---

## Chapter 2. The Schema

TraDisc can annotate the tokens of a corpus with two-dimensional functions or features. Tokens possibly to be annotated, i.e., which have a potentially annotatable feature, are called “markables”. In order for the program to know which features are possible for a specific markable, a TraDisc schema must be made. As TraDisc annotates two-dimensional features, the markables are entered into a two-dimensional table. The row and column titles of this table correspond to the features this markable can assume (cf. also [here](#)).







If one wanted to annotate the nominative personal pronouns of a text with person and number, a corresponding simple schema table could look like this:


	1st person	2nd person	3rd person
singular	I	you	he she it
plural	we	you	they

This schema table is entered into TraDisc by using the schema editor.

### The Schema Editor

The schema editor is opened in the menu under "Schema → Edit...". The currently loaded Schema is then shown. If no schema is yet loaded, a new empty schema will be shown, with two columns called "column 1" and "column 2", and two rows called "row 1" and "row 2". In order to adjust the schema to your needs, you can add and delete columns and rows with the following buttons:

-  Adds a new row at the bottom of the table
-  Adds a new row above the row containing the selected field
-  Deletes the row containing the selected field
-  Adds a new column on the right end of the table
-  Adds a new column to the left of the column containing the selected field
-  Deletes the column containing the selected field

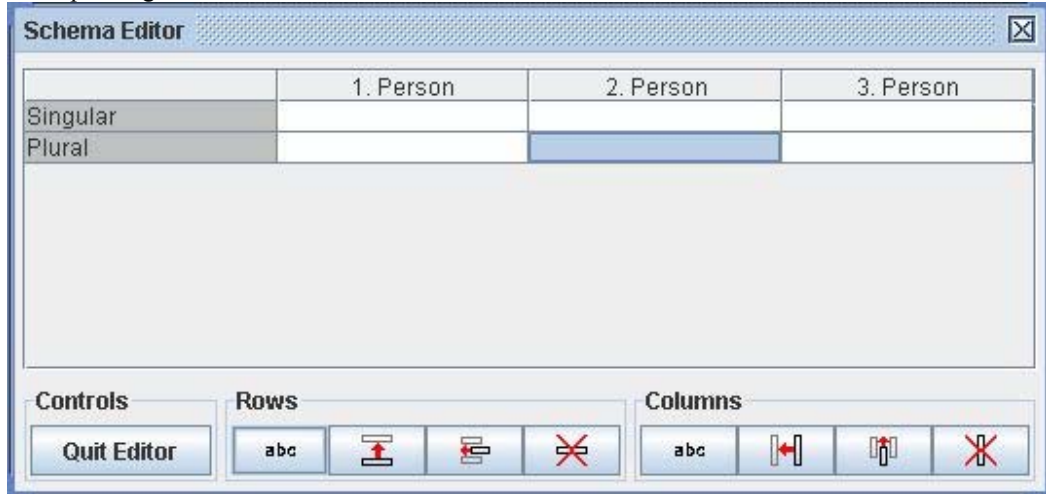
The selected field is marked blue. You can navigate between the fields with the arrow keys. The row and column titles can be changed with the button . The button in the frame "rows" changes the name of the selected row, the one in the frame "columns" changes the name of the selected column. If you would like to use the [printing function](#) of TraDisc, it is helpful to put an abbreviation followed by a space in front of the actual column name.

#### Note

---

Once an annotation has been started, the dimensions of the schema can **no longer** be changed. This means that no further rows or columns can be added or deleted, nor can the names of the rows and columns be changed. The entries in the fields of the table, however, can be edited as described below.

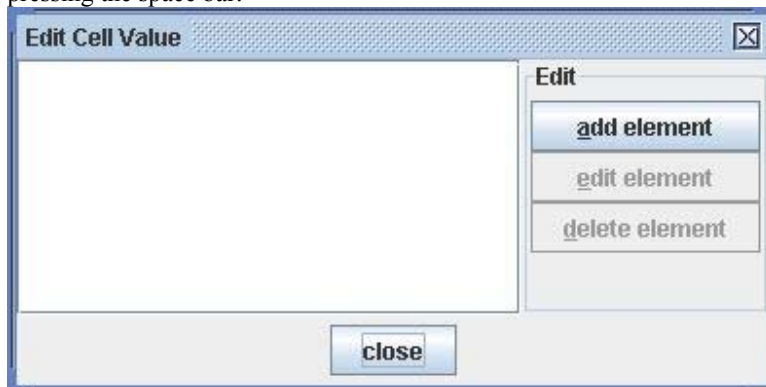
For our example with the personal pronouns, a new column must be entered into the empty schema, and the row and column titles must be renamed and provided with the features of person and number. The corresponding table in the schema editor looks like this:



In order to fill the fields of the schema (the schema cells) with entries, the cell editor must be opened.

## The Cell Editor

The cell editor for changing the entries of a field in the schema table is opened by a click on the corresponding field; you can also select the cell with the arrow keys and then open the cell editor by pressing the space bar.



*The cell editor of a table field without any entries.*

New markables can be added by clicking on **add element**. Entered markables can be changed by selecting the entry and then clicking on **edit element**. In both cases, the *markable editor* opens. If a markable is to be deleted from the cell, select it and then click on **delete element**.

## The Markable Editor

An empty markable editor with the TraDisc standard input format.

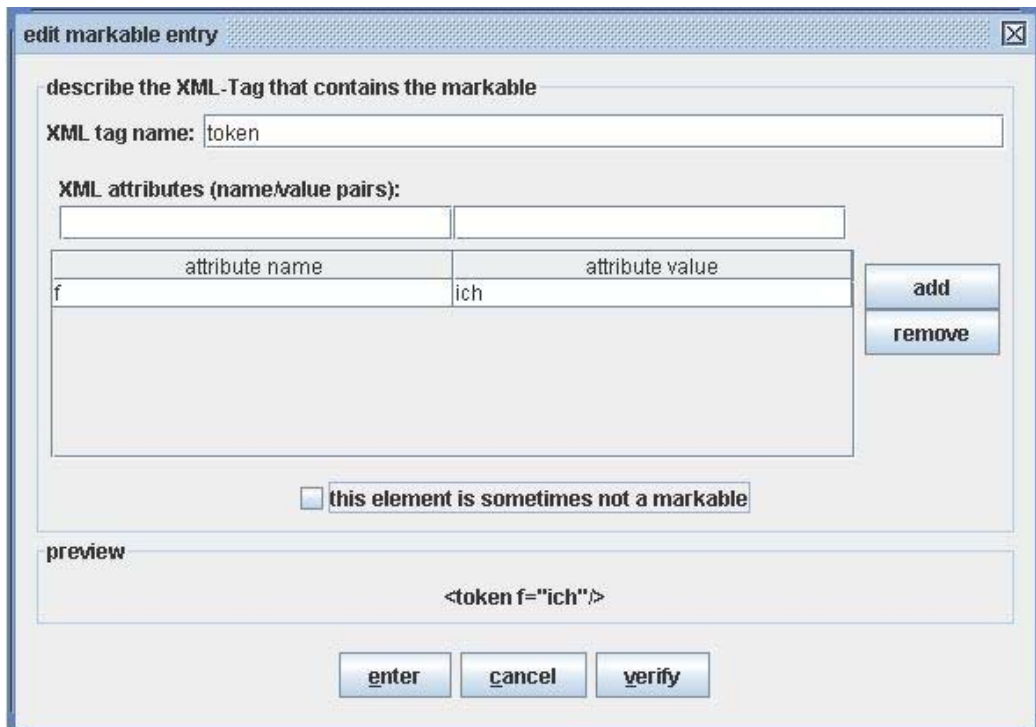
In order for TraDisc to be able to find markables in the corpus, they must first be entered into the schema table in the [XML format](#) of the corpus.

Enter the XML tag containing the tokens into the field **XML tag name**. As the name of this tag is specified in the XML input format, TraDisc will suggest this name. In the [TraDisc standard format](#) the name is `token`. Furthermore, the XML attributes a token must have must be entered. Usually, that is the XML attribute containing the actual word (the "display attribute"); in the TraDisc standard format it is `f`. These entries take place in the text field **XML attributes (name/value pairs)**. Further markables can be added by clicking on the corresponding button. The attribute values are not distinguished by capitalization, so it is sufficient to enter one version into the schema. The asterisk `*` can be used as a variable in the attribute value. `*` can be at the beginning and/or end of an attribute value and is a variable for any succession of letters. If e.g. the value of an attribute was `laugh*`, then all tokens whose corresponding attribute had a value beginning with `laugh` would be recognized: `laughing`, `laughs`, `laughed`, `laughable`....

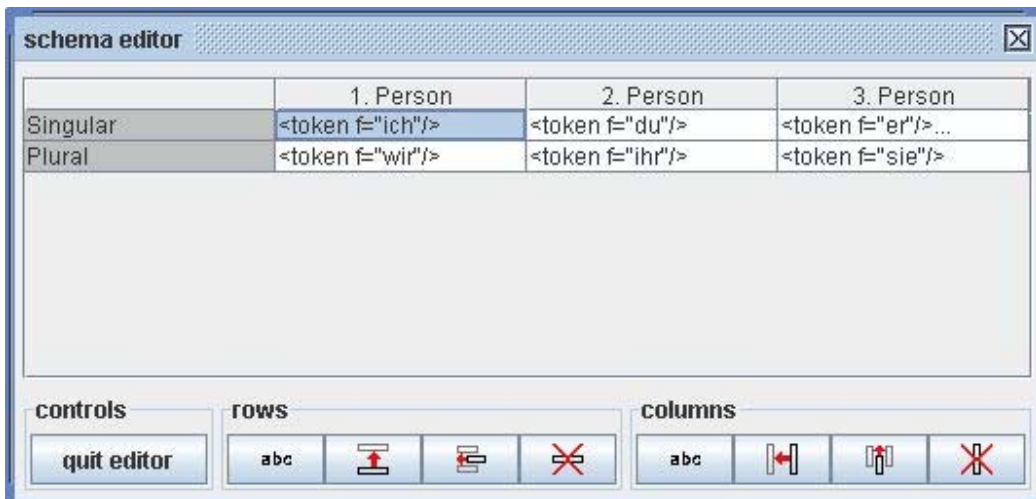
In order to delete an attribute, select it from the list and click **remove**. If you would like to check whether a markable has the correct format and get a preview of the markable XML element, click **verify**. A preview window will appear.

Some markables don't have the function or feature being annotated in all contexts. For example, the markable `you` can also appear in a context in which it is not a nominative personal pronoun. These markables are not always to be annotated with a function from the table. If this is the case, the box **this element is sometimes not a markable** must be checked.

In our example concerning the personal pronouns, the token `I` must be entered into the upper left cell of the table by selecting the cell and clicking on **add element** in the cell editor. Then, in the markable editor, an attribute called `f` and the value `I` are added. With a click on **verify**, you can see the desired XML element `<token f="I" />` in the preview window. As `I` always functions as a nominative personal pronoun, the box **this element is sometimes not a markable** is not checked.



The other entries to the table are added correspondingly. As *it* and *you* can also appear in other functions (dative or accusative personal pronouns) the field *this element is sometimes not a markable* is marked here. The schema editor with all entries then looks like this:



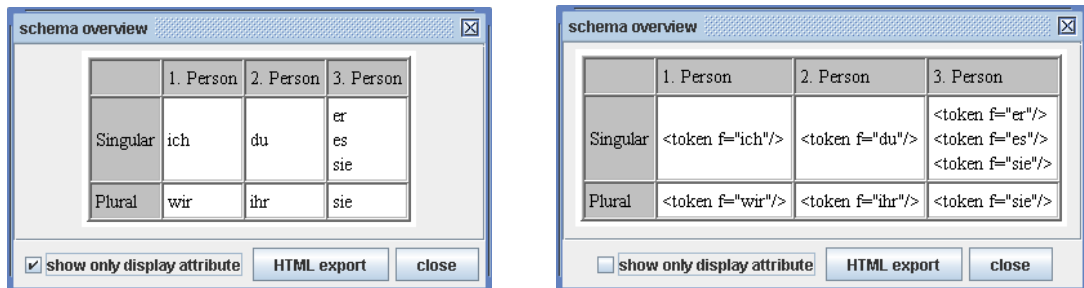
The cell "3rd person - singular" has three entries. Only the first one is visible; the following ones are indicated by the points of ellipsis ... .

## The Schema Overview

Open the overview under the menu item "Schema → Overview". The schema table is then presented with all entries. You can choose between two modes of presentation by the box *show only display attribute*. If it is checked, only the values of the display attribute of the entered markable are shown. In

---

the TraDisc standard format this is f, so only the word I is displayed of a table entry `<token f="I"/>`. If the box is not checked, the entire XML element is displayed:



## Saving and Loading a Schema

Schema tables can be saved and used for multiple annotations. To do this, select the menu point "Schema → Save" or "Schema → Save as...", respectively. A saved schema can be opened by selecting "Schema → Open". (Schemas are saved in a special TraDisc XML format.)

## Beginning a New Schema

Under the menu item "Schema → New", a new, empty Schema is loaded in TraDisc and the schema editor. If there are changes to the present schema which haven't yet been saved, TraDisc will indicate this to the user and give the option of saving first.

## Merging one Schema with Another

TraDisc offers the option of adding a saved schema to the presently active one, or, respectively, of merging both schemata. The new schema will then contain the entries of both initial schemata. The schema being added must fulfill the following conditions: it must have the same dimensions, i.e. the same number of columns and rows and the same names for those. Furthermore, the XML input format of the markables must be the same (cf. [here](#)). All entries of the added schema which are not part of the active schema are added to it. The menu item "Schema → Merge with schema..." allows to select a saved schema and merge it with the active schema in the described way.

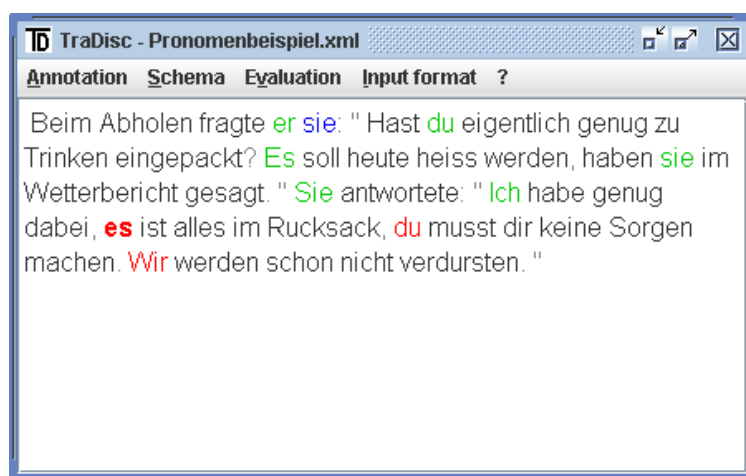
---

# Chapter 3. Annotations

## Beginning a New Annotation

In order to begin a new annotation, select "Annotation → New..." in the menu. A file selecting dialog box will open, where the user must select the corpus file he or she wants to work with. It must correspond to the [XML corpus format](#) already specified. The corpus text is then loaded and displayed in the annotation text window. The annotation control window opens as well.

## The Annotation Text Window



The corpus to be processed is depicted in the annotation text window. Markables, i.e. tokens, from the [schema](#), are marked in various colors:

- Red: The markable is not yet annotated.
- Green: The markable is annotated with a tag (i.e. with a function or feature corresponding to the row or column name of one of the table cells containing the markable in the schema. In the example of the personal pronouns, one tag is "singular-1st person", another "singular-2nd person", etc.).
- Blue: In this context, the markable has none of the features being annotated (cf. [here \[12\]](#)).

Tokens which do not appear in the schema and thus aren't markables are displayed in black. All other displayed elements of the text (usually punctuation, cf. [here \[6\]](#)) are displayed in gray. The active element is in bold print. In order to activate an element, either use the navigation options of the annotation control window, or select the desired element with a double click.

---

## The Annotation Control Window



The annotation control window is only visible when an annotation is open. It has three fields: the field **token information** shows the number of the presently activated token, counting from the beginning of the text. The XML element of the token is indicated as well (i.e. the XML element of the token in the original XML corpus file, cf. [here](#)). The field **navigation** offers various options of [navigation](#) through the text. And finally, a tag for the presently active markable can be selected in the field **select tag** (if a markable is presently active.).

## Navigation

The selection box **active markable** allows you to select which markables in a text you would like to navigate through. All markables entered into the [schema](#) can be selected; there are three further entries ALL MARKABLES, ALL TOKENS and ALL ELEMENTS.

After opening an annotation, the entry ALL MARKABLES is selected. The navigation buttons are therefore distributed as follows:


- Jumps to the first/last markable of a text.
- Jumps to the next markable that hasn't been annotated in the respective direction (i.e. to the next red markable).
- Jumps to the next markable having various annotation options (tags) in the respective direction. All markables not annotated so far which lie in between and have only one possible tag are automatically annotated and skipped with this button. (These markables can be annotated distinctly, as they are cited in only one field of the [schema](#), and the field **this element is sometimes not a markable** is not checked. They are thus annotated with the tag corresponding to the column and row of the one table entry.)
- Jumps to the next markable in the respective direction.



---

If a specific markable is selected in the box `active markable` (i.e. not `ALL MARKABLES`, `ALL TOKENS` or `ALL ELEMENTS`), you will only navigate through markables of this type in the text, all others will be skipped and ignored. If `ALL TOKENS` is selected, the navigation will comprise all tokens of the text, if they are markables or not (thus including the ones in black). The selection `ALL ELEMENTS` permits navigation through all displayed XML elements of the text, not only through the tokens. This especially means that one can navigate through punctuation marks.

The option `jump to token number` enables you to jump to a specific token. The number of the token has to be entered into the corresponding field, and a click on `go!` will bring the annotation window to the desired token.

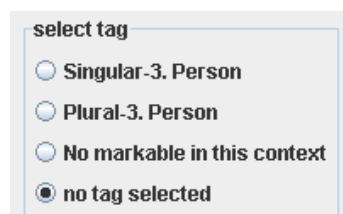
The right and left arrow keys can also be used to navigate through the text. They correspond to the buttons . A double click of the mouse on a token or other displayed element in the text window will make the annotation jump to this element.

## Actually Annotating: Selecting Tags

If the active element is a markable, it can be annotated. All entries in the schema that fit the markable are displayed for selection in the annotation control window in the frame `select tag`. There are two special tags:

- "No markable in this context": This option is provided if the field `this element is sometimes not a markable` is activated in the schema. Select it if the token does not possess any of the features or functions relevant to the annotation in this context (cf. [here](#) [12]).
- "no tag selected": This tag is always an option. It is selected if the markable concerned hasn't been annotated yet. In a new annotation, all markables have this tag.

In order to annotate a markable with a tag, select the button in front of the tag. Then, when you navigate to another element of the text, the markable in the annotation text window will change its color according to the chosen tag (cf. [here](#)). This selection can be made via mouse or keyboard: you can go through the possible tags with the tab key and make the final selection by pressing the space bar when the desired tag is highlighted.



*The range of tags for the German markable "sie" in the example of the personal pronouns.*

## Saving and Loading Annotations

In order to save an annotation, select the menu item "Annotation → Save" or "Annotation → Save as...". Two files must be saved: the corpus and an annotation data file, which essentially comprises the schema and the information on which token has been given which tag; it also contains the specification of the XML format of the corpus. Therefore, when saving, you will be requested to specify two file names. It is advisable to choose two names which can easily be associated with each other when loading, e.g. "Annotation1\_corpus.xml" and "Annotation1\_data.xml". Under "Save as..." file names are required, under "Save" only in the case that you are saving the annotation for the first time.

In order to open a saved annotation, select "Annotation → Open..." in the menu. You will be requested to select two subsequent files, the annotation corpus file and the annotation data file.

---

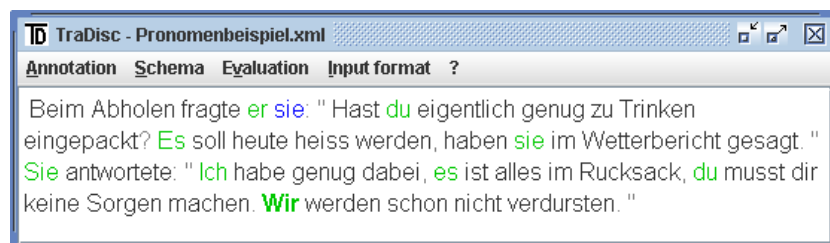
## Splitting and Merging Annotations

**Splitting an annotation.** TraDisc enables you to split the active annotation into two annotations. The corpus is then divided at a place defined by the user. To do this, select the element of the corpus that is to be the first element of the second corpus after separation via double click or the navigation buttons. Then select "Annotation → Split" in the menu. You will now be requested to enter four file names: a name for the annotation corpus file and a name for the annotation data file (cf. [here](#)) of both corpus halves. You then have two saved annotations. Both have the same schema and XML corpus format.

**Merging two annotations.** In order to merge two annotations, they must both have the same XML corpus format. Furthermore, the dimensions of the respective schemata must be the same, i. e. the number of rows and columns and their names must correspond. The annotation whose text is supposed to begin the new, composed text must be loaded in TraDisc. Select the menu item "Annotation → Merge with..."; you will be requested to select the annotation corpus file and the annotation data file of the annotation you would like to add (cf. [here](#)). The text of the newly loaded corpus will appear after the text of the active corpus in the annotation text window. The annotation data, i.e. the tags of the elements of the new part of the corpus, will be integrated. The schemata of both annotations will be merged to one new schema, comprising all the entries of both the active and the added annotation (cf. [here](#)).

## Printing Annotations

By selecting "Annotation → Print" in the menu, you can print the corpus text with the annotations. All annotated markables displayed in green in the annotation text window will be printed in bold. You will find an abbreviated version of the tag a markable has been annotated with in braces behind it. "Abbreviated" means that only the beginning of the row and column names defining the tag will be printed. So if, as in the example of the personal pronouns, you have a tag *Singular-3rd Person*, only *{Singular,3rd}* will be printed behind a token annotated with this tag. It is advisable to consider this when naming the schema dimensions.



Beim Abholen fragte **er** *{Singular,3}* sie: " Hast **du** *{Singular,2}* eigentlich genug zu Trinken eingepackt? **Es** *{Singular,3}* soll heute heiss werden, haben **sie** *{Plural,3}* im Wetterbericht gesagt. " **Sie** *{Singular,3}* antwortete: " **Ich** *{Singular,1}* habe genug dabei, **es** *{Singular,3}* ist alles im Rucksack, **du** *{Singular,2}* musst dir keine Sorgen machen. **Wir** *{Plural,1}* werden schon nicht verdursten. "

*The model annotation and its printout.*

---

# Chapter 4. Evaluating and Analyzing Annotations

TraDisc offers multiple options of evaluating and analyzing annotations. They are summed up under the menu item "Evaluation".

## The Evaluation Table

The evaluation table is opened by selecting the menu item "Evaluation → Show evaluation table". The table shows how often the various tags (i.e., row and column titles) have been used as annotations. In our example, the table shows how many personal pronouns of the 1st person singular, 2nd person singular, 3rd person singular, etc. were annotated in the text. Below the table there are three boxes: normalized, combined with loaded tables and only selected markables. When the evaluation table is opened for the first time, none of the three boxes are checked, and the table displays the distribution of the tags throughout all markables of the corpus text. The three options controlled by the boxes are described in the following paragraphs.

not normalized. corpus contains 44 words.			
	1. Person	2. Person	3. Person
Singular	1.0	2.0	4.0
Plural	1.0	0.0	1.0

evaluation display options

normalized  combined with loaded tables  only selected markables

controls

set normalization factor export manage evaluation files select markables close

*The evaluation table of the completely annotated pronoun example.*

## Normalizing the Evaluation Values

In TraDisc, normalizing means putting the results of the annotation into proportion with a corpus of a desired text length, that length being the number of tokens in the corpus. The desired text length is called the normalization factor. The conversion of the results in the table cells follows the formula:

$$\text{Normalisierter Wert} = \text{Normalisierungsfaktor} \cdot \frac{\text{Absoluter Wert}}{\text{Textlänge}}$$

You can set the normalization factor with the button **set normalization factor**. A normalization factor of 1000 tokens is set by default. If the field **normalized** is checked, the entries of the table will be converted as described.

normalized to 100 words.			
	1. Person	2. Person	3. Person
Singular	2.27	4.55	9.09
Plural	2.27	0.0	2.27

evaluation display options

normalized  combined with loaded tables  only selected markables

controls

set normalization factor export manage evaluation files select markables close

The evaluation table of the pronoun example, normalized for 100 tokens.

## Selecting the Markables to Evaluate

If you are not interested in the distribution of all markables to the tags of the table, but only in the distribution of a few specific ones, click on **select markables**. A dialog box comprising two lists will open: on the left side under the title **All markables** you will find a list of the markables entered into the [schema](#). You can select the markables you would like to have evaluated here. By clicking on **add markable(s)**, they are added to the list on the right side, **selected markables**. You can remove markables from this list by selecting them and clicking on **remove markable(s)**. A click on **close** closes the selection dialog box.



The dialog box for selecting markables.

If the list **selected markables** is not empty, checking the field **only selected markables** below the evaluation table will apply the list to the table, which will then only count the markables on the list.

not normalized. corpus contains 44 words.			
	1. Person	2. Person	3. Person
Singular	0.0	0.0	2.0
Plural	0.0	0.0	1.0

evaluation display options

normalized    combined with loaded tables    only selected markables

controls

set normalization factor   export   manage evaluation files   select markables   close

The evaluation table, displaying only the values of the markables selected above, she and he.

## Exporting Evaluation Data

The evaluation table can be saved as a csv-file (comma separated values) by clicking on the button **export**. This format is supported by common spreadsheet programs such as OpenOffice.org Calc or Microsoft Excel, which makes it possible to carry out further calculations or perform visualization in diagrams with one of these programs.

The table values exported are always those visible on the screen. It is important to pay attention to how the option boxes below the evaluation table are checked, as this determines the displayed values.

## Combining the Evaluation Table with Loaded Evaluation Files

TraDisc permits you to load evaluation data from a file and add the entries from the table of this file to the entries of the current evaluation table. The desired file must be an export file in the csv-format of a TraDisc annotation (cf. the previous paragraph), and the dimensions (row and column amount and names) of the loaded table must correspond with the current table.

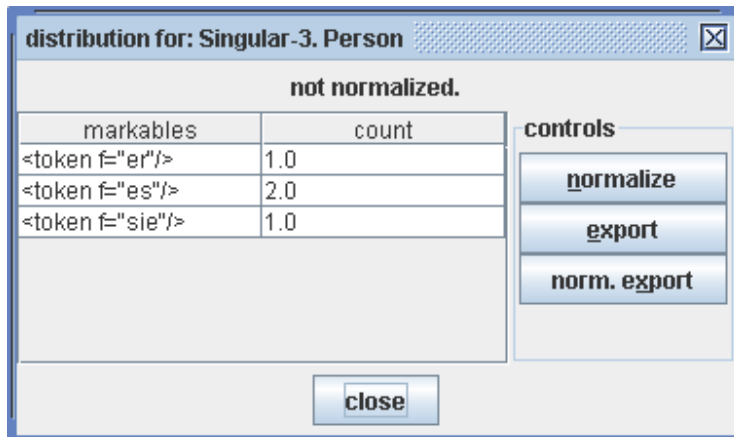
In order to load a file, click on the button **manage evaluation files** below the evaluation table. A small dialog box with the title **current evaluation files** will open, consisting of a list of the presently loaded evaluation files (which initially is empty) and three buttons. By clicking on the button **add evaluation file**, a file selection dialog box opens which lets you open an evaluation file. Multiple files can be loaded simultaneously this way. In order to delete an evaluation file from the list, select it and click on the button **remove evaluation file**. The dialog box is closed by clicking on **close**.

In order to add the values of the loaded evaluation files to the current table, check the box **combined with loaded tables**. The values in the loaded tables will then be added to those of the current table and the sums shown. If the table is also [normalized](#), the values will be assessed according to the number of tokens in the respective corpus. So if two tables are combined, one of which is based on a more extensive corpus, and the combined table is normalized, the result will be more influenced by that table than by the one pertaining to a shorter corpus.

## Evaluating a Tag

In order to find out which markables a tag was used as an annotation for and how often, click on the cell

of the evaluation table corresponding to the tag. A window is then displayed with a list of all markables entered into this cell of the schema. After the markables, it states how often these were annotated with the selected tag. The button **normalize** enables you to normalize the values for a specific corpus length here as well; the normalization factor entered in the evaluation table is used (cf. [here](#)). A further click on **normalize** will again display the counted, not normalized values. This table can be exported into a csv-file as well by clicking on **export**.

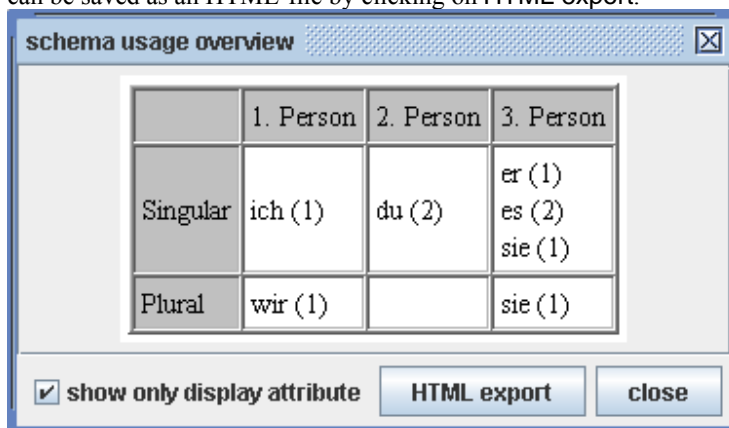


The distribution of the tag "Singular-3rd Person", in the upper right cell of the evaluation table in our example.

## Overview of the Schema Usage

In order to obtain an overall view on which markables of the [schema](#) appear in the text and are annotated, select the menu item "Evaluation → Schema usage". A dialog box will open that is very similar to the [schema overview](#). However, here only the markables are shown which are annotated with one of the tags from the table. Markables entered into the schema but not appearing in the text are not shown in the table. The number of times a markable has been annotated with a specific tag is noted in brackets following the entry of the markable in the cell corresponding to the tag. The values in a field of this table correspond to the values listed in the dialog box for [evaluating a tag](#), which opens when you click on a cell of the evaluation table.

The check box **show only display attribute** as described for the [schema overview](#). The table can be saved as an HTML-file by clicking on **HTML export**.



The schema usage of the personal pronoun example.

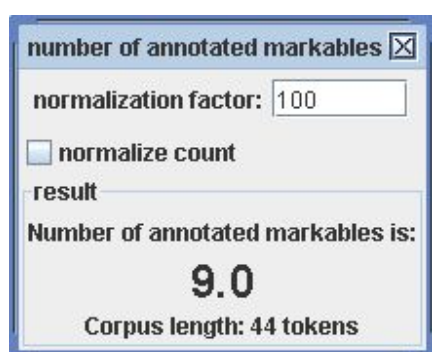
## The Total Number of Annotated Markables

---

If you are not interested in the distribution of annotated markables in the table, but in the total of all annotated markables together, TraDisc gives you two options.

## The Number of All Annotated Markables in the Entire Corpus

If you would like to know how many markables of the text are annotated with any tag from the table (that is the number of markables displayed in green in the [annotation text window](#); red, not yet annotated markables, and blue markables, which have none of the features being annotated in their context, are not counted), select the menu item "Evaluation → Count annotated markables". A small window will appear, showing the desired value. If the number of annotated markables is to be normalized for a specific text length, enter it in the field normalization factor and check the box normalize count.



*The dialog box for counting the total number of annotated markables in our example.*

## Distribution of Annotated Markables over Parts of the Corpus

You can use TraDisc to divide the corpus into parts of equal length and have the number of annotated markables counted separately in each of these parts. By doing this, you obtain a distribution of the annotated markables in the corpus and can identify interesting parts which, for example, contain an amount of annotated markables above average. This function can be found in the menu under "Evaluation → Distribution of annotated markables". The dialog box shows a table containing the respective numbers of annotated markables in the parts of the corpus. Each part of the corpus takes up a row of the table. The number of the first token of a part is mentioned in the column **start token**, the number of the last token is mentioned in the column **end token**. One can picture a window of a certain text length passing over the corpus text. TraDisc counts how many annotated markables appear in the part of the corpus visible through the window for every window position. How large this window should be (i.e., how many tokens it should contain), and by how many tokens it should be repositioned (i.e., how large the "step" from one window position to the next should be), can be determined with the parameters "Window size" and "Step length". By clicking on **change parameters**, a small dialog box will open and let you change the parameter values.

### Note

The last window may be shorter than the set window size, as there may not be enough tokens left at the end of the corpus. You can check this by the numbers of the start token and end token in the last table row.

By clicking on the button **export**, the distribution table can be saved as a csv-file (cf. [here](#)).

start token	end token	annotated m...
1	10	2
11	20	2
21	30	3
31	40	2
41	44	0

*The distribution of annotated markables in the model annotation, with a window size and step length of 10 tokens.*

## Calculating the Complexity Score of an Annotation

TraDisc gives you the option of ascribing a so-called complexity score to an annotation. The complexity score pertains to how often the various tags of the schema were assigned as annotations. Every tag is ascribed a value according to its complexity; this is done via the complexity score table.

### The Complexity Score Table

You can get to the complexity score table through the menu: "Evaluation → Complexity score table". The dimensions are the same as in the [schema table](#), as are also the names of the rows and columns. Each cell of the table contains the complexity score of the tag corresponding to the row and column. In order to change the complexity score of a tag, double click the corresponding cell. You can then enter a desired value.

If the user has not changed them, the complexity scores of the tags are set to default. If you would like to return all values in the table to this setting, click on the button **set default scores**. The default scores of the cells get larger the higher the row and column numbers get. This is because TraDisc was developed to annotate the junctors of corpora. In the junctor schema used, the complexity of the junctors increases the further to the bottom right in the schema table a cell is.

By clicking on **save**, the current complexity score table can be saved as a csv-file. Such files can be loaded again with the button **load**, provided that the dimensions and row and column names of the complexity score table you would like to load correspond with those of the active schema.



	1. Person	2. Person	3. Person
Singular	2.0	3.0	4.0
Plural	4.0	6.0	8.0

The default complexity score table for the personal pronoun example.

## Calculating the Complexity Score for an Entire Annotation

The menu item "Evaluation → Complexity score" opens a dialog box displaying the complexity score of the annotation. This value is calculated as follows: For every tag, there is a count how often it was assigned as annotation to a markable. (This is the same number you will find in the [evaluation table](#) in the cell corresponding to the tag.) This amount is multiplied with the complexity score of the tag from the complexity score table. You then have the complete complexity of the annotation concerning one tag. This value is calculated for every tag, and the sum of all tag complexity scores is drawn. The sum of all tag complexities is the final complexity score of the entire annotation, which is displayed in the dialog box global complexity score. The calculation, to put it more formally, is the following formula:

$$\text{Gesamtkomplexitätswert} = \sum_{\text{Tag}} (\text{Anzahl der Markables mit dem Tag als Annotation}) \cdot (\text{Komplexitätswert des Tags})$$

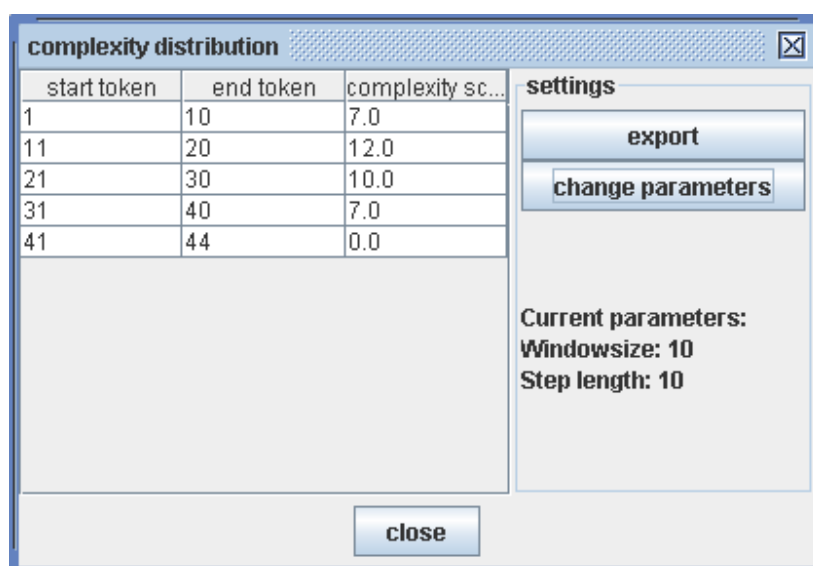
The complexity score can also be normalized for any text length; to do this, enter the desired normalization factor in the field normalization factor and check the box normalize score (cf. [here](#)).

The complexity score of the model annotation, normalized for 100 tokens.

## Distributing the Complexity to Parts of the Corpus

If you would like to know whether the complexity of the annotated markables is the same throughout the corpus or if there are areas in the corpus with a higher or lower complexity, select the function "Evaluation → Complexity Distribution" in the menu. The dialog window will show a table displaying the complexity scores for the various parts of the corpus; for each part the number of the start and end token is listed. For this calculation, only the complexities of the markables within the boundaries are summed.

The parameters "Window size" and "Step length" as well as the buttons in the dialog box correspond to those in the dialog box for the [distribution of annotated markables](#).



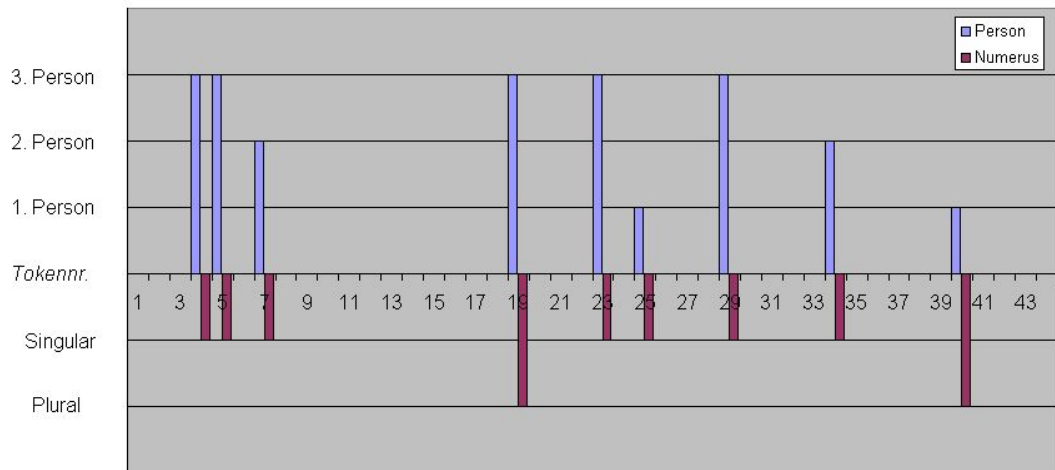
The complexity score distribution of the model annotation, with a window size and step length of 10 tokens.

## Creating Junctograms

A junctogram is a visualization of the annotated functions throughout the course of a text (the name *junctogram* derives from the fact that TraDisc was originally developed to annotate junctors). Every annotated markable is ascribed two numbers; these depend on which tag the markable was annotated with, more precisely which schema cell the tag is represented by. Every column name is ascribed a positive number according to the position of the column. Every row is ascribed its position as a negative number. In the model schema of the personal pronouns, the allocations are therefore: *1st Person: 1, 2nd Person: 2, 3rd Person: 3* for the columns and *Singular: -1, Plural: -2* for the rows. The other, not annotated markables and the tokens that are not markables are ascribed a double zero. Now a diagram can be produced which displays the token number on the x-axis and coordinates the value mentioned above with each token number on the y-axis. Using the visualization options of spreadsheet programs such as OpenOffice.org Calc or Microsoft Excel, a useful illustration of the annotation can be created.

In order to be compatible with a spreadsheet program, this distribution of values is exported into a table in a csv-file. The menu item "Evaluation → Export junctogram" first takes you to a dialog box in which you may specify the markables to be entered into the junctogram. You can either select the button **all markables** (which is set to default), or the button **markables selected below**. Below both buttons, two lists are displayed; the left one contains all markables of the schema. You can select desired markables there and add them to the right list by clicking on **add markable(s)**. The right list is used to produce junctograms in the case that **markables selected below** is activated. Markables not contained in the list are then ascribed a double zero, if they are annotated or not.

A click on **export** will save the data as a csv-file. This table has four columns: the number of each token of the corpus in the column *token number*, the token in the column *token*, and the two values ascribed to the token as described above in the columns *vertical dimension value* for the value of the schema column and *horizontal dimension value* for the value of the schema row. The allocation of the row and column names of the schema to the values is displayed in the csv-file above the actual junctogram table. This table can now be used to create a diagram.



*A junctogram created from the model annotation of the personal pronouns.*

---

## Part II. Tokenizer

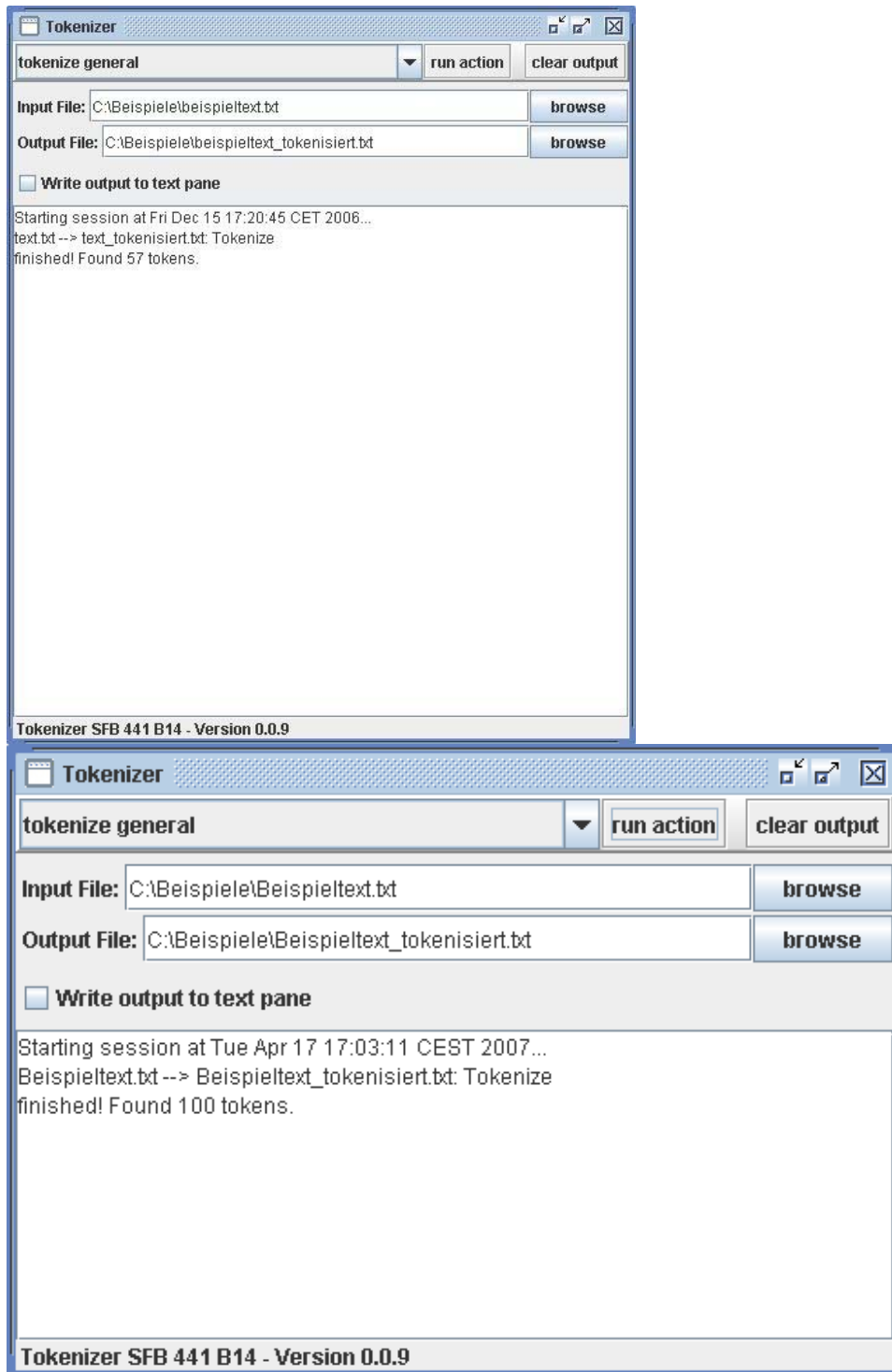
Tokenizer makes it possible to put a corpus in an XML format TraDisc can read, the [TraDisc standard format](#). This enables the user to annotate any corpus available in a simple text format.

---

---

# Chapter 5. Using Tokenizer

In order to be able to use a text as input for Tokenizer, it must be available in a *plain text* file. That is the standard format for text files with the frequent ending *.txt*.



*Tokenizer after tokenizing the text in our example.*

---

## Tokenizing the Text

Tokenizing a corpus means identifying the single tokens of the corpus text. In the simplest case, those are the single words (and punctuation marks). Tokenizer does this automatically, writing a temporary file in which every token has its own line. In order to create this file, enter the name and path of the file the corpus is saved in in the field **Input File** in Tokenizer. The button **Browse** will open a file selecting dialog box where you can select this file. In the text field **Output File** directly beneath, you must enter the name of the temporary file; you can open the file selecting dialog box here with **Browse** as well.

You can choose which action Tokenizer is to perform in the selection box in the upper left corner. To tokenize a text, select **tokenize general**, which is set by default. When the input and output files have been determined, a click on **run action** will start the desired action. The large text field in the bottom half of Tokenizer will provide information whether the action was successfully carried out or not and how many tokens were found.

The text of the corpus in the model annotation goes like this:

*Beim Abholen fragte er sie: "Hast du eigentlich genug zu Trinken eingepackt? Es soll heute heiss werden, haben sie im Wetterbericht gesagt." Sie antwortete: "Ich habe genug dabei, es ist alles im Rucksack, du musst dir keine Sorgen machen. Wir werden schon nicht verdursten."*

If a file with this text in the above form is entered as the input file, and the text is tokenized as described above, the contents of the temporary file will look like this:

Beim  
Abholen  
fragte  
er  
sie  
:  
"  
Hast  
du  
eigentlich  
genug  
zu  
Trinken  
eingepackt  
?  
Es  
soll  
heute  
heiss  
werden  
,  
haben  
sie  
im  
Wetterbericht  
gesagt  
.  
"  
Sie  
antwortete  
:  
"

---

```
Ich
habe
genug
dabei
,
es
ist
alles
im
Rucksack
,
du
musst
dir
keine
Sorgen
machen
.
Wir
werden
schon
nicht
verdursten
;
"
```

If you check the box **Write output to text pane** before beginning to tokenize, the tokens will also be produced in separate lines in the large text field of the Tokenizer window, in the same way as they are written in the temporary output file. In order to erase the text from the output field, use the button **clear output**.

### Note

The selection box of Tokenizer contains a few special options for tokenizing texts in Old Spanish and Sursilvan, as well as for eliminating certain punctuation marks. These functions were developed in regard to the initial use of TraDisc for annotating junctors in Romance texts. They are not explained further in this manual.

## Creating the XML File

In order to change the temporary file with one token per line into an XML file with the text in the [TraDisc standard XML format](#), this temporary file must be selected in the field **Input file** (which means that the output file from tokenizing must now be determined as the input file). For the output file, select the path and file name the XML corpus file is to have, usually ending with *.xml*. Then choose **create xml output** in the action selection box and click on **run action**. Tokenizer will now create an XML file, in which every line of the input file corresponds with an XML element. Tokenizer automatically recognizes if a line contains a word or a punctuation mark and accordingly selects the XML tag names of the elements as **token** or **other** (cf. [here](#)). The XML file can now be used as a corpus file in TraDisc.

If the temporary token file of the example above was converted into an XML file as described, the contents of the XML file would look like this:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<corpus>
  <token f="Beim"/>
```

---

```
<token f="Abholen"/>
<token f="fragte"/>
<token f="er"/>
<token f="sie"/>
<other f="."/>
<other f="&quot;"/>
<token f="Hast"/>
<token f="du"/>
<token f="eigentlich"/>
<token f="genug"/>
<token f="zu"/>
<token f="Trinken"/>
<token f="eingepackt"/>
<other f="?"/>
<token f="Es"/>
<token f="soll"/>
<token f="heute"/>
<token f="heiss"/>
<token f="werden"/>
<other f=","/>
<token f="haben"/>
<token f="sie"/>
<token f="im"/>
<token f="Wetterbericht"/>
<token f="gesagt"/>
<other f="."/>
<other f="&quot;"/>
<token f="Sie"/>
<token f="antwortete"/>
<other f="."/>
<other f="&quot;"/>
<token f="Ich"/>
<token f="habe"/>
<token f="genug"/>
<token f="dabei"/>
<other f=","/>
<token f="es"/>
<token f="ist"/>
<token f="alles"/>
<token f="im"/>
<token f="Rucksack"/>
<other f=","/>
<token f="du"/>
<token f="musst"/>
<token f="dir"/>
<token f="keine"/>
<token f="Sorgen"/>
<token f="machen"/>
<other f="."/>
<token f="Wir"/>
<token f="werden"/>
<token f="schon"/>
<token f="nicht"/>
<token f="verdursten"/>
<other f="."/>
<other f="&quot;"/>
</corpus>
```



---