

# ZStringTool

Hayley Iben  
Connectix Corp.  
Draft 1.0  
Last Modified June 15, 2001

## Overview

The ZStringTool class is generally not used within the application source code itself. It provides the core implementation of tools to be used with ZStrings. The three most important tools are extraction, comparison and override dictionary generation. Several options can be set in the ZToolOptions class to cause different behavior of the ZStringTool and ZStringParser.

## Options

There are several options that can be set before calling the operations of the ZStringTool. These options are contained in the ZToolOptions class. These options include:

### 1. mOutputNumericTags

This option indicates the format of the html tags written to the output. If it is true, the html tag is in numeric form. For example, the tag representing an ampersand would be generated as “&#038;”. If it is false, the html tag is in alphabetic form. For example, the ampersand tag would be generated as “&amp”. The default value for this option is true.

### 2. mCategorizeOutput

This option indicates the sorting used for the output file. If it is true, the output is grouped according to category. The order for the sort is:

- Strings added
- Pairs of strings modified new and old
- Strings deleted
- Strings duplicated
- Strings formed incorrectly
- Strings containing line breaks
- String containing a possible malformed html tag
- Strings exceeding the specified limit
- Strings with high ASCII characters changed
- String still containing high ASCII characters.

Strings within the same category are sorted alphabetically by ZString name. The strings that contain no errors are printed after the last category. If it is false, the output is sorted alphabetically by the ZString name. The default value for this option is true.

### 3. mAllowTagSemicolon

This option indicates how semicolons are handled at the end of alphabetic tags. If it is true, semicolons at the end of alphabetic tags are treated as part of the tag and removed from the output stream. For example, `&#038;` is converted to `&#038;` or `&#038;` (based on the `mOutputNumericTags` option). If it is false, semicolons are left in the stream for alphabetic tags. For example, `&#038;` is converted to `&#038;` or `&#038;` (based on `mOutputNumericTags`). The default value for this option is false.

### 4. mFlagDuplicates

This option indicates how duplicate ZStrings are treated in the output file. If it is true, duplicate ZString tags are flagged in the output. If it is false, they are not. The default value is false.

### 5. mHasOTags

This option indicates which ZString name tag the parser should look for, “Z name=” or “O name=". If it is true, the parser looks for “O name=". If it is false, the parser looks for “Z name=". The default value is false. (This option is only set to true when the input file is a dictionary file, enabling them to be compared.)

### 6. mOutputWarnings

This option indicates when warnings are printed to the output file. If it is true, warnings are printed. If it is false, they are not. The default value is true.

### 7. mPrintErrorsOnly

This option indicates how much information is to be printed to the output file. If it is true, then only errors and warnings (depending on the `mOutputWarnings` option) are printed to the output file. If it is false, everything is printed. The default value is false. (This option is used for reporting only errors and possibly warnings during the creation of the override dictionary.)

### 8. mConvertHighASCIIChar

This option determines the action for high ASCII characters. If it is true, high ASCII characters are automatically converted to their html tags and a warning is produced. If it is false, they are not converted. The default value is true.

## **Extraction Tool**

The ZString extraction tool scans a specified file for any embedded named strings and prints out a list for the localization team. The input file can be any file containing ZStrings, such as an application binary or html file.

## **Comparison Tool**

The ZString comparison tool scans two files to determine what strings have changed since the application was last localized. Each file can be any file containing ZStrings, such as an application binary or html file. Additionally, two override dictionary files can be compared.

## **Override Tool**

The ZString override dictionary generation tool takes a file similar to the output of the extraction tool and creates an override dictionary that can be loaded at runtime.