# README

## Preambule

This GitHub repository contains the sources of a library aiming to validate any UCD (Unified Content Descriptor). This current version aims to respect as much as possible the definition provided by the IVOA standard: An IVOA Standard for Unified Content Descriptors - Version 1.1. The parser is by default configured with the list of all validated UCD words listed in The UCD1+ controlled vocabulary 1.3.

## Functionalities

- Check whether each UCD word is:
  - syntactically valid
  - recognised (i.e. the word is among a list of well known UCD words)
  - recommended by the IVOA (The UCD1+ controlled vocabulary 1.3)

- Possibility to customise the list of known UCD words *(by default all validated UCD1+ are automatically loaded)*
- Validate a full UCD with
  - a list of human readable errors
  - an automatic correction suggestion (particularly for typo)
  - a list of advice to improve the readability of the UCD
- Detection of deprecated UCD words *(when detected a clear error message is returned and a correction suggestion is proposed)*
- Different ways to search UCD words
  - exact match
  - starting with
  - closest match *(take into account possible typo)*
- Support namespace prefix

### Java version

This library is developed using **Java 1.7** *(should be compatible with Java 1.7 or newer)*.

### Download

The compiled JAR, the runnable JAR, the sources and the Javadoc API are available on GitHub for all releases and especially for the latest one.

### Documentation

- Javadoc
- UML
- Documentation/Wiki

### License



## Collaboration

I strongly encourage you **to declare any issue you encounter** here. Thus anybody who has the same problem can see whether his/her problem is already known. If the problem is known the progress and/or comments about its resolution will be published.

In addition, if you have forked this repository and made some corrections on your side which are likely to interest any other user of the libraries, please, **send a pull request** here. If these modifications are in adequation with the IVOA definition and are not too specific to your usecase, they will be integrated (maybe after some modifications) on this repository and thus made available to everybody.

# Repository content

## Dependencies

*No dependency.*

## Resources

The `resources` directory contains two files for the moment:

- `ucd1p-words.txt` . It lists all official IVOA UCD1+ words as provided at http://cdsweb.u-strasbg.fr/UCD/ucd1p-words.txt.
- `ucd1p-deprecated.txt` . This is a list of all *deprecated* UCD1+ words as provided at http://cdsweb.u-strasbg.fr/UCD/ucd1p-deprecated.txt.

These files are loaded by the default parser initialised in the class UCDParser.

If the file `ucd1p-words.txt` is renamed or removed, the default parser will raise a warning on the standard error output and will be initialized with an empty list of known UCD1+ words. Consequently any UCD parsed using this parser will be systematically flagged as *not recognised* and so *not recommended*.

## JUnit

The sources of these three libraries come with some JUnit test files. You can find them in the `test` directory.

In the `test-lib` directory, you will find all JAR files needed to compile and run these JUnit tests. You can use the task `test` of the ANT script as explained below.

## ANT scripts

At the root of the repository, there is an ANT script. It is able to generate JAR for sources, binaries and Javadoc.

This ANT script have the following main targets:

- `build` : Compile all classes of this project.
- `test` *DEFAULT*: Compile all classes and run all the JUnit tests.
- `javadoc` : Generate the Javadoc.
- `publish` : Compile all classes and run all the JUnit tests. If these latter are passed, the library JAR (also runnable) is generated, in addition of a JAR containing all the sources and of another with the complete Javadoc.

## Releases 3

**v1.1** Latest
on 14 Jun 2018

+ 2 releases

## Packages

No packages published

## Languages

**Java** 100.0%