**C-CDA Comparison Tool Documentation:**

Table of Contents

[1.0 Introduction 2](#_Toc16091792)

[2.0 Getting Started 2](#_Toc16091793)

[2.1 System Requirements 2](#_Toc16091794)

[2.2 Launching the Tool 2](#_Toc16091795)

[2.3 User Interface Components 2](#_Toc16091796)

[3.0 Forms of Use 3](#_Toc16091797)

[3.1 Loading Files 3](#_Toc16091798)

[3.2 Section Comparisons 3](#_Toc16091799)

[3.3 Analyzing Results 3](#_Toc16091800)

[4.0 Technical Details 4](#_Toc16091801)

[4.1 Backend Components 4](#_Toc16091802)

[4.1.1 SAX Parser 4](#_Toc16091803)

[4.1.2 Spring Framework 5](#_Toc16091804)

[4.1.3 Comparison Algorithm 5](#_Toc16091805)

[4.2 User Interface Components 5](#_Toc16091806)

[4.2.1 Vue.js 5](#_Toc16091807)

[4.2.2 Bootstrap 6](#_Toc16091808)

# Introduction

This document will serve as the main source of documentation for the C-CDA document comparison tool. This tool will serve to aid the process of C-CDA document comparison in the UAT process. By flagging mismatches between documents irrespective of order, the document defects can be quickly and easily identified by a UAT tester.

# Getting Started

## System Requirements

This tool requires the following technologies/configurations:

* Java
* Google Chrome
* *MacOS users only:* device must be configured to run windows batch (.bat) files.

## Launching the Tool

You will be distributed a file titled “C-CDA\_Comparison\_Tool\_Release.piz”. Despite the “.piz” extension, this is actually a “.zip” file. Unzip this file into a location of your choice. After you have unzipped this file, you will see the following files:

* **webapp-1.0.jar** – The executable jar file that contains the byte code to run the comparison tool
* **ComparisonTool.bat** – A windows batch file to execute that “webapp-1.0.jar” and open Chrome to the appropriate URL of <http://localhost:8080>

To execute the jar file and open the comparison tool User Interface, double click the “ComparisonTool.bat” file.

## User Interface Components

The Comparison Tool user interface has the following components:

* **Nav Bar:** Moving left to right, the Nav Bar contains the following:
  + Toggle between Instructions and Compare tabs
  + Source and Target Document Load
  + Compare Button (displayed only if user is on the Compare tab)
* **Instructions:** The instructions tab is accessible by clicking “Instructions” on the left side of the Nav Bar. This tab will provide instructions for use of the Comparison Tool.
* **Compare:** The compare tab is accessible by clicking “Compare” on the right side of the Nav Bar. This tab contains two collapsible sections:
  + **Document Displays:** This section contains the expandable and collapsible representations of the source and target XML documents.
  + **Compare Results:** This section will display the results of a comparison after the green “Compare” button is clicked on the upper right of the Nav Bar. Additionally, on the right side of this section are various filters for the list of results.

# Forms of Use

## Loading Files

On the top of the screen you will see two file load components, both with the placeholder text “Select Document…”. Load your source document by clicking the “Browse” button on the left document selection component and selecting the appropriate file from your documents. Do the same for the target file using the “Browse” button on the right document selection component.

## Section Comparisons

To compare one section in the source document to another section to the target document, we need to start by selecting these sections. By default, the “section” that is selected is the entire document. To change the section selection, expand the document to the desired location in the document and click the “-“ to the left of the section. After clicking, the “-“ should change to a green check mark. This check mark indicated the section that has been selected. Additionally, the section and its children will have a gray background indicating that this will be the content compared.

After you have selected both sections, compare these sections by clicking the green compare button in the upper right of the nav bar. Upon completion of comparison, the comparison results will be displayed in the “Compare Results” section.

Please note, that while the entire document can be compared all at one time, it is recommended that you compare subsections one at a time. This will not only help with performance, but also help to keep the number of comparison results manageable.

Additionally, as you move through the documents section by section, you can double click on a specific section. By doing so, the text turn green and bolded, marking it as a section that has been reviewed.

## Analyzing Results

After a comparison is complete, mismatches between the documents will be displayed in the “Compare Results” section of the page.

There are several ways that you can analyze each mismatch. First, if you click on the mismatch, it will expand the source and target documents to the mismatch’s location. Additionally, the exact location of the mismatch will be shown in red on the source and target documents. After you have analyzed this result and the surrounding XML elements, you can classify the result as a defect or discard is as an acceptable different.

* **Defect:** If this result should be classified as a defect, then click the “!” button on the right side of the result list item. This will give the result list item a red background.
* **Discard:** If the result is deemed to be an acceptable difference and is not a defect between the two documents, click the “X” button on the right side of the result list item. This will give the result list item a gray background.

To the left of this list are several sets of filters.

* **Search Filter:** A search capability to filter results based on the displayed text for each mismatch.
* **Value, Attribute, and Section Mismatch Filter:** Results are classified into Value, Attribute, and Section Mismatches:
  + **Value:** The value inside of a specific XML tag is mismatched. For example, the following would be a value mismatch:

<para>Value1</para> vs <para>Value2</para>

* **Attribute:** The attribute on a specific XML tag is mismatched. For example, the following would be an attribute mismatch:

<para att=“Value1”/> vs <para att=“Value2”/>

* + **Section:** An XML element in the source document that does not exist in the target document. For example the following would be a section mismatch:

<para>Value1</para> vs <para>Value1</para>

<sec>Value2</para>

The Value Mismatches, Attribute Mismatches, and Section Mismatches can all be filtered based in the result list using the checkboxes on the left.

* **Defects, Discarded, and Unclassified Mismatch Filter:**
  + **Defects:** Mismatches that have been flagged as a defect by the user. These are indicated with a red background.
  + **Discarded:** Mismatches that have been deemed an acceptable difference by the user. These are indicated with a gray background.
  + **Unclassified:** Mismatches that have not yet been classified as defects or discarded continue to have the original white background.

The Defect Mismatches, Discarded Mismatches, and Unclassified Mismatches can all be filtered based in the result list using the checkboxes on the left.

# Technical Details

## Backend Components

### SAX Parser

SAX (Simple API for XML) is an event-based parser for XML documents. A SAX parser reads an XML document from top to bottom, recognizing the tokens that make up a well-formed XML document. Tokens (elements, attributes, and associated values) are read in the order that they appear in the XML document.

The C-CDA Comparison Tool makes use of a SAX parser to read the source and target XML documents into custom “node” class objects. This node class has element name, value, a list of children nodes, a list of attribute nodes, and a unique id. This provides the XML data in a format that is conducive to the implementation of comparison algorithms.

### Spring Framework

Spring provides an application framework and inversion of control container for the Java platform.

The C-CDA Comparison Tool uses the Spring Framework to provide the basic functionality for the Web Application.

### Comparison Algorithm

After being read in, the XML document is represented as a tree using Java objects of the “node” class. The C-CDA Comparison Tool makes use of a recursive level-by-level breadth-first tree traversal algorithm. The source and target trees are traversed simultaneously.

At each level, for each source element, a list of matching element types in the target tree is returned. This list will only contain target elements that have not already been “matched” to another element in the source tree and will be in order that the elements appear in the target document.

This list of target tree elements is iterated over one-by-one to find a “match” to the source element. A match is defined as an element that contains all attributes and matching attribute values that exist in the source element AND a matching element value if present.

Through this process, the comparison tool can identify differences in element order in the source and target documents **at a specific level**. Please note that when looking for a matching target element, this algorithm will NOT look at the children of the source and target elements. It will ONLY look at the attributes, attribute values, and element values at that level.

If a mismatch between the documents is not found, it is grouped into three categories – attribute mismatch, value mismatch, and section mismatch as described in Section 3.3.

## User Interface Components

### Vue.js

Vue.js is an open-source JavaScript framework for building user interfaces and single-page applications.

The C-CDA Comparison Tool uses the Vue.js framework for its User Interface.

### Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web developments. It contains CSS and JavaScript design templates for typography, forms, buttons, navigation and other interface components.

The C-CDA Comparison Tool uses the Bootstrap framework for many of the components in the User Interface.