Group 3 Team 5

DB Edit Software

Test plan

Version 3.0

21st April 2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 1.0 |
| Current Release: | 3.0 |
| Indicator of Last Page in Document: | $ |
| Date of Last Review: | Tue Apr 21 |
| Date of Next Review: |  |
| Target Date for Next Update: | Fri Apr 24 |

Distribution List

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:

Dr. Steve Roach

Customer:

Software Team Members:

Nouri, Ali

Guajardo, Patricia S.

Sanchez, Briana

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | Fri Apr 10 | Ali Nouri | The initial version of test plan |
| 2.0 | Sat Apr18 | Ali Nouri | Revise the document based on the feedbacks from reviewers |
| 3.0 | Tue Apr 21 | Ali Nouri | Apply all the revision provided by Briana |

Note: The template presented in this document was taken from:

Donaldson, S., and S. Siegel, *Successful Software Development*. Upper Saddle River, NJ: Prentice Hall, 2001, pp. 321-323.

Note: The template presented in this document was taken from: Donaldson, S., and S. Siegel, *Successful Software Development*. Upper Saddle River, NJ: Prentice Hall, 2001, pp. 321-323 and modified by Humberto Mendoza and Steve Roach.

Supplementary information is from:

Pfleeger, S. *Software Engineering, Theory and Practice*. Upper Saddle River, NJ: Prentice Hall, 1998, p. 365.

Table of Contents

[Document Control ii](#_Toc38213605)

[Approval ii](#_Toc38213606)

[Document Change Control ii](#_Toc38213607)

[Distribution List ii](#_Toc38213608)

[Change Summary ii](#_Toc38213609)

[1. Introduction 1](#_Toc38213610)

[1.1. Purpose 1](#_Toc38213611)

[1.2. Scope 1](#_Toc38213612)

[1.3. System Overview 1](#_Toc38213613)

[1.4. Suspension and Exit Criteria 1](#_Toc38213614)

[1.5. Document Overview 1](#_Toc38213615)

[1.6. References 2](#_Toc38213616)

[2. Test Items and Features 3](#_Toc38213617)

[2.1. Undo 3](#_Toc38213618)

[2.2. Copy Row 3](#_Toc38213619)

[2.3. Paste Row 3](#_Toc38213620)

[2.4. Insert Row 3](#_Toc38213621)

[2.5. Delete Row 3](#_Toc38213622)

[2.6. Search 3](#_Toc38213623)

[2.6.1. Search and Replace Elements 3](#_Toc38213624)

[2.6.2. Find Match Cases 4](#_Toc38213625)

[2.6.3. Find Match Whole Words 4](#_Toc38213626)

[3. Testing Approach 5](#_Toc38213627)

[4. Test Edit Menu Functions 6](#_Toc38213628)

[4.1. Requirements 6](#_Toc38213629)

[4.2. Test Undo Functionality 6](#_Toc38213630)

[4.3. Test Copy and Paste A Row Functions 7](#_Toc38213631)

[4.4. Test Insert A Row Functionality 8](#_Toc38213632)

[4.5. Test Delete A Row Functionality 9](#_Toc38213633)

[4.6. Test Search Function 10](#_Toc38213634)

[4.7. Test Replace with A New Value Function 11](#_Toc38213635)

[5. Test Schedule 14](#_Toc38213636)

[6. Appendix 15](#_Toc38213637)

[6.1. Screenshots of The Testing 15](#_Toc38213638)

# Introduction

In this test plan, we try to provide a comprehensive test suite for a database editor. Database editor is a software that stores data in XML files. There is no relation between the columns. This editor allows users to access the database, read, add, and modify the data. The goal of this test plan is to cover all possible bugs or glitches in the software. Here, we focus on edit the menu functionality of the software. We do not have any access to the code of the software. Thus, our approach is black-box testing. Each test cases explain with all the necessary details. The test cases are covered undo, copy row, paste row, insert row, delete row, and search functions in the editor. This test plan started on April 8th and will be finished on April 21st. The test plan improved by the review of other team members and the writer used their feedback to provide a comprehensive test plan. GitHub was the platform to communicate between the team members.

## Purpose

This test plan contains all the requirements needed to test the software, a test suite that covers all editing menu functionality, and documentation of all the steps to apply each test case. The purpose of this project is to provide a comprehensive test plan document. This document covers all the steps that is needed to test each test case, elaborate on the purpose of each test case, and analyze the performance of the system after each test case.

## Scope

In this test plan, we focus on edit menu operations. The functionalities in the edit menu help the user to interact better with the interface of the software. These operations are undoing the changes to the previous action on the table, copy and paste a row in the table, insert or delete a row from the table, and search and replace a value in the table. Other functionalities of the software are tested in other test plans by other teams.

## System Overview

The system is a database editor that uses XML to store the data. User can add, remove, change, or search the data easier by using this system. There are some functions that help the user to interact better with the system. These basic functions are provided by the software in the edit menu. The system doesn’t allow the user to modify the schema of the data base. the database in the system is a set of flat files organized as rows and columns. A collection of related files is contained in a database.

## Suspension and Exit Criteria

The exit criteria for this test plan is all the high critical tests must be passed and more than 60% of the medium critical test must be passed. Otherwise, we consider the system as a failed project and the test must be repeated after debugging the problems.

## Document Overview

In section 1, the system introduced, the scope of the project explained, the suspension and exit criteria specified, and an overview of the system provided.

In section 2, the test items and features explained.

Section 3 is about the test approach and listed the test cases, the critical status of each test case, and demonstrated each test case objective.

In section 4, each test case explained, the operation actions and the purpose of them provided. The expected results and the actual result can be found for each test case in this chapter.

Finally, in section 5 the test schedule is listed.

The appendix is available in the section 6 of this document.

## References

[1] Spec.docx

# Test Items and Features

In this test, our focus will be on editing menu operations. All the possible functions and operations must be tested. There are some common user actions that need to be tested. The functions that are provided on the menu are inserting, deleting, copying, pasting, Search, and undo. We will go through each of them and provide appropriate tests for each of these functions.

## Undo

This function reverses the last command or deletes the last entry the user typed. This function can be active by pointer or just press its shortcut (^z). This function works by storing the last changes made. By calling this function, the last changes will be overwritten on what we have.

## Copy Row

This function copies the selection from the table and stores it in the clipboard of the O.S. It can be paste in another cell or can be paste in space in another software or document. The shortcut to copy a row is (^c). The clipboard of the O.S. just can store one content. That means by copy two different objects at a different time, the second one will overwrite on the first one.

## Paste Row

This function pastes the content in the clipboard of the O.S. into the selected cell. This function can apply for multiple times. So, the users can paste the content into multiple cells without losing the content. This function can be called by using the pointer or using the (^p) shortcut.

## Insert Row

This function helps the users to add a new entity to the dataset. Technically it creates a new row in the table. This function stores the new entity by adding a new node to the XML file too. This function can be active by using a pointer or press the shortcut (^i).

## Delete Row

It deletes selected row from the table. This function removes an entity from the database. It can be called by pointer or pressing the shortcut (^d).

## Search

This function helps users to find a specific string they are looking for. This function doesn’t have any shortcut and must be called by the pointer. The GUI could provide a shortcut for this action, but it didn’t.

### Search and Replace Elements

Users can use the find function or they can use find and replace action. It helps the users to find specific data and replace the cell by another value. The ReplaceAll function is already implemented in the GUI. Thus, the users don’t need to replace the previous value by a new value one by one.

### Find Match Cases

Find Match Case option helps users to find the exact match they are looking for. This option helps them to find the result they want more accurate. It shows the cells that have the word inside them.

### Find Match Whole Words

This option helps the user to find the exact word they want on the table. This option just returns the cells that have the exact word the users searched.

# Testing Approach

In this test, we try to test the functionalities of the editing menu in the database editor software in any possible scenarios that cause an error. We will use black-box testing as our approach. We do not have any access to the software codes, and this is the reason we chose black box testing. We will try different scenarios, inputs, and check the boundaries to be sure the edit menu works and all the functions inside the edit menu are working without any problem. We will categories the test cases by low, medium, and high critical condition. These labels help debuggers to address the problems based on the priority of fixing the issues.

Table 1: Test Plan

|  |  |  |
| --- | --- | --- |
| **TEST SUITE <Identifier>** | | |
| **Description of Test Suite** | **In this test suite, the focus is to test edit menu operations on the software. This test suite is included testing for inserting, deleting, copying, pasting, Search, and undo** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| UFT | **Test the functionality of UNDO** | **High** |
| CRT | **Test the functionality of COPY ROW** | **Medium** |
| PRT | **Test the functionality of PASTE ROW** | **Medium** |
| IRT | **Test the functionality of INSERT ROW** | **High** |
| DRT | **Test the functionality of DELETE ROW** | **High** |
| SFT | **Test the functionality of SEARCH** | **High** |
| RFT | **Test the functionality of REPLACE** | **Medium** |

We provide 10 different test cases in our proposed Test Suite. We labeled them based on the critical status they have. We labeled high critical those functions which are the main editing functions of a table. The medium labels belong to those that help users to work faster or improving the user-friendly interface functions.

# Test Edit Menu Functions

In this section, we provide the requirements of each test case, the steps to apply those test cases, the inputs for each test case, and the expected output. We try to cover all the possible scenarios that cause any bugs or glitch in the software. In this section, each test case documented well, and we provided details and information to repeat the test later if it is necessary or analyze the results of any cases.

## Requirements

Before starting the test, please be sure that you have these following files in a same directory.

DbEdit.jar

TEST\_DB.xml

TEST\_DB\_\_BIG\_TABLE.XML

TEST\_DB\_\_CONSTRAINT\_TABLE.XML

TEST\_DB\_\_TYPE\_TABLE.XML

DB/ TEST\_DB\_\_TYPE\_TABLE.XML

You need to be sure that Java Compiler already is installed on your machine.

## Test Undo Functionality

**Objective:** This test checks the functionality of UNDO in the edit menu.

**Notes:** In this test, Undo button will be checked, the functionality of it, and the shortcut to reach it. This test will apply this function repeatedly and after some other actions and before any actions too.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: UFT | | | | Current Status: Passed | | |
| Test title: Test functionality of Undo in edit menu | | | | | | |
| Testing approach: We didn’t have access to the source code of the software so, for this testing, we choose black box testing approach. | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Click Open from File menu Choose TEST\_DB.xml from current directory  Click CONSTRAINT\_TABLE  -Select a cell and use shortcut ^d, or from the menu, edit tab, choose delete option to remove the cell. Repeat this task for five times until you get the UNDO BUFFER IS EMPTY warning.  -Check to be sure all deleted values are returned, and their position and value are the same as you removed. | PURPOSE  Navigate program to reach a data base.  Operate some actions and make some changes in the table.  Try to undo the actions and check the consistency of the new and old values. | | | EXEPCTED RESULTS  TTC XML Editor pops up.  Remove some values from the selected cells.  The expected result in this test is to have same table after using the Undo operation. | COMMENTS  Pictures [1..3] in the Appendix shows the steps. |
| Concluding Remarks:  The Undo action works correct without any problem. It uses a buffer to store all the actions. It returns changes properly. There is just one problem, it returns the deleted rows at the end of the table, not at the location they were. It can cause some problem, specifically in the sorted tables. Our suggestion is to keep the location of the entity as same as it stores the actions. | | | | | | |
| Testing Team:  Ali Nouri | | | Date Completed:  11/04 | | | |

## Test Copy and Paste A Row Functions

**Objective:** This test checks the functionality of Copy Row and Past in the edit menu.

**Notes:** In this test case we want to check the functionality of Copy a row and Paste a Row. Our approach is to copy a row, switch the app and in another app, copy some text then, going back and paste in the table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: CRT, PRT | | | | Current Status: Failed | | |
| Test title: Test functionality of Copy Row and Paste Row from edit menu | | | | | | |
| Testing approach: We didn’t have access to the source code of the software so, for this testing, we choose black box testing approach. | | | | | | |
| STEP  1  2  3  4  5 | OPERATOR ACTION  -Run the program  Click Open from File menu Choose *Test\_DB.xml* from current directory  Click *CONSTRAINT\_TABLE*  -In the new windows, from Edit menu choose Copy Row.  -Copy a cell by using the shortcut (^c) and paste it into another cell  -Open Notepad and type a few words. Select those words and copy them.  -From the Edit menu, choose Paste and be sure you are pasting the same content you copied from the table in the previous action. | PURPOSE  Navigate program to reach a data base.  Copy a content from the table  Testing the Copy function  Store other value from other software to overwrite the O.S. clipboard.  Paste a content from another software in the table | | | EXEPCTED RESULTS  TTC XML Editor pops up.  The copy operation must work even by using the shortcut (^c).  The software shouldn't use the O.S. clipboard to store a value.  It must return the same value that we copy from the table | COMMENTS  Picture [5..7] shows the steps of this test. |
| Concluding Remarks:  The Copy Row and Paste Row function works well until the user tries to switch to other software, copy a new content and return to the software. In this situation, the content will be overwritten, and the original content will be lost. The test failed in this test case. It needs to re-write the code to fix this issue. | | | | | | |
| Testing Team:  Ali Nouri | | | Date Completed:  11/04 | | | |

## Test Insert A Row Functionality

**Objective:** This test checks the functionality of Insert Row in the edit menu.

**Notes:** In this test, we want to check the functionality of Insert Row from the edit menu. This option helps the user to add a new row of data in the table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: IRT | | | | Current Status: Passed | | |
| Test title: Test functionality of Insert Row from edit menu | | | | | | |
| Testing approach: We didn’t have access to the source code of the software so, for this testing, we choose the black box testing approach. | | | | | | |
| STEP  1  2  3  4  5 | OPERATOR ACTION  -Run the program  Click Open from File menu Choose *Test\_DB.xml* from current directory  Click *CONSTRAINT\_TABLE*  -In the new windows, from Edit menu choose Insert Row. Add some data and save the table.  - Remove all the data from the table and try to insert a new row into an empty table.  -Use the shortcut to add a new row, add some content there, then store the table.  -Try to load the table and check if the rows are stored | PURPOSE  Navigate program to reach a data base.  Try to insert a new row into the table  Test the operation in an empty table  Test operation by using its shortcut.  Check the rows after store the data | | | EXEPCTED RESULTS  TTC XML Editor pops up.  A new row is added into the table.  Having an empty table.  Same functionality as if you choose the operation from the menu.  Both new rows are existed in the table. | COMMENTS  Pictures [8..9] elaborate this action |
| Concluding Remarks:  The Insert Row functionality in the edit menu works well without any problem. There is just one concern. It adds a new row all the time at the end of the table. In some database it needed to have a sorted table to improve the functionality and speed of the indexing. | | | | | | |
| Testing Team:  Ali Nouri | | | Date Completed:  11/04 | | | |

## Test Delete A Row Functionality

**Objective:** In this test case, we try to check the functionality of the Delete Row option in the edit menu.

**Notes:** This action is one of the essential actions to manipulate the data in the table. Users need this function to remove a whole row of data.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DRT | | | | Current Status: Passed | | |
| Test title: Test functionality of Delete Row from edit menu | | | | | | |
| Testing approach: We didn’t have access to the source code of the software so, for this testing, we choose the black box testing approach. | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  - Run the program  Click Open from File menu Choose *Test\_DB.xml* from current directory  Click *CONSTRAINT\_TABLE*  -First select a row from the table.  -In the new windows, from Edit menu choose Delete Row.  -Use shortcut of delete too. | PURPOSE  Navigate program to reach a data base.  Select a row to check the functionality of the Delete operation  Removing a row from the table  Check the shortcut functionality for deleting a row | | | EXEPCTED RESULTS  TTC XML Editor pops up.  A row in the table is selected.  The selected row is deleted.  The selected row is deleted | COMMENTS  Pictures [10..12] show the steps of doing these actions |
| Concluding Remarks:  The Delete action works correctly without any problem. Users can delete a row by just using the shortcut or calling the function from the edit menu. | | | | | | |
| Testing Team:  Ali Nouri | | | Date Completed:  11/04 | | | |

## Test Search Function

**Objective:** Check the functionality of search option in the edit menu

**Notes:** This Function used to find an entity from the entire table. We want to check the different scenarios in searching for an entity. Looking for an empty string, look for a string and looking for a non-ascii code are the test targets.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: SFT | | | | Current Status: Passed | | |
| Test title: Test functionality of Search function from edit menu | | | | | | |
| Testing approach: We didn’t have access to the source code of the software so, for this testing, we choose the black box testing approach. | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  -Run the program  Click Open from File menu Choose *Test\_DB.xml* from current directory  Click *CONSTRAINT\_TABLE*  -In the new windows, from Edit menu choose Search.  -In a new box try these values: “CORRECT”, “Bj��rk����oacute�”, “ “. click the Find button | PURPOSE  Navigate program to reach a data base.  Calling the search function  Search for a value in the table | | | EXEPCTED RESULTS  TTC XML Editor pops up.  The search window pops up  It must highlight the value in the table. | COMMENTS  Be sure before searching for “Bj��rk����oacute�”, you already added it in the file.  Pictures [13..15] explain these steps. |
| Concluding Remarks:  The Search function works well. Users can search for any type of string. It iterates through all rows and it highlights all possible match values. In this test we tried a non ascii value, and the software found it without any problem. | | | | | | |
| Testing Team:  Ali Nouri | | | Date Completed:  11/04 | | | |

## Test Replace with A New Value Function

**Objective:** In this test case, we want to check the functionality of the Replace function.

**Notes:** In this test case, you need to search for the same value to be sure all the previous values are replaced by the new one.

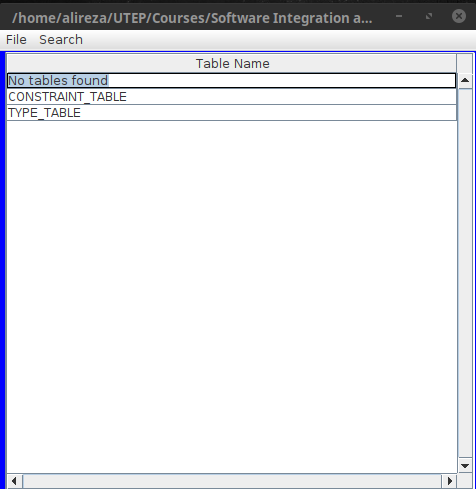
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: SFT | | | | Current Status: Passed | | |
| Test title: Test functionality of Replace from Search function | | | | | | |
| Testing approach: We didn’t have access to the source code of the software so, for this testing, we choose the black box testing approach. | | | | | | |
| STEP  1  2  4 | OPERATOR ACTION  -Run the program  Click Open from File menu choose *Test\_DB.xml* from current directory  Click *CONSTRAINT\_TABLE*  -In the new windows, from Edit menu choose Search. In the new window, put the word “CORRECT” in front of search box, and put the word “WRONG” in front of replace box. Press button Replace ALL.  -Import the table into EXCEL, and try to search for the word “CORRECT” | PURPOSE  Navigate program to reach a data base.  Replace the word CORRECT with another word  Try to check the correctness of the Replace function in another software | | | EXEPCTED RESULTS  TTC XML Editor pops up.  All the CORRECT words in the table must be replaced with the word WRONG.  All the CORRECT words replaced and there is no more of this word in the table | COMMENTS  We could test the functionality of the replacement by the same software, but we preferred to use another software. In this way we can claim we test functions in this software independently.  Pictures [16..18] explain the steps of this test. |
| Concluding Remarks:  The Replace All function worked well, and all possible values replaced with the new one. But consider this function is case sensitive. So, there is a difference between “A” and “a”. We didn’t test this aspect of the function, because we don’t have any idea that it is implemented by purpose or not. | | | | | | |
| Testing Team:  Ali Nouri | | | Date Completed:  11/04 | | | |

# Test Schedule

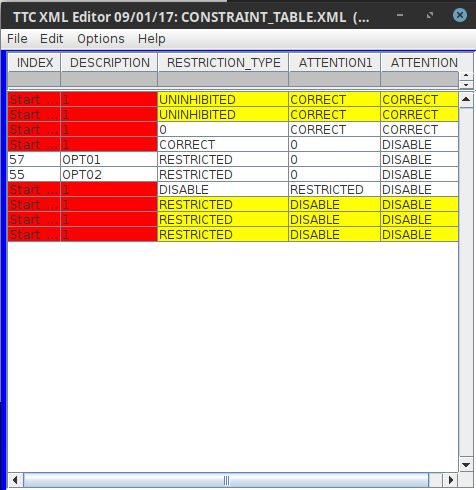
|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
| 11/ April | Alireza Nouri | Write the first edition of the test plan document. |
| 21/April | Alireza Nouri | Final testing the software to be sure the software can be tested by this test plan. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Appendix

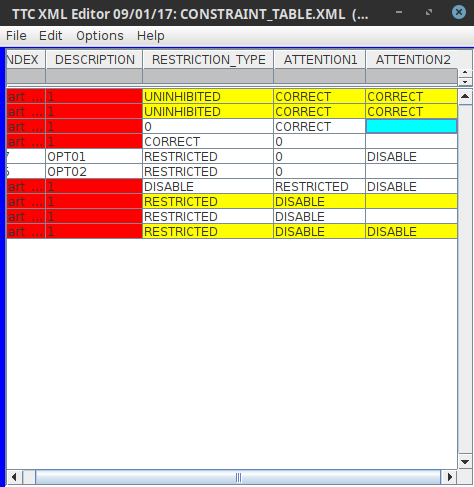
## Screenshots of The Testing



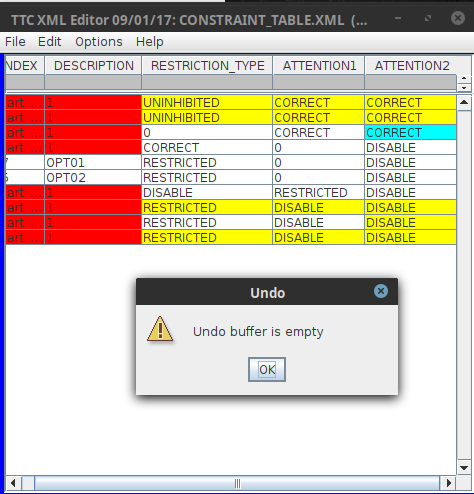
Picture 1



Picture 2



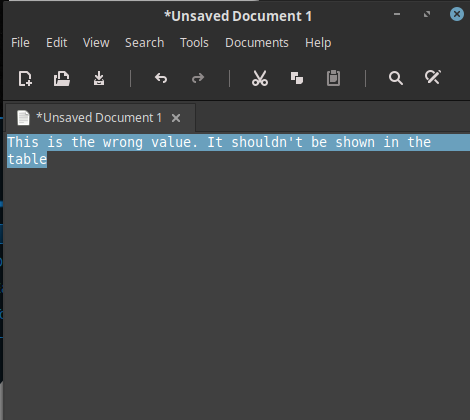
Picture 3



Picture 4



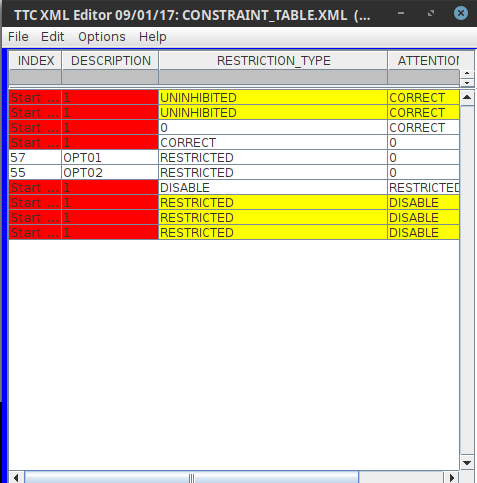
Picture 5



Picture 6



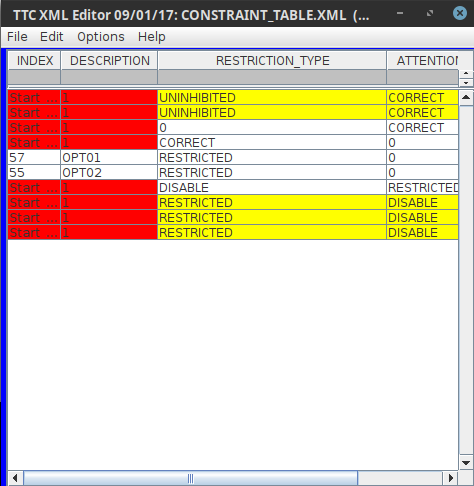
Picture 7



Picture 8



Picture 9



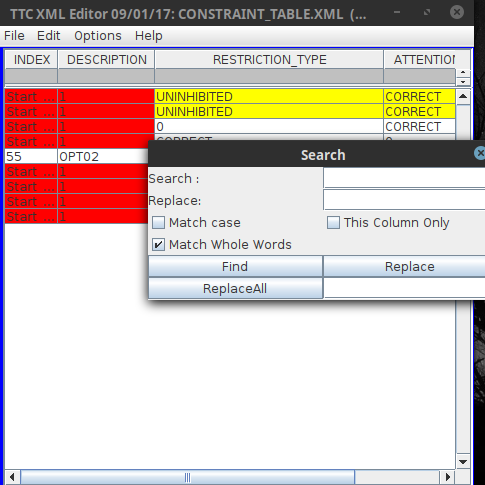
Picture 10



Picture 11



Picture 12



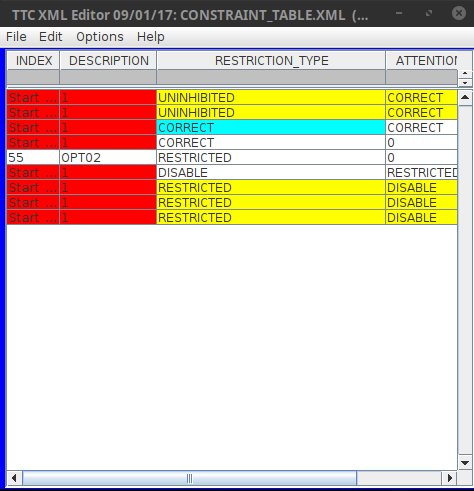
Picture 13



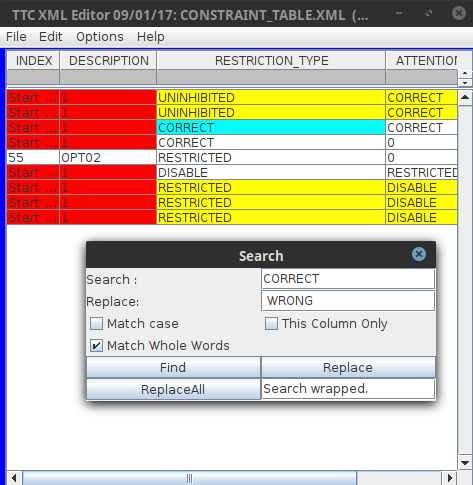
Picture 14



Picture 15



Picture 16



Picture 17



Picture 18

$