Test Plan for Edit Menu Operations

of dbEdit Software

Test plan

Version 2.0

4/17/20

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 4/6/20 |
| Current Release: | 4/17/20 |
| Indicator of Last Page in Document: | $$ |
| Date of Last Review: | 4/13/20 |
| Date of Next Review: |  |
| Target Date for Next Update: |  |

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: Juan A. Gaucin, Carlos I. Vargas, Valeria Macias

Customer: Dr. Steven Roach

Software Team Members: Dr. Steven Roach

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 4/6/20 | Valeria Macias | Edited Sections 1,2,3 and 4 |
| 1.0 | 4/7/20 | Valeria Macias | Edited every section of the document |
| 2.0 | 4/16/20 | Valeria Macias | Edited every section of the document |
| 2.0 | 4/17/20 | Valeria Macias | Finished editing every section of the document |

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](#_Toc22915465)

[Approval ii](#_Toc22915466)

[Document Change Control ii](#_Toc22915467)

[Distribution List ii](#_Toc22915468)

[Change Summary ii](#_Toc22915469)

[1. Introduction 1](#_Toc22915470)

[1.1. Purpose 1](#_Toc22915471)

[1.2. Scope 1](#_Toc22915472)

[1.3. System Overview 1](#_Toc22915473)

[1.4. Suspension and Exit Criteria 1](#_Toc22915474)

[1.5. Document Overview 1](#_Toc22915475)

[1.6. References 1](#_Toc22915476)

[2. Test Items and Features 3](#_Toc22915477)

[3. Testing Approach 4](#_Toc22915478)

[4. Test EDIT MENU 6](#_Toc22915479)

[4.1. Test SUITE: Insert Row 6](#_Toc22915480)

**4.1.1 Test No.: I001: Insert Row……………………………………………………………….7**

**4.2. TEST SUITE: DELETE ROW………………………………………………………………...8**

**4.2.1 Test No.: D001: Delete First Row………………………………………………………..8**

**4.2.2 Test No.: D002: Delete Last Row………………………………………………………...9**

**4.2.3 Test No.: D003: Delete Multiple Middle Rows………………………………………....10**

**4.2.4 Test No.: D0004: Delete Header Row…………………………………………………...11**

**4.3 TEST SUITE: COPY ROW / PASTE ROW…………………………………………………12**

**4.3.1 Test No.: CP001: Copy / Paste First Row………………………………………………12**

**4.3.2 Test No.: CP002: Copy / Paste Last Row………………………………………………13**

**4.3.3 Test No.: CP003: Copy / Paste Multiple Middle Rows………………………………..14**

**4.4. TEST SUITE: UNDO………………………………………………………………………….15**

**4.4.1 Test No.: U001: Undo Insert…………………………………………………………….16**

**4.4.2 Test No.: U002: Undo Delete……………………………………………………………17**

**4.4.3 Test No.: U003: Undo Copy / Paste…………………………………………………….18**

[5. User Interface Testing 19](#_Toc22915481)

[6. Test Schedule 20](#_Toc22915482)

[7. Appendix 21](#_Toc22915484)

# Introduction

This Test Plan is created for the purposes of understanding Test Plans and Reviews for CS 5387 Software Verification and Validation at the University of Texas at El Paso. This Test Plan is designed to test the edit menu operations of a database editing software.

## Purpose

The purpose of this Test Plan is to test the functionality of the edit menu operations of a database editing software. The edit menu operations include: “Insert Row”, “Delete Row”, “Copy Row”, “Paste Row”, and “Undo”. This Test Plan follows a unit test plan approach for two operations (Insert and Delete), then an integration of units for two operations (Copy and Paste), then an integration of units for all operations (Undo Insert, Undo Delete, Undo Copy/Paste). The details of this test plan system are elaborated in the 1.3 System Overview section below.

## Scope

This Test Plan is designed to test the latest version of the software released by Dr. Steven Roach on the Github repository created on April 5, 2020. The software is a database editing software named: dbEdit. The software is designed to allow a user to open and edit .xml tables.

## System Overview

The testing approach for this Test Plan is of unit test, unit test, integration test between two functions, and integration test between all functions as follows: This Test Plan follows a unit test plan approach for the edit menu operations: “Insert” and “Delete” because these two operations can be performed without the other edit menu operations. However, the edit menu operation “Copy” is not testable without the edit menu operation “Paste” and vice versa. Thus, this Test Plan then follows an integration of units approach for “Copy” and “Paste” as these two functions necessarily work together. Finally, the edit menu operation “Undo” can only be performed after other edit menu operations are performed. Thus, this Test Plan takes an integration of units approach for “Undo” as the operation undo is tested after all other operations.

## Suspension and Exit Criteria

Because the final operation, “Undo” is tested in conjunction with the other operations, the Test Suite “Undo” relies heavily on the pass rate of the other operations. Thus, a criticality system is created to ensure that the “Undo” tests are not performed until the proper tests have passed to ensure that any failure in the “Undo” test suite is purely because of a failure in the “Undo” operation. For this reason, the criticality system has three ratings: High, Medium and Low. High criticality tests must pass before performing Low criticality tests.

If a High criticality test fails, testing may continue on High criticality and Medium criticality tests. However, if a High criticality test fails, testing must be suspended before testing any Low criticality tests. Low criticality tests must only be performed once all High criticality tests have passed. A 100% pass rate of all tests is required for the completion of edit menu operation functionality testing.

## Document Overview

The remainder of this document outlines the edit menu features to be tested as well as other features of the software which will not be tested. The Testing Approach is an overview of each test plan. Each test plan focuses on one edit menu operation. Each test plan has more than one test to test the edit menu operation. Following the test plans is a comment on the user interface testing, then a test schedule and finally the appendix.

## References

Spec.docx

DbEditTestPlanAssign.pdf

TestPlanAssign040120.pptx

# 2. Test Items and Features

The following is a table which outlines which features are to be tested and which features are not to be tested.

|  |  |
| --- | --- |
| *Features to be tested* | *Features not to be tested* |
| Copy | Check for Duplicate Rows |
| Delete | Close |
| Edit Menu Operations | Column Selection |
| Insert | Compare |
| Paste | File Menu |
| Undo | File Save |
|  | Filter |
|  | Help |
|  | Open |
|  | Options |
|  | Options Display |
|  | Print |
|  | Search Menu |

# 3. Testing Approach

The following are four tables which describe four test suites. Each test suite tests each of the operations in the Edit Menu: Insert Row, Delete Row, Copy Row/Paste Row, and Undo. Each table contains a description of the Test Suite and each test case within that test suite. The criticality is a dependency rating elaborated in detail in Section 1.4 above.

Table 1: Test Plan 1

|  |  |  |
| --- | --- | --- |
| **TEST SUITE: INSERT ROW** | | |
| **Description of Test Suite** | **This test suite is designed to test the functionality of the Insert Row operation.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| I001 | **Insert a new row in a table.** | **High** |

**Table 2: Test Plan 2**

|  |  |  |
| --- | --- | --- |
| **TEST SUITE: DELETE ROW** | | |
| **Description of Test Suite** | **This test suite is designed to test the functionality of the Delete Row operation.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| D001 | **Delete the first row in a table.** | **Medium** |
| D002 | **Delete the last row in a table.** | **High** |
| D003 | **Delete multiple middle rows in a table.** | **Medium** |
| D004 | **Delete Header row of a table.** | **Medium** |

**Table 3: Test Plan 3**

|  |  |  |
| --- | --- | --- |
| **TEST SUITE: COPY ROW/PASTE ROW** | | |
| **Description of Test Suite** | **This test suite is designed to test the functionality of the Copy Row/Paste Row operations.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| CP001 | **Copy/Paste First Row** | **Medium** |
| CP002 | **Copy/Paste Last Row** | **High** |
| CP003 | **Copy/Paste Multiple Middle Rows** | **Medium** |

**Table 4: Test Plan 4**

|  |  |  |
| --- | --- | --- |
| **TEST SUITE: UNDO** | | |
| **Description of Test Suite** | **This test suite is designed to test the functionality of the Undo operation.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| U001 | **Undo Insert** | **Low** |
| U002 | **Undo Delete** | **Low** |
| U003 | **Undo Copy/Paste** | **Low** |

# Test Edit Menu

Each of the following sections represent a test suite. Each test suite contains a series of tests as shown in the tables in the previous section above. Each test uses the TEST\_DB.xml file in conjunction with the dbEdit system. Each test contains a series of steps which are to be followed sequentially. A description of the purpose of each step and the expected output is included. A test fails if the expected output does not match the actual output. A test passes if the expected output matches the actual output. A test is pending if the test has not been performed or if a test suite has been suspended due to the failure of High criticality tests. The test team should identify themselves at the bottom of each test, additional comments should be made in case of any unforeseen problems or constraints. Finally, the date should be recorded for configuration management purposes.

## Test Suite: Insert Row

**Objective:** To test the functionality of the Insert Row operation in the Edit Menu.

**Notes:** There is only one test in this section because the Insert operation should execute only one function. The Insert operation is strictly a row operation and thus, only one expected outcome exists when executing the Insert operation.

**4.1.1: Test No. I001: Insert Row**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: I001 | | | | Current Status: Pending | | |
| Test title: Insert Row | | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select “Edit”, then “Insert Row (^i)” | This conducts the Insert Row operation. | | | A new 11th row should appear in the table. The row’s first four columns are red. It’s last column is white. There are no letters displayed anywhere on the row. *See appendix d*. |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test Suite: Delete Row

**Objective:** To test the functionality of the Delete operation in the Edit menu.

**Notes:** There are five tests in this section. The Delete function is strictly a row operation. Thus, the deletion functionality is tested on the first, last, middle, and multiple rows. However, any attempts to delete the Header Row should be unsuccessful.

**4.2.1: Test No.: D001: Delete First Row**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: D001 | | | | Current Status: Pending | | |
| Test title: Delete First Row | | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select the first row by selecting the red box in the first row in the first column under “INDEX” named “Start\_Name” | This selects the row to be deleted. | | | The selected box displays a gray outline. *See appendix e.* |  |
| 5 | Select “Edit”, then “Delete Row (^d)” | This conducts the Delete Row operation. | | | The first row should be removed from the table so that there are only 9 rows: 3 red, 2 white, 4 red. *See appendix f.* |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

**4.2.2. Test No.: D002: Delete Last Row**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: D002 | | | | Current Status: Pending | | |
| Test title: Delete Last Row | | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select the last row by selecting the red box in the last row in the first column under “INDEX” named “Start\_Name” | This selects the row to be deleted. | | | The selected box displays a gray outline. *See appendix g.* |  |
| 5 | Select “Edit”, then “Delete Row (^d)” | This conducts the Delete Row operation. | | | The last row should be removed from the table so that there are only 9 rows: 4 red, 2 white, 3 red. *See appendix h.* |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

**4.2.3: Test No.: D003: Delete Multiple Middle Rows**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: D003 | | | | Current Status: Pending | | |
| Test title: Delete Multiple Middle Rows | | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select multiple rows by selected the boxes which contain the labels “CORRECT”, “RESTRICTED” and “DISABLED” under the column labeled “RESTRICTION\_TYPE”. | This selects the rows to be deleted. | | | The selected boxes turn from white to cyan. *See appendix k.* |  |
| 5 | Select “Edit”, then “Delete Row (^d)” | This conducts the Delete Row operation. | | | The selected rows should be removed from the table so that there are only 6 rows: all red in the INDEX column. *See appendix l.* |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

**4.2.4. Test No.: D004: Delete Header Row**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: D004 | | | | Current Status: Pending | | |
| Test title: Delete Header Row | | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select the Header Row by selecting the first box in the Header Row labeled “INDEX” | This selects the rows to be deleted. | | | The selection on the Header Row should not be allowed. Instead the system should rearrange the table rows by INDEX. *See appendix m.* |  |
| 5 | Select “Edit”, then “Delete Row (^d)” | This conducts the Delete Row operation. | | | The table should remain unchanged and it should look exactly as it did in the previous step. *See appendix m.* |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test Copy Row/Paste Row

**Objective:** To test the functionality of the Copy Row and Paste Row operations.

**Notes:** The followings tests take an integration of units approach to test the Copy function with the Paste function because these functions are necessarily used in conjunction together and cannot be tested separately. Note that the user cannot choose where to paste a copied row. Each pasted row must appear at the bottom of the table.

**4.3.1. Test No. CP001: Copy/Paste First Row**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: CP001 | | | | Current Status: Pending | | |
| Test title: Copy/Paste First Row | | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select the First Row by selecting the first red box under the column labeled “INDEX” | This selects the row to be copied. | | | The selected box displays a gray outline. *See appendix e.* |  |
| 5 | Select “Edit”, then “Copy Row (^c)” | This conducts the Copy Row operation. | | | The table should remain unchanged and it should look exactly as it did in the previous step. *See appendix e.* |  |
| 6 | Select “Edit”, then “Paste Row (^v)” | This conducts the Paste Row operation. | | | There should now be 11 rows. The 11th row is a copy of the first row. *See appendix n.* |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

**4.3.2 Test No.: CP002: Copy / Paste Last Row**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: CP002 | | | | Current Status: Pending | | |
| Test title: Copy/Paste Last Row | | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select the Last Row by selecting the last red box under the column labeled “INDEX” | This selects the row to be copied. | | | The selected box displays a gray outline. *See appendix g.* |  |
| 5 | Select “Edit”, then “Copy Row (^c)” | This conducts the Copy Row operation. | | | The table should remain unchanged and it should look exactly as it did in the previous step. *See appendix g.* |  |
| 7 | Select “Edit”, then “Paste Row (^v)” | This conducts the Paste Row operation. | | | There should now be 11 rows. The 11th row is a copy of the 10th row. *See appendix o.* |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

**4.3.3. Test No.: CP003: Copy / Paste Multiple Middle Rows**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: CP003 | | | | Current Status: Pending | | |
| Test title: Copy/Paste Multiple Middle Rows | | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select multiple rows by selected the boxes which contain the labels “CORRECT”, “RESTRICTED”, “RESTRICTED” and “DISABLED” under the column labeled “RESTRICTION\_TYPE”. | This selects the rows to be deleted. | | | The selected boxes turn from white to cyan. *See appendix k.* |  |
| 5 | Select “Edit”, then “Copy Row (^c)” | This conducts the Copy Row operation. | | | The display should remain unchanged and look exactly as it did in the previous step. *See appendix k.* |  |
| 6 | Select “Edit”, then “Paste Row (^p)” | This conducts the Paste Row operation | | | There should now be 14 rows. Under “RESTRICTION\_TYPE”, the 11th row should display “CORRECT”, 12th row should display “RESTRICTED”, the 13th row should display “RESTRICTED” and the 14th row should display “DISABLED”. *See appendix p.* |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test Undo

**Objective:** To test the functionality of the Undo operation.

**Notes:** The following tests are to be conducted using the TEST\_DB.xml file. The tests take an integration of units approach to test the Undo function against all of the other functions because the undo function is meant to undo the edits done by the other functions. Please suspend testing here if less than a 90% pass rate has been achieved by sections 3.1, 3.2 and 3.3 tests. Please continue otherwise.

**4.4.1. Test No.: U001: Undo Insert**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: U001 | | | | Current Status: Pending | | |
| Test title: Undo Insert | | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select “Edit”, then “Insert Row (^i)” | This conducts the Insert Row operation. | | | A new 11th row should appear in the table. The row’s first four columns are red. It’s last column is white. There are no letters displayed anywhere on the row. *See appendix d*. |  |
| 5 | Select “Edit”, then “Undo (^z)” | This conducts the Undo operation. | | | The 11th row should disappear and the table should look exactly as it did in step 3. *See appendix c.* |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

**4.4.2: Test No.: U002: Undo Delete**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: U002 | | | Current Status: Pending | | |
| Test title: Undo Delete | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select the last row by selecting the red box in the last row in the first column under “INDEX” named “Start\_Name” | This selects the row to be deleted. | | The selected box displays a gray outline. *See appendix g.* |  |
| 5 | Select “Edit”, then “Delete Row (^d)” | This conducts the Delete Row operation. | | The last row should be removed from the table so that there are only 9 rows: 4 red, 2 white, 3 red. *See appendix h.* |  |
| 7 | Select “Edit”, then “Undo (^z)” | This conducts the Undo operation. | | The last row should reappear at the bottom of the table. The window should look exactly as it did in step 3. *See appendix c.* |  |
| Concluding Remarks: | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | |

**4.4.3: Test No.: U003: Undo Copy/Paste**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: U003 | | | Current Status: Pending | | |
| Test title: Undo Copy Paste | | | | | |
| Testing approach: To perform this test, the testing environment should have access to the dbEdit software and the TEST\_DB.xml file. The following operations may be done manually or may be automated. | | | | | |
| STEP  1 | OPERATOR ACTION  Run dbEdit software. | PURPOSE  This runs the software to be tested. | | EXEPECTED RESULTS  The system displays a window named “TTC XML Database Table Viewer 09/01/17”. *See appendix a.* | COMMENTS |
| 2 | Select “File”, then “Open”, then select “TEST\_DB”, and click “Open”. | This opens the database which contains the table which will be used to test the edit menu operations. | | The window displays three rows under “Table Name”. The first row displays “BIG\_TABLE”, the second row displays, “CONSTRAINT\_TABLE”, and the third row displays, “TYPE\_TABLE.” *See appendix b.* |  |
| 3 | Double click on CONSTRAINT\_TABLE | This takes the tester to the window where the Edit Menu operations are accessible. | | The system opens a new window which contains ten rows. *See appendix c*. |  |
| 4 | Select the Last Row by selecting the last red box under the column labeled “INDEX” | This selects the row to be copied. | | The selected box displays a gray outline. *See appendix g.* |  |
| 5 | Select “Edit”, then “Copy Row (^c)” | This conducts the Copy Row operation. | | The table should remain unchanged and it should look exactly as it did in the previous step. *See appendix g.* |  |
| 6 | Select “Edit”, then “Paste Row (^v)” | This conducts the Paste Row operation. | | There should now be 11 rows. The 11th row is a copy of the 10th row. *See appendix o.* |  |
| 8 | Select “Edit”, then “Undo (^z)” | This completes the Undo operation. | | The 11th row should disappear and the table should look exactly as it did in step 3*. See appendix c.* |  |
| Concluding Remarks: | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | |

# User Interface Testing

The following tests the user interface of the system as it concerns the Edit Menu. The tests are represented as questions and each question needs a response of “Yes” to pass the test.

|  |  |
| --- | --- |
| Tests | Pass/Fail |
| *Spelling* |  |
| In the file menu, is the edit button displayed as “Edit”? |  |
| In the edit menu, is the insert button displayed as “Insert Row (^i)”? |  |
| In the edit menu, is the delete button displayed as “Delete Row (^d)”? |  |
| In the edit menu, is the copy button displayed as “Copy Row (^c)”? |  |
| In the edit menu, is the paste button displayed as “Paste Row (^p)”? |  |
| In the edit menu, is the undo button displayed as “Undo”? |  |
|  |  |
| *Usability* |  |
| Do each of the operations execute using one click from the edit menu? |  |
| Is the edit menu displayed at the top of the table view window? |  |

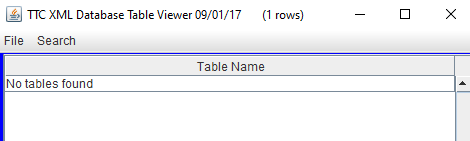
# Test Schedule

The following is a table which schedules the task, the task completion date, the persons to conduct the task and a description of the task for purposes of scheduling organization and completion. The test schedule will be determined at a later date.

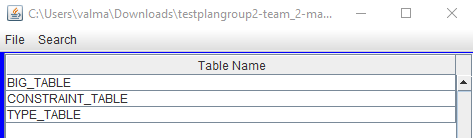
|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Appendix

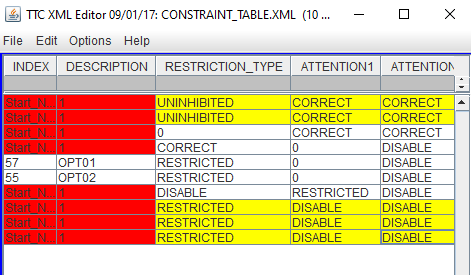
1. The initial display when the program is run:



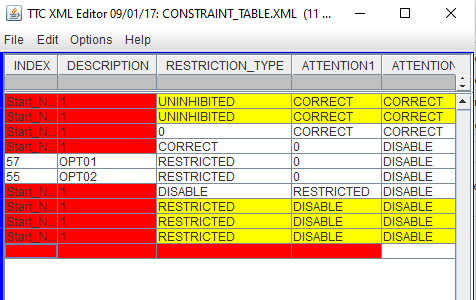
1. The display after the user opens “TEST\_DB.xml”:



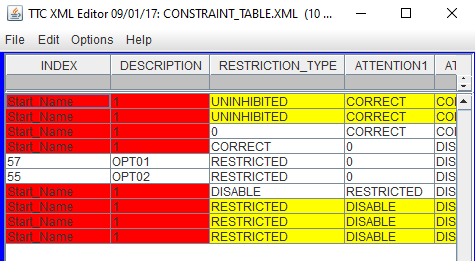
1. The display after the user double clicks on “CONSTRAINT\_TABLE”:



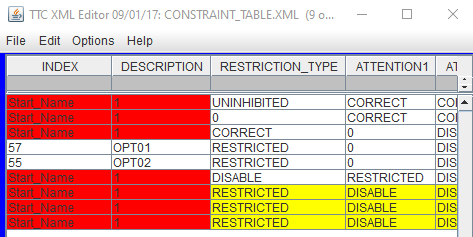
1. The display after the Insert Row operation:



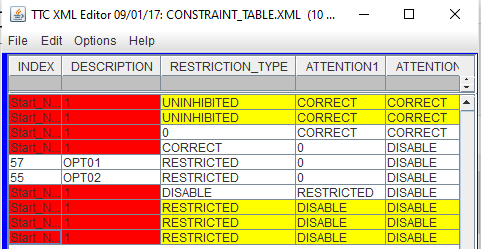
1. First row selected display:



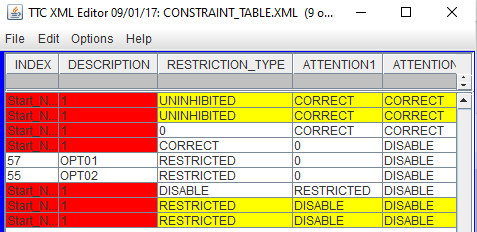
1. Display after first row is deleted:



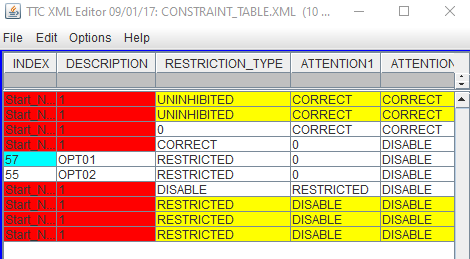
1. Last row selected display:



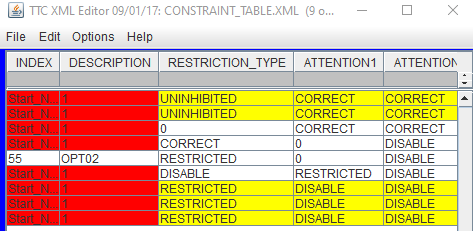
1. Display after last row is deleted:



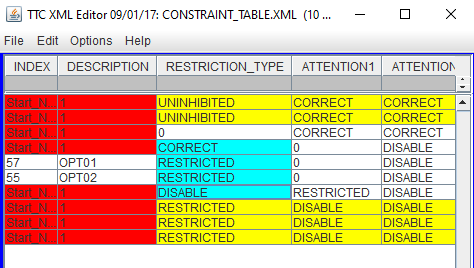
1. 5th row is selected display:



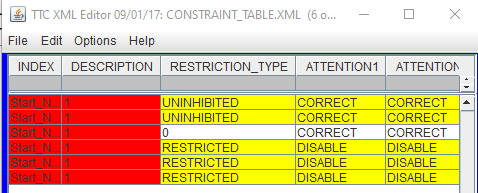
1. Display after 5th row is deleted:



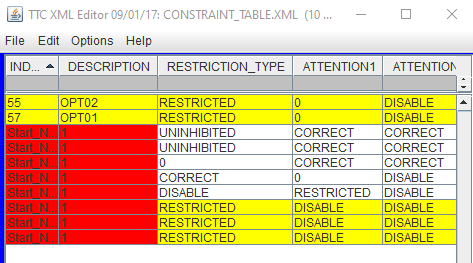
1. Rows 4 through 7 are selected display:



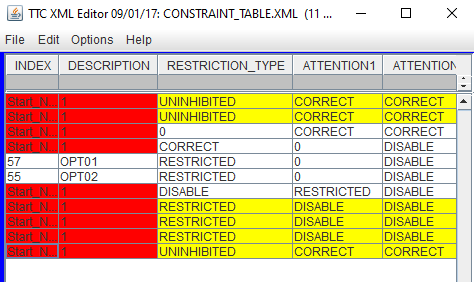
1. Display after rows 4 through 7 are deleted:



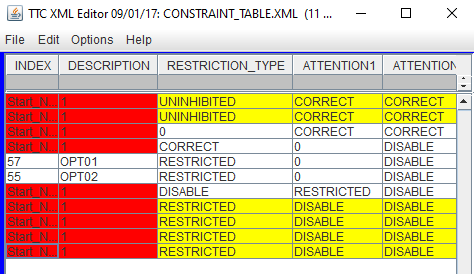
1. Display after “Index” is selected in the Header Row:



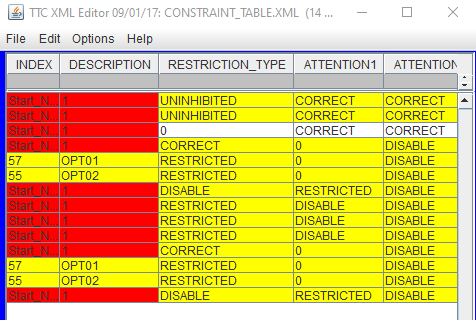
1. Display after the first row is copied and pasted:



1. Display after the last row is copied and pasted:



1. Display after the 4th through 7th rows are copied and pasted:



$$