Database Table Editor

Test plan

Version 4

April 22, 2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | April 8, 2020 |
| Current Release: | April 10, 2020 |
| Indicator of Last Page in Document: | \*\* |
| Date of Last Review: |  |
| 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: Dr. Roach

Team Members: Torres, Andrea | Pincus, Nicholas R.| De La Cruz, Julio

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1 | April 8, 2020 | Andrea Torres | Initial Release of Document |
| 2 | April 19 2020 | Andrea Torres | Table is updated based on reviewers’ comments |
| 3 | April 22 | Andrea Torres | Tests for comparison are added. |
| 4 | April 24, 2020 | Andrea Torres | Tests for duplicate corrected. |

Table of Contents

[Document Control ii](#_Toc38648432)

[Approval ii](#_Toc38648433)

[Document Change Control ii](#_Toc38648434)

[Distribution List ii](#_Toc38648435)

[Change Summary ii](#_Toc38648436)

[1. Introduction 4](#_Toc38648437)

[1.1. Purpose 4](#_Toc38648438)

[1.2. Scope 4](#_Toc38648439)

[1.3. System Overview 4](#_Toc38648440)

[1.4. Suspension and Exit Criteria 4](#_Toc38648441)

[1.5. Document Overview 4](#_Toc38648442)

[1.6. References 4](#_Toc38648443)

[2. Test Items and Features 5](#_Toc38648444)

[3. Testing Approach 6](#_Toc38648445)

[4. Compare Tests 7](#_Toc38648446)

[4.1. Test DBComp1 7](#_Toc38648447)

[4.2. Test DBComp2 10](#_Toc38648448)

[4.3. Test DBComp3 11](#_Toc38648454)

[5. Duplicate Tests 12](#_Toc38648455)

[5.1. Test DBDup1 12](#_Toc38648461)

[5.2.](#_Toc38648463)  [Test DBDup2 13](#_Toc38648462)

[5.3. Test DBDup3 14](#_Toc38648464)

[6. User Interface Testing 15](#_Toc38648465)

[7. Test Schedule 16](#_Toc38648466)

# Introduction

<< This section gives introductory information regarding the project, the system to be tested, and the testing approach.>>

## Purpose

<< Identify the project and stipulate the test plan purpose by indicating what the document contains (e.g., organizational responsibilities, test approach, test schedule. There are generally four different types of test plans: project test plan that describes the overall strategy for testing; the system test plan that describes the system from the customer’s point of view; integration test plan that describes integration of units and subsystems; unit test plan that describes modules or classes. This section needs to identify which of these this document is.>>

## Scope

<<Specify the project software releases/versions encompassed by the plan. >>

## System Overview

<<Describe the system to be exercised by the testing approach specified in the plan. This overview serves to identify aspects of the system operation that will be the focus of the plan’s testing approach. This should align with the systems overview of other documents in the project.>>

## Suspension and Exit Criteria

<< “suspension criteria” describes when we suspend testing, to be resumed at a later time. For example, if 40% of the test cases fail, or if any of the critical test cases fail. If there are no suspension criteria, indicate that all tests cases will be executed. “Exit criteria” indicates when testing stops. This could be based on run rate (number of test cases run divided by number of test cases specified) or pass rate (number of test cases passed divided by number of test cases run, or test cases passed divided by number of test cases specified). Nominally, we expect to run all of the specified tests. We want the pass rate to be high. We might specify that all critical tests must pass, and 90% of the non-critical must pass. In general, we want this to be high. >>

## Document Overview

<<Describe the remainder of the document.>>

## References

<<List all the references applicable to the test plan. Generally, this includes project standards, SRS, SDD, and a product assurance plan.>>

# Test Items and Features

<< This section describes the test items (e.g., components, classes, functions or methods) and the features to be tested. It may also list features not to be tested. A class diagram is useful. A table of features is useful. >>

# Testing Approach

These tests will determine the accuracy of the compare and duplicate row features. These testes were created to determine if the features work as expected. These are not stress tests, they are barley acceptance tests in that if these tests pass the system displays expected performance, and if the test fails then these features are not working and we are able to conclude that the features are not implemented in a manner that is acceptable for general release.

Table 1: Test Plan

|  |  |  |
| --- | --- | --- |
| **TEST SUITE DBCompDup** | | |
| **Description of Test Suite** | This test suite will befocused on the ability to compare files to other versions and check for duplicate rows | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| DBComp1 | **To determine if the compare option works as expected when a row is added.** | **Medium** |
| DBComp2 | **To determine if the compare option works as expected when a row is deleted.** | **Medium** |
| DBComp3 | **To determine if the compare option works as expected when a row is edited.** | **Medium-High** |
| DBDup1 | **To determine if the duplicate option works as expected when there are no duplicates.** | **Medium** |
| DBDup2 | **To determine if the duplicate option works as expected when a new unique row is added.** | **Medium** |
| DBDup3 | **To determine if the duplicate option works as expected when a new duplicate row is added.** | **Medium** |

# Compare Tests

The purpose of this section is to:

* Determine if the Compare option compares two versions of a table.

for each test case, complete the following:

* open dbEdit software
* Select file
* Select Open
* Select Test\_DB

## Test DBComp1

**Objective: Compare two tables that share only one row change.**

**Notes:** This test is to help determine if the software is able to detect change in one row.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBComp1 | | | | | Current Status: Pass | |
| Test title: Data Base Comparison one. | | | | | | |
| Testing approach: For this test we will be using the TEST\_DB\_\_CONSTRAINT\_TABLE, this will be done manually. | | | | | | |
| STEP  1  2  3  4  5  6  7  8  9  10  11  12  13 | OPERATOR ACTION  Select CONSTRAINT\_TABLE  Select “Edit” from menu bar.  Select “Insert Row” from “Edit” Dropdown menu  Type in 55 into the first column of the table in the last row.  Type in 1 into the second column of the table in the last row.  Select File drop down menu.  Select “Save as” from “File” dropdown menu.  Type “TEST\_DB\_CONSTRAINTCHANGE\_TABLE” and select save.  Input “4/20/2020” into the date box. Input “Tester” into the author box. Input “Added a row” into the description box.  Select “OK” in the “File write” popup box.  Go to “CONSTRAINT\_TABLE” view.  Open “File” dropdown menue. Select “Compare table to another version”  Select “TEST\_DB\_\_CONSTAINT\_ TABLE.XML” Press “Open” | PURPOSE  To have a table that is the foundation for the test.  Get to edit menu so that we may change the table.  To have a new row to make changes to the table.  To have a row that is different from the original table.  To have a row that is different from the original table.  To open the file menu so that we may be able to create a second table that is similar.  To start the process of saving a new table that is like the original table.  To save a new table.  To fill in requested information.  To proceed with the workflow.  To proceed with the workflow.  To begin the comparison.  To select the table to compare. | EXEPCTED RESULTS  A new window is open with CONSTRAINT\_Table  Dropdown with edit results appear.  A new empty row is added to the bottom of the table.  “55” is in the first column last row.  “1” is in the second column last row.  Dropdown of the file menu appears.  Save File window opens.  XML History Input window opens.  File write popup appears showing that the table was saved. The file is created and saved.  “File write” popup box is closed. New table is created.  Table is displayed in window.  Open a file window is displayed with available tables.  Compare result popup window opens and shows 1 mismatch. Added row is shown in a new window. | | | COMMENTS |
| Concluding Remarks: This tests passes. We select the original table to compare because when we save as the window is now holing the “new” table so we need to compare the original table to the newly saved table. | | | | | | |
| Testing Team:  Torres, Andrea | Pincus, Nicholas R.| De La Cruz, Julio | | | | Date Completed: April 24, 2020 | | |

## Test DBComp2

**Objective: To determine if the compare option works as expected when a row is deleted.**

**Notes:** This test determines of the compare option works when a row is deleted. Also we compare a file that has been edited but not saved to the original version of the file. This test should take no more than 15 min.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBComp2 | | | | | Current Status: Passed | |
| Test title: Database Comparison 2 | | | | | | |
| Testing approach: For this test we will be using the TEST\_DB\_\_CONSTRAINT\_TABLE, this will be done manually. | | | | | | |
| STEP  1  2  3  4  5  6 | OPERATOR ACTION  Select CONSTRAINT\_TABLE  Select “Edit” from menu bar.  Highlight the 5th row with contents “57| OPTO1|RESTRICTED…”  Select “Delete Row” from “Edit” Dropdown menu  Select “File” dropdown menu then select “compare table to another version”  Select “TEST\_DB\_\_CONSTRAINT\_ TABLE.XML” then Select “Open”. | PURPOSE  To have a table that is the foundation for the test.  Get to edit menu so that we may change the table.  To select a row to delete later.  To change the table.  To select a table to compare.  To have a table to compare to. | | EXEPCTED RESULTS  A new window is open with CONSTRAINT\_TABLE  Dropdown with edit results appear.  The selected row is highlighted.  .  The selected row is removed from the table.  File explorer window pops up.  Compare Result popup window opens and shows 1 change. The table contains the row that we had previously deleted. | | COMMENTS |
| Concluding Remarks: For this test we did not create a second file we just compared a file that is not saved to the original file. | | | | | | |
| Testing Team:  Torres, Andrea | Pincus, Nicholas R.| De La Cruz, Julio | | | Date Completed: | | | |

## Test DBComp3

**Objective: To determine if the compare option works as expected when a row is edited.**

**Notes:** This test determines of the compare option works when a row is deleted. Also, we compare a file that has been edited but not saved to the original version of the file. This test should take no more than 15 min.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBComp3 | | | | Current Status: Passed | | |
| Test title: Database Comparison three | | | | | | |
| Testing approach: For this test we will be using the TEST\_DB\_\_CONSTRAINT\_TABLE, this will be done manually. | | | | | | |
| STEP  1  2  3  4 | OPERATOR ACTION  Select CONSTRAINT\_TABLE  Double click on the fifth column “Attention2” first row. Change contents from “CORRECT” to “DISABLE”  Select “File” dropdown menu then select “compare table to another version”  Select “TEST\_DB\_\_CONSTRAINT\_ TABLE.XML” then Select “Open”. | PURPOSE  To have a table that will be the original table for this test.  To change the contents of the table.  To select a table to compare.  To have a table to compare to. | | | EXEPCTED RESULTS  A new window is open with CONSTRAINT\_Table  The first row is changed.  File explorer window pops up.  Compare Result popup window opens and shows 1 change. The table contains the row that we had previously deleted. | COMMENTS |
| Concluding Remarks: This test has demonstraited expected behavior when the system is running as expected. | | | | | | |
| Testing Team:  Torres, Andrea | Pincus, Nicholas R.| De La Cruz, Julio | | | Date Completed: April 22, 2020 | | | |

# Duplicate Tests

The purpose of this section is to:

* Determine if the Duplicate functionality works as expected by conducting tests.

for each test case, complete the following:

* open dbEdit software
* Select file
* Select Open
* Select Test\_DB

## Test DBDup1

**Objective: To determine if the duplicate option works as expected when there are no duplicates.**

**Notes:** This test determines if the duplicate option works when there are no duplicate rows. This test should take no more than 15 min.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: DBDup1 | | | | Current Status: Passed | |
| Test title: Database Duplicate No Duplicates | | | | | |
| Testing approach: For this test we will be using the NODUPLICATE\_TABLE, this will be done manually. To conduct this test first open the dbEdit.jar and then run TEST\_DB file. Select NODUPLICATE\_TABLE. This will open the NODUPLICATE\_TABLE, it is observable that there are no duplicates between any row. | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Select NODUPLICATE\_TABLE  Select “File” menu option.  Select “Check for Duplicate Rows” | PURPOSE  This step is to start the test properly.  To open the “File” dropdown menu.  To look for duplicates in the table. | EXEPCTED RESULTS  NODUPLICATE\_TABLE is opened. The table does not have any duplicate rows.  “File” drop-down meu is displayed.  “Duplicate Check” popup window is displayed. Window shows 0 duplicates. | | COMMENTS |
| Concluding Remarks: Software works as expected. | | | | | |
| Testing Team:  Torres, Andrea | Pincus, Nicholas R.| De La Cruz, Julio | | | | Date Completed: April 24, 2020 | |

## Test DBDup2

**Objective: To determine if the duplicate option works as expected when a new unique row is added.**

**Notes:** This test determines if the duplicate option works when a row that is not a duplicate row is added to a table that does not have duplicate rows. This test should take no more than 15 min.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBDup2 | | | | Current Status: Passed | | |
| Test title: Database Duplicate No Duplicates add a row that is not a duplicate. | | | | | | |
| Testing approach: Testing approach: For this test we will be using the NODUPLICATE\_TABLE, this will be done manually. To conduct this test first open the dbEdit.jar and then run TEST\_DB file. Select NODUPLICATE\_TABLE. This will open the NODUPLICATE\_TABLE, it is observable that there are no duplicates between any row. The tester will manually add a row that does not have a duplicate. The table has 5 columns if something were to go wrong when inputting new information, it should not fail in the first middle or last columns. After each entry, the row should still be unique and there should be no yellow highlighting because all the rows are unique and not duplicates of each other. | | | | | | |
| STEP  1  2  3  4  5  6  7  8 | OPERATOR ACTION  Select NODUPLICATE\_TABLE  Verify the table has no duplicates by looking for yellow highlighting.  Select “Edit” from the top tool bar. Select “Insert Row” from the dropdown menu.  Populate the first column of new row “Index” with “56”.  Populate the third column “RESTRICTION\_TYPE” with “0”.  Populate the fifth column “ATTENTION2” with “CORRECT”  Select “File” menu option.  Select “Check for Duplicate Rows” | PURPOSE  This step is to start the test properly.  To visually verify that the software did not mistakenly highlight a row.  To create a new empty row to later in the test populate.  To populate the first cell in the new row.  To populate the middle cell in the new row.  To populate the last cell in the new row.  To open the “File” dropdown menu.  To look for duplicates in the table. | | | EXEPCTED RESULTS  NODUPLICATE\_TABLE is opened.  No rows are highlighted yellow because the table does not have any duplicates.  A new row is added, the row is not highlighted yellow because the empty row is unique.  The row is still unique no row is highlighted yellow.  The row is still unique no row is highlighted yellow.  The row is still unique no row is highlighted yellow.  “File” drop-down meu is displayed.  “Duplicate Check” popup window is displayed, shows 0 duplicates. | COMMENTS |
| Concluding Remarks: Software works as expected | | | | | | |
| Testing Team:  Torres, Andrea | Pincus, Nicholas R.| De La Cruz, Julio | | | Date Completed: April 24, 2020 | | | |

## Test DBDup3

**Objective: To determine if the duplicate option works as expected when a new duplicate row is added.**

**Notes:** This test determines if the duplicate option works when a row that is a duplicate row is added to a table that does not have duplicate rows. This test should take no more than 15 min.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBDup3 | | | | Current Status: Pending | | |
| Test title: Database Duplicate No Duplicates | | | | | | |
| Testing approach: To conduct this test first open the dbEdit.jar and then run TEST\_DB file. Select NODUPLICATE\_TABLE. This will open the NODUPLICATE\_TABLE, it is observable that there are no duplicates between any row, this will be done manually. To conduct this test first open the dbEdit.jar and then run Dup\_DB file. No\_Dups\_Table. This will open the No\_Dups\_Table, it is observeable that there are no duplicates between any row. | | | | | | |
| STEP  1  2  3  4  5  6 | OPERATOR ACTION  Select NODUPLICATE\_TABLE  Verify the table has no duplicates by looking for yellow highlighting.  Select “Edit” from the top tool bar. Select “Insert Row” from the dropdown menu.  Populate the row with “57”, “OPT0 1”, “RESTRICTED”, “0”, and “DISABLE” into their respected column.  Select “File” menu option.  Select “Check for Duplicate Rows” | PURPOSE  This step is to start the test properly.  To visually verify that the software did not mistakenly highlight a row.  To create a new empty row to later in the test to populate.  To populate the newly created row with information that would make the row a duplicate.  To open the “File” dropdown menu.  To look for duplicates in the table. | | | EXEPCTED RESULTS  NODUPLICATE\_TABLE is opened.  No rows are highlighted yellow because the table does not have any duplicates.  A new row is added, the row is not highlighted yellow because the empty row is unique.  The new row is populated and is the duplicate of the fifth row in the table. Once the row is fully populated both the new and the original row are highlighted yellow.  “File” drop-down meu is displayed.  “Duplicate Check” popup window is displayed, shows 1 duplicate. | COMMENTS  Sometimes you must highlight the table to get the rows to be the correct coloring. |
| Concluding Remarks: Software works as expected, alternatively steps 3 and 4 can be replaced with selecting the fifth row and using the copy row and paste row functions found in the Edit drop down menu. | | | | | | |
| Testing Team:  Torres, Andrea | Pincus, Nicholas R.| De La Cruz, Julio | | | Date Completed: April 24, 2020 | | | |



# User Interface Testing

These tests have been integrated into the pervious tests.

# Test Schedule

|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
| DBComp1 | Andrea Torres |  |
| DBComp2 | Andrea Torres |  |
| DBComp3 | Andrea Torres |  |
| DBDup1 | Andrea Torres |  |
| DBDup2 | Andrea Torres |  |
| DBDup3 | Andrea Torres |  |
| DBComp1 | Andrea Torres |  |
| DBComp2 | Andrea Torres |  |
| DBComp3 | Andrea Torres |  |