

**PMR Insight Collective Knowledge (PICK)**  
**Test Plan**  
Version 0.3  
4/28/2020

## Document Control

### Approval

The Guidance Team and the customer shall approve this document.

### Document Change Control

Initial Release:	0.1
Current Release:	0.3
Indicator of Last Page in Document:	[END]
Date of Last Review:	4/27/2020
Date of Next Review:	4/30/2020
Target Date for Next Update:	4/30/2020

### 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. Ann Gates, Dr. Steve Roach, Mr. Jake Lasley

Clients:

Mr. Vincent Fonseca, Mr. Baltazar Santaella, Ms. Herandy Vazquez, and Mr. Erick De Nava

Software Team Members:

Mr. Anthony Desarmier, Mr. Angel Villalpando, Mr. Mario Delgado, Mr. David Rayner,  
Mr. Valentin Becerra, Mr. Jorge Garcia

### Change Summary

The following table details changes made between versions of this document

Version	Date	Modifier	Description
0.1	04/07/2020	Anthony DesArmier	Added Template
0.2	04/14/2020	Angel Villalpando	Completed sections 1.1-1.4
0.2	04/15/2020	David Rayner	Completed 1.5, added suite to section 3, and two test cases to section 4.
0.2	4/15/2020	Anthony DesArmier	Formatting, grammar
0.3	04/27/2020	Angel Villalpando	Completed Test case tables for TBM 4, 5, 6.
0.3	04/27/2020	Jorge Garcia	Completed Test case tables for TBM 1-3 and 7-9.
0.3	04/27/2020	David Rayner	Completed test case tables for ING 1-5 and DP 1-2. Completed section 6.
0.3	04/27/2020	Valentin Becerra	Completed test case tables for Graphing GPH 1-7
0.3	04/27/2020	Anthony DesArmier	Completed test case tables for Data persistence DP 3-5.
0.3	04/27/2020	Mario Delgado	Completed test case tables for Data persistence DP 6-7.

Test Plan	Keikaku 企画	Date 4/28/2020 11:27 AM	Page ii
-----------	------------	----------------------------	------------

## TABLE OF CONTENTS

<b>DOCUMENT CONTROL.....</b>	<b>II</b>
<b>APPROVAL.....</b>	<b>II</b>
<b>DOCUMENT CHANGE CONTROL .....</b>	<b>II</b>
<b>DISTRIBUTION LIST .....</b>	<b>II</b>
<b>CHANGE SUMMARY .....</b>	<b>II</b>
<b>1. INTRODUCTION.....</b>	<b>4</b>
<b>1.1. PURPOSE.....</b>	<b>4</b>
<b>1.2. SCOPE .....</b>	<b>4</b>
<b>1.3. SYSTEM OVERVIEW .....</b>	<b>4</b>
<b>1.4. SUSPENSION AND EXIT CRITERIA .....</b>	<b>4</b>
<b>1.5. DOCUMENT OVERVIEW .....</b>	<b>4</b>
<b>1.6. REFERENCES .....</b>	<b>5</b>
<b>2. TEST ITEMS AND FEATURES .....</b>	<b>6</b>
<b>3. TESTING APPROACH .....</b>	<b>7</b>
<b>4. TESTS.....</b>	<b>8</b>
<b>4.1. START INGESTION.....</b>	<b>9</b>
<b>5. TEST SCHEDULE .....</b>	<b>11</b>
<b>6. OTHER SECTIONS.....</b>	<b>32</b>
<b>7. APPENDIX.....</b>	<b>33</b>

# 1. Introduction

## 1.1. Purpose

The purpose of this document is to outline the Test Plan for the PMR Insight Collective (PICK) system. This document will include the organizational responsibilities, the test approach, and the test schedule. This document will primarily discuss testing from the customer's point of view and should not be considered a general testing strategy, an integration test plan, or a unit test plan. By conducting the test cases proposed in this document, the customer should be able to demonstrate that the system performs that which it is intended to do.

## 1.2. Scope

The PMR Insight Collective Knowledge (PICK) is the software system for which this Test Plan is written for. PICK is a software system to help Prevent, Mitigate, and Recover Analysts analyze vast amounts of data collected during an Adversarial Assessment (AA) by allowing them to quickly search through, view, correlate, and build visual documents which help explain the AA itself to uninvolved personnel. The customers - in this case PMR Analysts - currently must sift through the vast amounts of generated data from the AA by hand which severely hinders their workflow and efficiency in developing a report with visual aids for which to explain the nature of the AA to other personnel.

PICK will allow the customers to insert all the data generated from an AA into its system and display an organized, searchable database of that information. The customers can then quickly and efficiently find and correlate relevant data events together and help craft timelines which describe the significant events and their relations to one another during the AA. PICK will then assist the customers in crafting a visual representation of these series of events as attack graphs in order to help visualize the timeline of the AA. This assistance of analyzing the data generated by the AA and constructing visual representations of significant events will substantially reduce the time and work hours needed by the customers to understand and construct a report on the results of the AA to deliver to other personnel.

## 1.3. System Overview

The PICK system utilizes several python libraries for the graphical user interface which must be tested to ensure that they perform their desired tasks. Additionally, the system heavily interacts with the Splunk Extract, Load, and Transform (ELT) system. The interaction with this system requires testing to ensure that the data sent to a from it follow the specifications outlined by the design. Finally, ensuring that the system correctly creates vectors, each with respective log entries, is important to the overall success of this system. These items are the focal points for the testing outlined in this document.

## 1.4. Suspension and Exit Criteria

If at any point a critical test fails, testing will be suspended. Critical tests are intended to assess the functionality of the major components within the system. If any of these major components are not functioning as intended, several subsequent tests dependent on this component will also fail or will not be testable. For this reason, testing shall be suspended source code redeveloped to restore functionality to such major components.

Once all critical tests have passed, testing shall be complete. Once critical components are exhibiting the desired behaviors, then the system satisfies the core requirements laid out in the initial specification of the system.

## 1.5. Document Overview

The test plan document consists of the following sections:

### Introduction:

This section describes the overview of the testing plan. It includes the purpose of the document, the overall scope of the project, and the suspension, exit criteria regarding system tests to be run.

TestPlan	Keikaku 企画	Date 4/20/2020	Page 4
----------	------------	-------------------	-----------

**Test Items and Features:**

This section describes the testing items (e.g. components, classes, functions or methods) and the features to be tested.

**Testing Approach:**

This section describes the testing approach we the development team are to establish. The type of tests to be run in order to test system functions. Each test is to contain a description and unique test identifier.

**Test cases:**

This section describes the tests that were run, including test input, test procedures and outcomes. Each test is divided by the following sections: test number, current status, title, approach, step, operator action, purpose, expected results, comments, remarks, conclusion, date completed, and team that performed the test.

**User Interface Testing:**

This section describes the interaction between the system and user components Including consistent terminology, shortcut keys, menu selections, and presentation, flexibility in navigation between windows and interface elements and potential error handling that will inform user of critical operations.

**Test Schedule:**

This section describes the completion dates of each test.

**Other:**

This section describes the other potential test documentation such as:

- Test Management Requirements: how testing is to be managed; a delineation of responsibilities of each project organization involved with testing
- Staffing and training needs: delineate the responsibilities of those individuals who are to perform the testing, level of skill required, and training to be provided
- Environmental Requirements: describe the hardware (including communication and network equipment) needed to support testing; describe configuration of hardware components on which software and database to be tested are to operate.
- Software Requirements: describe the software needed to support testing; include the software code and databases that are object of the testing. Also include software tools such as compilers, CASE instruments and simulators that are needed to model the user's operational environment.
- Risk and contingencies
- Cost: include an estimate of costs.
- Approvals
- Test Deliverables

**Appendix:**

References of expected output and explicit directions for analysis of output.

## 1.6. References

- [1] V. Becerra, A. DesArmier, J. Garcia, D. Rayner, A. Villalpando and Mario Delgado, "Keikaku\_SDD\_V2," El Paso, 2020.

TestPlan	Keikaku 企画	Date 4/20/2020	Page 5
----------	------------	-------------------	-----------

## 2. Test Items and Features

**Feature:** File Ingestion

Class: Validator

Class: SplunkManager

**Feature:** File Cleansing

Class Validator

**Feature:** File Validation

Class: Validator

Class: EnforcementActionReport

**Feature:** Log entry to vector assignment

Class: LogEntry

Class: IDDict

Class: Vector

**Feature:** Sort/Filter log entries and nodes

Class: Sort

Class: Filter

**Feature:** Export vector table

Class: ExportGraph

Class: Vector

**Feature:** Export vector graph

Class: ExportTable

Class: Vector

**Feature:** Graphing

Class: GraphEditor

Class: GraphEditorScene

Class: GraphEditorView

Class: GraphEditorWindow

Class: NodeItem

Class: RelationshipItem

Class: VectorItemGroup

**Feature:** Search and Filter

Class: Sort

Class: Filter

**Feature:** Data storage

Storage is to be done through a file system (pickle serialization operation)

**Feature:** Lead-Host data management

Class: Sync

Class: ProjectMerge

**Feature:** Commit management

Class: History

TestPlan	Keikaku 企画	Date 4/20/2020	Page 6
----------	------------	-------------------	-----------

### 3. Testing Approach

Table 1.

TEST SUITE <Start Ingestion>		
<b>Description of Test Suite</b>	The following test suite is to evaluate the functionality of the start ingestion process the system is to perform.	
<b>Test Case Identifier</b>	<b>Objective</b>	<b>Criticality</b>
ING 1	Open Event configuration dialog in response to File->Event selection. Save Event configuration (name, description, start, and end times) in response to save button clicked.	Critical
ING 2	Open directory configuration in response to Directory button clicked. Start ingestion process once valid directories (root, red, white, and blue) specified.	Critical
ING 3	Create copies of root directory files. Initiate cleansing operation on root directory files.	Critical
ING 4	Initiate validating operation on cleansed root directory files. Generate enforcement action reports for invalid (non-ingested) files.	Critical
ING 5	Initiate ingestion operation on validated root directory files. Populate log entry table with ingested parsed entries. (if log entries have made it to splunk)	Critical

Table 2.

TEST SUITE <Graphing>		
<b>Description of Test Suite</b>	The following test suite is to evaluate the functionality of the graph editing process.	
<b>Test Case Identifier</b>	<b>Objective</b>	<b>Criticality</b>
GPH 1	Use the GraphEditor to add NodeItems to the GraphEditorView.	Critical
GPH 2	Use the GraphEditor to add RelationshipItem to the GraphEditorView.	Critical
GPH 3	Use the GraphEditor to remove NodeItems from the GraphEditorView.	Critical
GPH 4	Use the GraphEditor to remove RelationshipItems from the GraphEditorView.	Critical
GPH 5	Update the position of RelationshipItems in relation to their parent and child NodeItems.	Critical
GPH 6	Toggle visibility of the elements within a VectorItemGroup.	Critical
GPH 7	Changes made on table views reflects on GraphEditorView.	Critical

Table 3.

<b>TEST SUITE &lt;Table Modifications&gt;</b>		
<b>Description of Test Suite</b>	The following test suite is to evaluate the modifications (add, remove, and editing) on the following tables: Vector, Node, and Relationship.	
<b>Test Case Identifier</b>	<b>Objective</b>	<b>Criticality</b>
TBM 1	Add entries to Vector table.	Critical
TBM 2	Remove entries to Vector table.	Critical
TBM 3	Edit entries on Vector table.	Critical
TBM 4	Add entries to Node table.	Critical
TBM 5	Remove entries from Node table.	Critical
TBM 6	Edit entries in Node table.	Critical
TBM 7	Add entries to Relationship table.	Normal
TBM 8	Remove entries from Relationship tables.	Critical
TBM 9	Edit entries on Relationship tables.	Critical

Table 4.

<b>TEST SUITE &lt;Data Persistence&gt;</b>		
<b>Description of Test Suite</b>	The following test suite is to evaluate the data persistence of the system. This includes (Event, Vector, Log File, Log Entry, Directory and Node, Relationship) configuration's data and the Graph's data.	
<b>Test Case Identifier</b>	<b>Objective</b>	<b>Criticality</b>
DP 1	Event configuration save and load.	Critical
DP 2	Directory configuration save and load.	Critical
DP 3	Log File configuration save and load.	Critical
DP 4	Log Entry configuration save and load.	Critical
DP 5	Vector configuration save and load.	Critical
DP6	Node configuration save and load.	Critical
DP7	Relationship configuration save and load.	Critical



## 4. Tests

### 4.1. Start Ingestion

**Objective:** To establish proper functionality of the start ingestion process.

**Notes:** Access to different test files with various data

Test No.: ING 1		Current Status: Passed		
Test title: Save event details (name, description, start time, end time)				
Testing approach: This test will be conducted on the event configuration dialog, field inputs are selected and then output messages are observed.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Begin test, click “file->Event”	Initial Condition	Event configuration dialog opens	
2	click “save event” button with input fields name and description empty.	Save event with missing fields.	Prompt stating “name or description” input fields empty.	
3	Enter text in name field but leave description field empty. Click “save event” button.	Check with one field (name) empty.	Prompt stating “name or description” input fields empty.	
4	Enter text in description field but leave name field empty. Click “save event” button	Check with one field (description) empty.	Prompt stating “name or description” input fields empty.	
5	Leave default (both start time and end time fields are the same). Click “save event”.	Check to see if time is in valid ranges. (start before end)	Prompt stating “invalid end time”.	
6	Set start time after end time.	Check to see if time is in valid ranges. (start before end)	Prompt stating “invalid end time”.	
7	Set end time before start time.	Check to see if time is in valid ranges. (start before end)	Prompt stating “invalid end time”.	

**Test Plan**

8	All valid fields entered.	Check to see if event with valid fields is saved.	Prompt stating “event saved”	
Concluding Remarks: Tests provided the correct response prompts.				
Testing Team: Keikaku		Date Completed: 04/15/2020		

Test No.: ING 2		Current Status: Pending		
Test title: Save team directories and initiate ingestion process.				
Testing approach: This test will be conducted on the directory configuration dialog, directory paths are selected an output is observed.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Pre-condition (ING 1)  Click on the “Directory” button on the Log File configuration tab.	Initial condition	Directory configuration window is displayed.	
2	Leave “Blue, Red, White, or Root directory” fields empty and click on start data ingestion.	Check with fields empty.	Prompt stating “field is empty”.	
3	Click the “browse” button next to the “Blue, Red, White, or Root” directory fields and make a selection from file picker.	Check to see if browse file picker works.	Respective field is populated with the file source path of the file picked.	
4	Enter Root directory path in field, and enter invalid (not found in Root directory) Red, Blue, or White team directory paths.	Check with invalid (directory not found in Root) directory.	Prompt stating “Directory not found in Root”.	
5	Enter Root directory path in field, and enter valid (found in the Root directory) Red, Blue, or White team directory paths.	Check with valid (directory found in Root) directory.	Prompt stating “Directory structure valid”. File Ingestion process initiated on Root directory.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

**Test Plan**

Test No.: ING 3		Current Status: Pending		
Test title: Initiate cleansing action on files				
Testing approach: This test will be conducted on the cleansing operation, two input files are selected one .csv file and one .log file.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Pre-condition (ING 1-2)  Add the “test_cleansse.log” from the test data directory file to either Red, Blue or White team directories in Pick Data directory.  Note: The cleanse file has empty lines and non-ascii characters planted.	Initial condition	N/A	
2	Click “ingest” button on the Log file configuration tab.	Start ingestion operation.	N/A	
3	Analyze log file table	Check to see if cleansing status is true	Log file details populated on log file table and green check mark under “cleansing status” field.	Displayed correct status.
4	Analyze file contents	Check to see if file has been cleansed.	Files updated and stripped of empty lines/rows, and invalid binary characters.	.log file was stripped of empty lines and non-ascii characters.
Concluding Remarks: Other file formats (.csv) need to be tested.				
Testing Team: Keikaku		Date Completed:		

**Test Plan**

Test No.: ING 4		Current Status: Pending		
Test title: Initiate validation action on files				
Testing approach: This test will be conducted on the validation operation, two input files are selected one .csv file and one .log file.				
STEP	OPERATOR ACTION	PURPOSE	EXEPTED RESULTS	COMMENTS
1	<p>Pre-condition (ING 3)</p> <p>Add the “test_invalid.log” and “test_valid.log” files from the test data directory file to either Red, Blue or White team directories in the Pick Data directory.</p> <p>Note: The valid file has timestamps that are in range based on the event start and end times and contains no missing timestamps.</p> <p>Note: The invalid file has missing timestamps and out of range timestamps.</p>	Initial Condition	N/A	
2	Click “ingest” button on the Log file configuration tab.	Start validation operation.	N/A	
3	Analyze log file table for the valid log file.	Check to see if validation status is True.	Log file details populated on log file table and validation is set to true. (green check mark under “validation status” field.)	
4	Analyze log file table for the invalid log file	Check to see if validation status is False.	Log file details populated on log file table and validation is set to false. (red X mark under “validation status” field.)	
5	Click on invalid log file on table.	Check to see if enforcement action report has been populated on the enforcement action report table.	The enforcement action report populated with errors found on each line (timestamp missing, or timestamp out of range)	
Concluding Remarks:				
TestPlan		Keikaku 企画	Date 4/20/2020	Page 12

**Test Plan**

Other file formats (.csv) need to be tested.	
Testing Team: Keikaku	Date Completed:

Test No.: ING 5		Current Status: Pending		
Test title: : Initiate ingestion action on files				
Testing approach: This test will be conducted on the ingestion operation, two input files are selected one .csv file and one .log file.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	<p>Pre-condition (ING 4)</p> <p>Add the “test_valid.log” file from the test data directory file to either Red, Blue or White team directories in the Pick Data directory.</p> <p>Note: The valid file has timestamps that are in range based on the event start and end times and has no missing timestamps.</p>	Initial condition	N/A	
2	Click “ingest” button on the Log file configuration tab.	Start ingestion operation	N/A	
3	Analyze log file table for the ingested log file.	Check to see if ingested status is True.	Log file details populated on log file table and ingestion is set to true. (green check mark under “ingestion status” field.)	
4	Analyze log entry table for the ingested log entries. (parsed by Splunk)	Check to see if log entry table updated.	Log entry details populated on log entry table.	
Concluding Remarks: Other file formats (.csv) need to be tested.				
Testing Team: Keikaku		Date Completed:		

TestPlan	Keikaku 企画	Date 4/20/2020	Page 13
----------	------------	-------------------	------------

## 4.2. Graphing

**Objective:** To establish proper functionality of the graphing process.

**Notes:** N/A

Test No.: GPH 1 - 2		Current Status: Pending		
Test title: Add items to graph editor view				
Testing approach: This test will be conducted on the GraphEditor control to test its ability to add items to the GraphEditorView				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click “Add Node” button in the Node Configuration Window	Check if new NodeItem appears in the GraphEditorView	The GraphEditorView displays new NodeItem in its respective VectorItemGroup	
2	Click “Add Node” button again for the same node in the Node Configuration Window	Make sure that duplicate NodeItems do not exist	The GraphEditorView remains unchanged. A message box pops up indicating that this NodeItem already exists	
3	Click “Add Relationship” button in the Node Configuration Window	Check if new RelationshipItem appears in the GraphEditorView	The GraphEditorView displays new RelationshipItem in its respective VectorItemGroup	
4	Click “Add Relationship” button again for the same node in the Node Configuration Window	Make sure that duplicate RelationshipItems do not exist	The GraphEditorView remains unchanged. A message box pops up indicating that this RelationshipItem already exists	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

**Test Plan**

Test No.: GPH 3 - 4		Current Status: Pending		
Test title: Remove items from graph editor view				
Testing approach: This test will be conducted on the GraphEditor control to test its ability to remove items from the GraphEditorView				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click “Remove Node” button in the Node Configuration Window	Check if the specified NoteItem is removed in the GraphEditorView	The GraphEditorView no longer displays the specified NodeItem from its respective VectorItemGroup	
2	Click “Remove Node” button again for the same node in the Node Configuration Window	Check that the user is advised that this NodeItem does not exist in the GraphEditorView	The GraphEditorView remains unchanged. A message box pops up indicating that this NodeItem does not exist.	
3	Click “Remove Relationship” button in the Node Configuration Window	Check if the specified RelationshipItem is removed in the GraphEditorView	The GraphEditorView no longer displays the specified RelationshipItem from its respective VectorItemGroup	
4	Click “Remove Relationship” button again for the same node in the Node Configuration Window	Check that the user is advised that this RelationshipItem does not exist in the GraphEditorView	The GraphEditorView remains unchanged. A message box pops up indicating that this RelationshipItem does not exist.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

Test No.: GPH 5		Current Status: Pending		
Test title: Dynamic movement of RelationshipItems and NodeItems				
Testing approach: This test will check the ability to update the position of RelationshipItems in relation to their parent and child NodeItems				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click and drag any NodeItem that is a parent or child of a RelationshipItem in the GraphEditorView	Check if the lines for the RelationshipItems move dynamically with their parent and child nodes	The line coordinates for the RelationshipItem change to match to new center point of the move NodeItem.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

**Test Plan**

Test No.: GPH 6		Current Status: Pending		
Test title: Toggle Vector visibility				
Testing approach: This test will check the ability to toggle visibility of the elements within a VectorItemGroup				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Select a Vector from the Vector Selection drop down menu	Check to make sure that only that vector is visible in the GraphEditorView	Only the NodeItems and their respective RelationshipItems are visible in the GraphEditorView	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		



**Test Plan**

Test No.: GPH 7			Current Status: Pending	
Test title: Synchronization of table views and GraphEditorView				
Testing approach: This test will check that the changes made on table views reflects on GraphEditorView				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Change node name on Node Configuration	Check to make sure that the Node name in the Node Configuration is the same as its respective NodeItem in the GraphEditorView	Displayed Node name on Node Configuration is the same displayed in the NodeItem	
2	Change node description on Node Configuration	Check to make sure that the Node description in the Node Configuration is the same as its respective NodeItem in the GraphEditorView	Displayed Node description on Node Configuration is the same displayed in the NodeItem	
3	Change node description on Node Configuration	Check to make sure that the Node description in the Node Configuration is the same as its respective NodeItem in the GraphEditorView	Displayed Node description on Node Configuration is the same displayed in the NodeItem	
4	Change node log creator on Node Configuration	Check to make sure that the Node log creator in the Node Configuration is the same as its respective NodeItem in the GraphEditorView	Displayed Node log creator on Node Configuration is the same displayed in the NodeItem	
5	Change node event type on Node Configuration	Check to make sure that the Node event type in the Node Configuration is the same as its respective NodeItem in the GraphEditorView	Displayed Node event type on Node Configuration is the same displayed in the NodeItem	
6	Change node description on Node Configuration	Check to make sure that the Node description in the Node Configuration is the same as its respective NodeItem in the GraphEditorView	Displayed Node description on Node Configuration is the same displayed in the NodeItem	
7	Change node icon type on Node Configuration	Check to make sure that the Node icon type in the Node Configuration is the same as its respective NodeItem in the GraphEditorView	Displayed Node icon type on Node Configuration is the same displayed in the NodeItem	
8	Change relationship label on Node Configuration	Check to make sure that the Relationship label in the Node Configuration is the same as its respective RelationshipItem in the GraphEditorView	Displayed Relationship label on Node Configuration is the same displayed in the RelationshipItem	
9	Change relationship parent on Node Configuration	Check to make sure that the Relationship parent in the Node Configuration is the same as its respective RelationshipItem in the GraphEditorView	The coordinates to the RelationshipItem change to the new parent Node specified in Node Configuration	

**Test Plan**

10	Change relationship child on Node Configuration	Check to make sure that the Relationship parent in the Node Configuration is the same as its respective RelationshipItem in the GraphEditorView	The coordinates to the RelationshipItem change to the new child Node specified in Node Configuration	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

### 4.3. Table Modifications

**Objective:** To confirm vector, node, and relationship entries are being added, deleted or modified inside their respective tables.

**Notes:** The vectors node, and relationship entries will have specific id's and will be generated once added.

Test No.: TBM 1 – TBM2		Current Status: Pending		
Test title: Add and delete entries to Vector table				
Testing approach: This test will be conducted in the vector table using an add vector button and a delete vector button.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Begin test, click “add vector” button	Initial condition	Vector with unique id is added to table.	
2	Click “add vector” button again	Intended to test if vector added.	New vector with unique id is added to table.	
3	Click on a single vector row	Intended to test the vector selection.	The vector is darkened to notify user has selected it.	
4	Click “delete vector” button	Intended to test the remove vector from table.	The vector is deleted from the table.	
5	Click on another vector row	Intended to test the select vector.	The vector is darkened to notify user has selected it.	
6	Click “delete vector” button	Intended to test the remove vector from table.	The vector is deleted from the table.	Table is left empty.
7	Click “delete vector” button	Intended to test the remove vector from table.	Nothing happens because table is empty.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

**Test Plan**

Test No.: TBM 3		Current Status: Pending.		
Test title: Edit entries on Vector table				
Testing approach: This test will be conducted on the vector table, field inputs are selected and then input text data is updated.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Begin test, click on existing vector row	Initial Condition	The vector row is darkened to notify the user that the vector is selected.	
2	Double click on the vector name	This is intended to test the Enable editing.	The vector cell allows user to input information.	
3	Press “Enter” key	This is Intended to test the Finish editing.	The vector cell exits editing mode.	
4	Double click on vector description	This is intended to test the Enable editing.	The vector cell allows user to input information.	
5	Press “Enter” key	This is intended to test the Finish editing.	The vector cell exits editing mode.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

## Test Plan

Test No.: TBM 4, 5		Current Status: Pending		
Test title: Node table entry addition and removal				
Testing approach: This intends to test the successful addition and removal of entries to the Node table, with each addition creating a unique Node ID, and each removal disposing of it.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Begin test, click on “Add node”	Intended to test the successful addition of a new Node entry to the Node table.	A new Node entry should be displayed on the table, along with the generated, unique Node ID.	**Addition of Vector must be completed prior to this initial step. Boundary value of 1 tested
2	Click on the “Add node” button $n$ times	Intended to test the successful addition of $n$ entries to the Node table.	$N$ new entries will be appended to the existing entry in the Node table, along with their unique Node IDs.	Testing the boundary value of 2 entries in Node table
3	Click a desired entry’s number on the table itself.	This highlights the selected entry along with all of its properties.	The entry should appear visibly highlighted.	Part of step 4
4	Click on the “Delete node” button with an entry selected (step 3)	This is intended to test the removal of a selected entry in the Node table.	The removal of the selected entry is reflected in the Node table that is displayed.	Removes a single entry from the list of size $n$
5	Holding shift on the keyboard, click the number of $n$ entries on the table itself.	This is intended to test the highlighting and selecting of $n$ entries from the Node table.	The selected entries should appear visibly highlighted.	Part of step 6
6	Click the “Delete node” button with the entries selected (step 5).	This is intended to test the removal of $n$ selected entries.	The removal of the selected entries is reflected in the Node table that is displayed. If all entries are selected, table is blank.	Tests the boundary value of 0
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

Test No.: TBM 6		Current Status: Pending		
Test title: Editing of entry in Node table				
Testing approach: This is intended to test the editing of the various cells of a given entry in a Node table, which include Node Name, Timestamp, Description, Log Entry Reference, Log Creator, Event Type, Icon Type, Source, and Node Visibility.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Begin test, click on “Add node” $n$ times to generate multiple entries.	This is intended to provide several entries for the editing of their various fields.	$N$ log entries will be displayed in the Node table.	Testing for addition and removal is a different test case.

TestPlan	Keikaku 企画	Date 4/20/2020	Page 21
----------	------------	-------------------	------------

## Test Plan

2	For any desired entry in the Node table, double-click on the cell in the column labeled “Node Name.”	This is intended to test the ability to select the “Node Name” as an editable field.	The Node entry’s cell under the column “Node Name” displays a cursor and is ready to take user input.	
3	Type the desired name for the Node entry.	This is intended to demonstrate the ability to edit the field “Node Name.”	The cell under the column “Node Name” should reflect the user’s keystrokes.	
4	Click outside of “Node Name” cell or press Tab.	This is intended to finalize the changes to the “Node Name” cell provided by the user.	The cell under the column labeled “Node Name” now reflects the user input.	
5	For any desired entry in the Node table, double-click on the cell in the column labeled “Node Timestamp.”	This is intended to test the ability to select the “Node Timestamp” as an editable field.	The Node entry’s cell under the column “Node Timestamp” displays a cursor and is ready to take user input.	
6	Type the desired timestamp for the Node entry.	This is intended to demonstrate the ability to edit the field “Node Timestamp.”	The cell under the column “Node Timestamp” should reflect the user’s keystrokes.	
7	Click outside of “Node Timestamp” cell or press Tab.	This is intended to finalize the changes to the “Node Timestamp” cell provided by the user.	The cell under the column labeled “Node Timestamp” now reflects the user input.	
8	For any desired entry in the Node table, double-click on the cell in the column labeled “Node Description.”	This is intended to test the ability to select the “Node Description” as an editable field.	The Node entry’s cell under the column “Node Description” displays a cursor and is ready to take user input.	
9	Type the desired description for the Node entry.	This is intended to demonstrate the ability to edit the field “Node Description.”	The cell under the column “Node Description” should reflect the user’s keystrokes.	
10	Click outside of “Node Description” cell or press Tab.	This is intended to finalize the changes to the “Node Description” cell provided by the user.	The cell under the column labeled “Node Description” now reflects the user input.	
11	Repeat steps 2-10 for “Log Entry Reference, Log Creator, Event Type, Icon Type, and Source”	This is intended to test the edit ability of the cells under the columns labeled “Log Entry Reference, Log Creator, Event Type, Icon Type and Source” on the Node table.	Any user provided input for the cells under the columns labeled “Log Entry Reference, Log Creator, Event Type, Icon Type, and Source” should be reflected on the displayed table.	

**Test Plan**

12	For any desired entry in the Node table, click on the “Node Visibility” check box.	This is intended to test the toggling on/off of the Node’s visibility in the graph from the Node Table.	The check box next to the selected entry in the Node table should appear checked if unchecked and vice versa.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

Test No.: TBM 7 – TBM 8			Current Status: Pending	
Test title: Add and remove entries to the relationship table				
Testing approach: This test will be conducted on the relationship configuration dialogue, using add relationship button and delete relationship button.				
STEP	OPERATOR ACTION	PURPOSE	EXEPECTED RESULTS	COMMENTS
1	Begin test, click “add relationship” button	Initial condition	Relationship with unique id is added to the table.	Parent and child columns remain empty
2	Click “add relationship” button again	This is intended to test the add another relationship	New relationship with unique id is added to table.	
3	Click on single relationship row	This is intended to select relationship	The relationship is darkened to notify the user the relationship is selected.	
4	Click “delete relationship” button	This is intended to test the remove relationship from table	The relationship is deleted from the table	
5	Click another relationship row	This is intended to select relationship	The relationship is darkened to notify the user the relationship is selected.	
6	Click “delete relationship” button	This is intended to remove relationship from table	The relationship is deleted from the table	Table is left empty.
	Click “delete relationship” button	This is intended to remove relationship from table	Nothing happens because table is empty	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

**Test Plan**

Test No.: TBM 9		Current Status: Pending		
Test title: Edit entries on the relationship tables				
Testing approach: This test will be conducted on the relationship configuration dialogue, field inputs are selected and the input text data is saved.				
STEP	OPERATOR ACTION	PURPOSE	EXEPECTED RESULTS	COMMENTS
1	Begin test, click on existing relationship row	Initial condition	The relationship row is darkened to notify the user that the relationship is selected.	
2	Double click on the “Parent” cell	This is intended to enable editing.	The relationship parent cell allows user to input information	
3	Press “Enter” key	This is intended to finish editing.	The relationship cell exits editing mode	
4	Double click “Child” cell	This is intended to enable editing.	The relationship child cell allows user to input information	
5	Press “Enter” key	This in intended to finish editing.	The relationship cell exits editing mode	This will cause a relationship between current vector parent and its child (if the child and parent ids are valid)
6	Double click “Label” cell	This is intended to enable editing.	The relationship label cell allows user to input information	
7	Press “Enter” key	This is intended to finish editing.	The relationship cell exits editing mode	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		



## 4.4 Data Persistence

**Objective:** To ensure data is persistent throughout the application's lifetime.

**Notes:** The storage is on a file system basis, where data is being serialized and stored to a file then retrieved once needed.

Test No.: DP1		Current Status: Pending		
Test title: Test the event configuration’s data is being saved.				
Testing approach: The event configuration window is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Select File->Event	Initial condition	Event configuration displayed.	
2	Populate valid fields (refer to ING 1) click “save event”	Check with data populated.	Event saved.	
3	Close window.	Exit application.	Window is closed.	
4	Launch application again. Observe the event configuration data.	Check to see if data was re-populated.	Event configuration has been repopulated with saved data.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

**Test Plan**

Test No.: DP2		Current Status: Pending		
Test title: Test the directory configuration’s data is being saved.				
Testing approach: The directory configuration window is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click “directory” button on log file configuration tab.	Initial condition	Directory configuration displayed.	
2	Populate valid fields (refer to ING 2) click “start Ingestion” wait till operation is complete.	Check with data populated.	directories saved.	
3	Close window.	Exit application.	Window is closed.	
4	Launch application again. Observe the directory configuration data.	Check to see if data was re-populated.	Directory configuration has been repopulated with saved data.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

Test No.: DP3		Current Status: Pending		
Test title: Test the Log file tabular data is being saved.				
Testing approach: The log file configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Pre-condition (log file has been added to root directory and event information has been entered) Wait until operation complete.  Click “ingest” button Wait until operation complete.	Initial condition	Ingest process initiated. Log file table populated with data.	
2	Close window.	Exit application.	Window is closed.	
3	Launch application again. Observe the log file configuration data.	Check to see if data was re-populated.	Log file configuration has been repopulated with saved data.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

TestPlan	Keikaku 企画	Date 4/20/2020	Page 26
----------	------------	-------------------	------------

**Test Plan**

Test No.: DP4		Current Status: Pending		
Test title: Test the Log entry tabular data is being saved.				
Testing approach: The log entry configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Pre-condition (valid log file has been added to root directory, and event and directory information has been entered)  Click “ingest” button Wait until operation complete.	Initial condition	Ingest process initiated. Log entry table populated with data once complete.	
2	Close window.	Exit application.	Window is closed.	
3	Launch application again. Observe the log entry configuration data.	Check to see if data was re-populated.	Log entry configuration has been repopulated with saved data.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

Test No.: DP5		Current Status: Pending		
TestPlan	Keikaku 企画	Date 4/20/2020	Page 27	

**Test Plan**

Test title: Test the vector tabular data is being saved.				
Testing approach: The vector configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click the “Vector” button .	Initial condition	Vector configuration table displayed.	
3	Add 20 vectors to Vector table. Click “add vector” button 20 times.	Add vectors to table.	Vector configuration has 20 entries.	
4	Modify random vectors with names and descriptions. (refer to TBM 1-2)	Populate table with random data.	Vector fields updated.	
5	Close window.	Exit application.	Window is closed.	
6	Launch application again. Observe vector table.	Check to see if data was re-populated.	Vector table is repopulated with saved data.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

Test No.: DP6		Current Status: Pending		
TestPlan	Keikaku 企画	Date 4/20/2020	Page 28	

**Test Plan**

Test title: Test the Node tabular data is being saved.				
Testing approach: The node configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click the “Node configuration” tab .	Initial condition	Node configuration view displayed.	
3	Add 20 nodes to Node table. Click “add node” button 20 times.	Populate table with nodes.	Node configuration has 20 entries.	
4	Modify random nodes with names, descriptions, timestamps etc. (refer to TBM 6)	Populate table with random data.	node fields updated.	
5	Close window.	Exit application.	Window is closed.	
6	Launch application again. Observe node table.	Check to see if data was re-populated.	Node table is repopulated with saved data.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

Test No.: DP7		Current Status: Pending		
TestPlan	Keikaku 企画	Date 4/20/2020	Page 29	

**Test Plan**

Test title: Test the Relationship tabular data is being saved.				
Testing approach: The relationship configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click the “Node configuration” tab.	Initial condition	Node configuration view displayed.	
2	Click the “relationship” button	View relationship configuration table.	Relationship configuration table is displayed.	
3	Add 20 relationships to relationship table. Click “add relationship” button 20 times.	Populate table with relationships.	Relationship configuration has 20 entries.	
4	Modify random relationships with parents, children, and labels. (refer to TBM 9)	Populate table with random data.	relationship fields updated.	
5	Close window.	Exit application.	Window is closed.	
6	Launch application again. Observe relationship table.	Check to see if data was re-populated.	Relationship table is repopulated with saved data.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed:		

## 5. Test Schedule

Task and date	People	Description
04/30/19	David Rayner	Log ingestion test suite (test cases ING 1-5)
04/30/19	Valentin Becerra	Graph test suite (GPH 1-4)
05/01/19	Valentin Becerra	Graph test suite (GPH 5-7)
05/01/19	Jorge Garcia	Table Modifications (TBM 1-4)
05/01/19	Angel Villapando	Table Modifications (TBM 4-8)
05/02/19	Anthony Desarmier	Data Persistence (DP 1-4)
05/02/19	Mario Delgado	Data Persistence (DP 5-7)

## 6. Other Sections

Tests are to be portioned off to each member of the development team based on the four test suites in section 3. The suites capture the main components of the system. Each suite will have a lead team member, this is to ensure that the tests are being performed and yield appropriate results.

No training is required, however the development team is to have an understanding of the system components and the type of testing that needs to be performed.

The program that needs to be installed is python 3, preferably version 3.8. This application supports Linux, Windows 10, and Mac OS. The application is tailored for offline usage, no dependencies on web.

The following is a list of the current required installations:

- PyQt5==5.14.2
- PyQt5-sip==12.7.2
- python-dateutil==2.8.1
- python-dotenv==0.12.0
- virtualenv==20.0.17
- virtualenv-clone==0.5.4
- splunk-sdk==1.6.12

Splunk Enterprise is required in order to run the Splunk server. This requires a Splunk Enterprise download on the respective OS. The host's username, password port of the Splunk server, and index name to store the entries is required and to be stored in the applications splunk\_manage.conf.

Storage is to managed through serialization and saved on the basis of a file system. Therefore, currently there is no need for installation of a database.

No cost is to be associated with any of the software, since each of the libraries are open-source.

TestPlan	Keikaku 企画	Date 4/20/2020	Page 32
----------	------------	-------------------	------------



## 7. Appendix

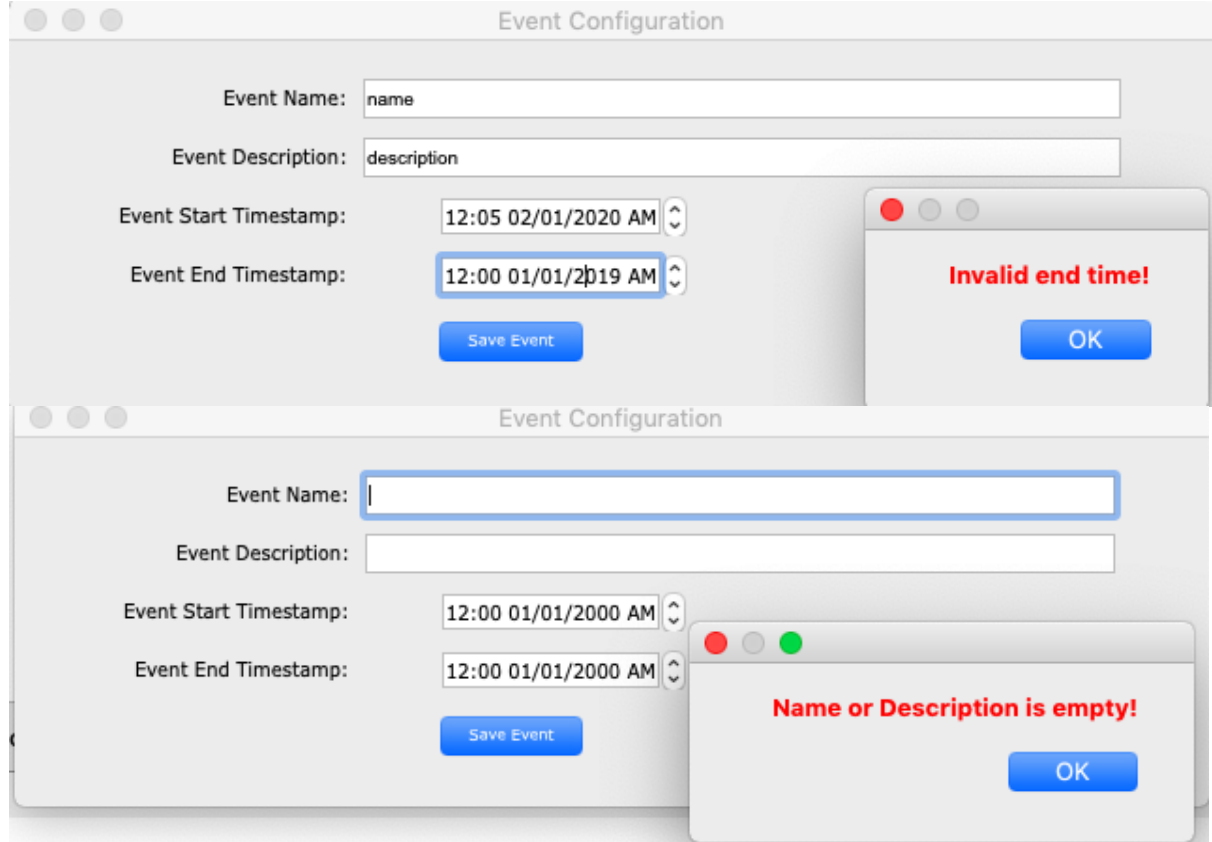


Figure ING 1.

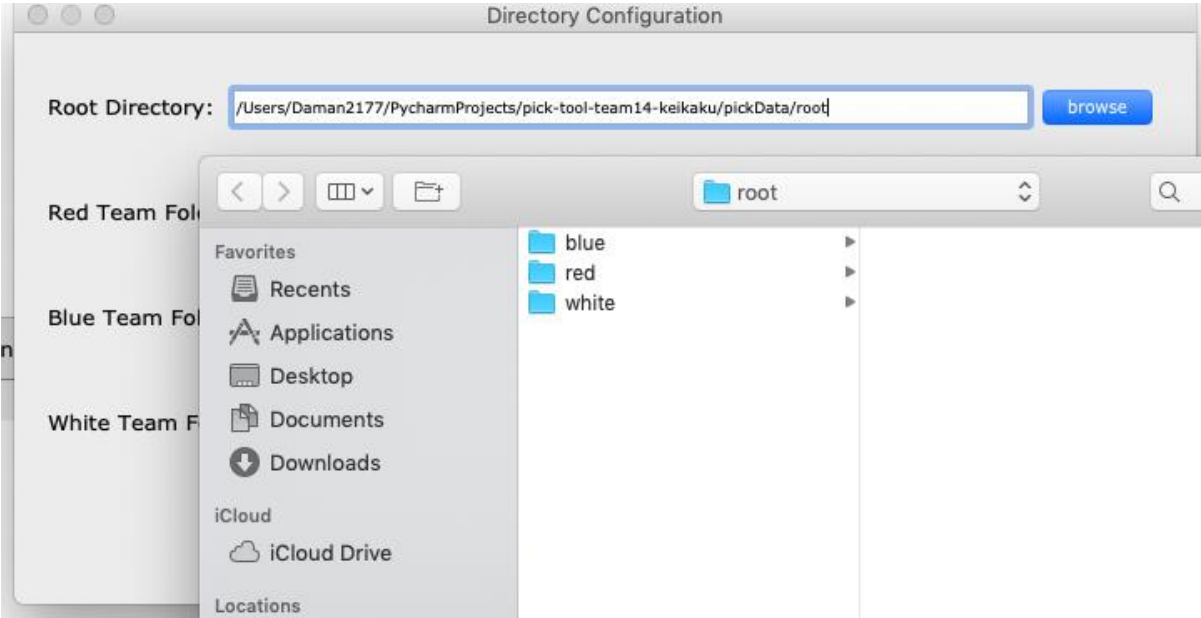


Figure ING 2.

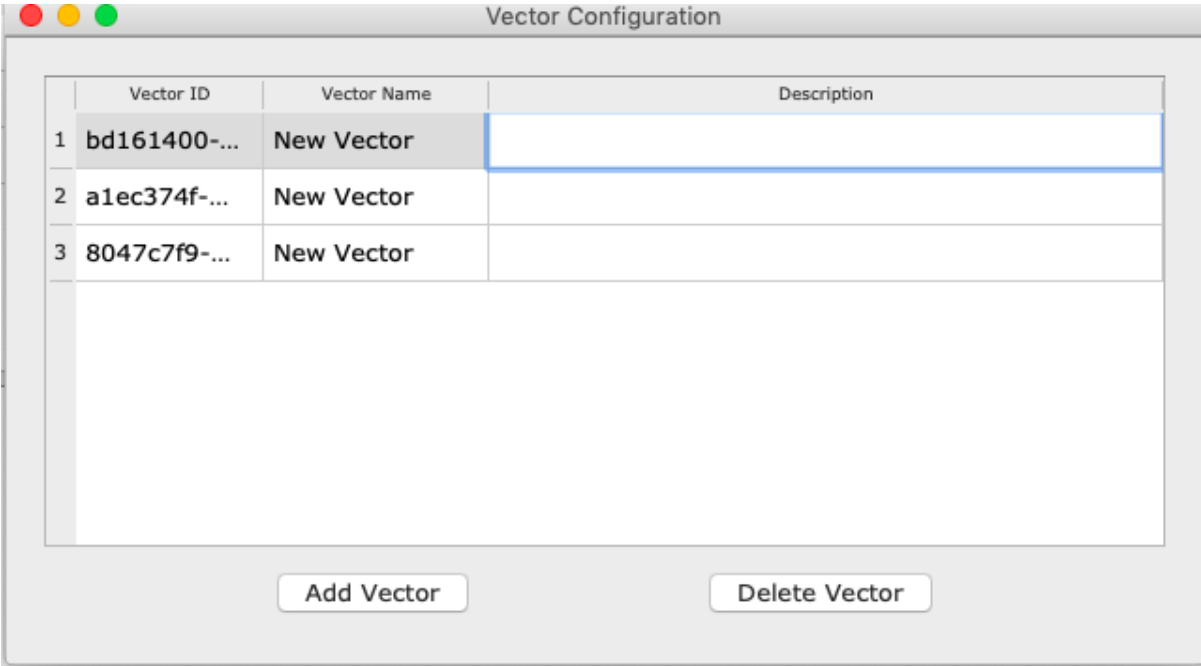


Figure TBM 3.

## Test Plan

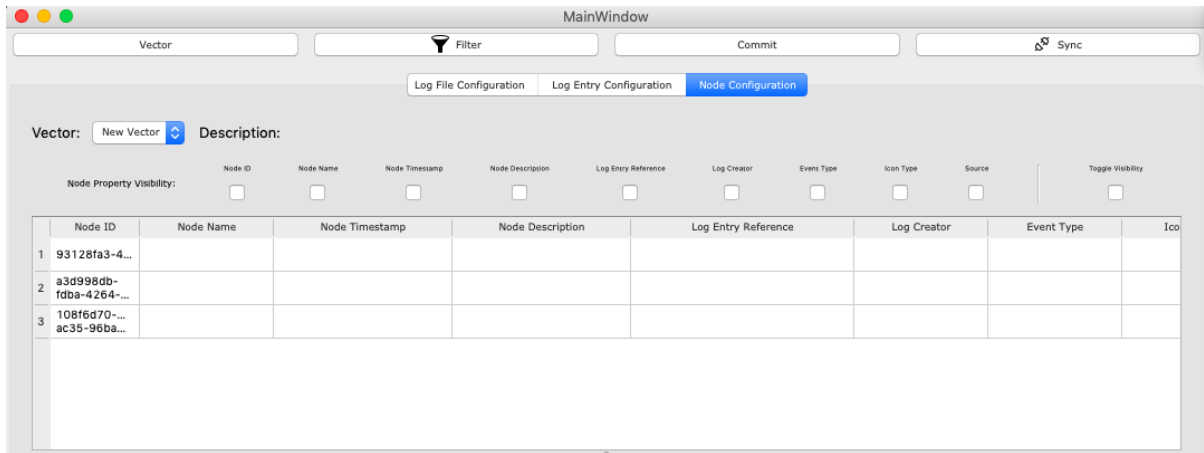


Figure TBM 4.