

**Vulnerability Tool Requirement**

Copyright © 2017 Risk Management Solutions, Inc. All Rights Reserved

Risk Management Solutions, Inc. CONFIDENTIAL INFORMATION

This document contains proprietary information of Risk Management Solutions, Inc. This information is highly confidential. Use, duplication and disclosure of the information contained in this document are restricted by Nondisclosure agreement. The information contained in this document may not be reproduced or shared with any third parties without the explicit prior written consent of Risk Management Solutions, Inc.

Risk Management Solutions, Inc.   
7575 Gateway Blvd.  
Newark, CA 94560  
Phone: 510-505-2500  
Fax: 510-505-2501

Risk Management Solutions, Inc. (RMS) presents the information in this document for internal review by its customers. No part of this document may be disclosed or reproduced in any form without prior consent from Risk Management Solutions, Inc.

Test Plan Revision Summary & Reference

Department: Model Certification

Author: Mohammad Razavi

Duration: June 2016– TDB

Responsible Parties:

|  |  |  |
| --- | --- | --- |
| Activity | Person(s) involved | Status |
| Write-up | Mohammad Razavi | Ongoing |
| Review | Ching-Yee Chang , Nervdeshwar Pandey, Srinivas Thupakula | TDB |

Revision History:

|  |  |  |
| --- | --- | --- |
| **Testplan Version** | **Date** | **Summary of Changes** |
| 0.1 | 06/23/2016 | Initial version |
| 0.2 | 07/06/2016 | Tests 9, and 10 are added |
| 0.3 | 07/12/2013 | Tests 0, 11, and 12 are added |
| 0.4 | 07/18/2013 | Test 13 added. Formats are corrected. Checks for Databases including Multiple Perils and Multiple Countries |

◙

Contents

[Introduction 4](#_Toc456620101)

[Stored data 4](#_Toc456620102)

[Test cases 5](#_Toc456620103)

[Test-0: database information 5](#_Toc456620104)

[Test-1: tables availability 5](#_Toc456620105)

[Test-2: empty tables 5](#_Toc456620106)

[Test-3: risklink tables without index 5](#_Toc456620107)

[Test-4: Comparison of Number of Rows in corresponding tables 5](#_Toc456620108)

[Test-5:Test vgeo and eqvgeo tables 5](#_Toc456620109)

[Test-6: Test cghs and eqcghsus tables 5](#_Toc456620110)

[Test-7: Test Imap and eqimapus tables 5](#_Toc456620111)

[Test-8: Test vdc0 table 5](#_Toc456620112)

[Test-10: Test vcc and eqvccus tables 6](#_Toc456620113)

[Test-11: Test vocc and eqvoccus tables 6](#_Toc456620114)

[Test-12: test existence of pdc\_num in vinv if the same modif\_pdc exists in imap for a given inv\_recnum 6](#_Toc456620115)

[Test-13: test for case sensitivity in Map\_InvOcc and Map\_InvKey tables 6](#_Toc456620116)

[Tool requirements 6](#_Toc456620117)

[Download and install R 6](#_Toc456620118)

[User inputs for validate\_VulnCache.bat 6](#_Toc456620119)

[Vulnerability database should be attached to a SQL server. 6](#_Toc456620120)

# Introduction

The objective of the vulnerability tool is to test the integrity of the vulnerability database and to verify that the data is properly transferred between RiskLink tables and Next Generation (NG) tables.

# Stored data

Three sets of tables are used in this tool which include RiskLink tables, NG tables, and Mapping tables. Table 1 shows the list of tables that are necessary for the vulnerability tool.

**Table-1: List of Required Tables for Utilizing Vulnerability Tool**

|  |  |  |
| --- | --- | --- |
| **RiskLink Tables** | **NG Tables** | **Mapping Tables** |
| ppvllcc | vll | Map\_HazardType |
| ppcghscc | cghs | Map\_HazardSCALE |
| ppfliinvcc | fliinv | Map\_COVERAGE |
| ppimapcc | imap | Map\_MAPCCLSSIF |
| ppLandMDRcc | LandMDR | Map\_MAPCCTier1 |
| ppLiqMDRcc | LiqMDR | Map\_MAPCCTier2 |
| ppmdsccc | mdsc | Map\_MAPCCTier3 |
| ppodsccc | odsc | Map\_InvOcc |
| ppvbicc | vbi | Map\_CvKey |
| ppvbimodcc | vbimod | Map\_BRTVKey |
| ppvbrdrcc | vbrdr | Map\_BRDRKey |
| ppvbrtvcc | vbrtv |  |
| ppvcccc | vcc |  |
| ppvcvcc | vcv |  |
| ppvdc0cc | vdc0 |  |
| ppvdscc | vds |  |
| ppvgeocc | vgeo |  |
| ppvhsrcc | vhsr |  |
| ppvinvcc | vinv |  |
| ppvinvaghazcc | vinvaghaz |  |
| ppvlifcc | vlif |  |
| ppvlifelinebicc | vlifelinebi |  |
| ppvlifelinebidccc | vlifelinebidc |  |
| ppvocccc | vocc |  |
| ppvparamcc | vparam |  |
| ppvrscc | vrs |  |
| ppvvrggeocc | vvrggeo |  |

# Test cases

## Test-0: database information

Check if database includes both RiskLink and NG Tables or only includes RiskLink Tables

## Test-1: tables availability

Tests if all tables listed in Table-1 of this document are included in the input database.

## Test-2: empty tables

Reports empty tables.

## Test-3: risklink tables without index

Reports RiskLink Tables without Index

## Test-4: Comparison of Number of Rows in corresponding tables

Compare the number of rows in each RislLink Table with its corresponding table in the NG and report any dissimilarity.

## Test-5:Test vgeo and ppvgeo tables

1. Checks and reports duplicate postal codes in vgeo table
2. Checks and reports duplicate district numbers
3. Checks for the number of distinct states in vgeo table
4. Check if records with the same GEO\_RECNUM have the same values in vgeo and ppvgeocc tables\
5. Check for the wild card at the end

## Test-6: Test cghs and ppcghscc tables

1. Checks if the records with same cghs\_ID have ppuivalent Hazard\_Type, Hazard\_Scale, and Coverage. It uses the mapping tables to find the equivalent values. It reports any dissimilarity between two tables.
2. Check if all CVG\_GRADEs include only 0 or all values between 0 and 4

## Test-7: Test Imap and ppimapcc tables

Compares equivalent records between imap and ppimapcc tables and reports dissimilarities. It uses INV\_Recnum as the key for comparison.

**Test-8: Check VINV and PPVINVCC Tables**

Compares equivalent records between vinv and ppvinvcc tables and reports dissimilarities. It uses INV\_RECNUM, PDC\_NUM, and PDC\_GROUP as keys for comparison.

## Test-9: Test vdc0 table

1. Check if all damage curves are monotonically increasing
2. Check if the x-axis on all damage curves are monotonically increasing
3. Check if curves with the same CGHS\_ID have the same number or records

## Test-10: Test vcc and ppvcccc tables

Compares ppuivalent records between vcc and ppvcccc tables and reports dissimilarities.

## Test-11: Test vocc and ppvocccc tables

Compares equivalent records between vocc and ppvocccc tables and reports dissimilarities.

## Test-12: test existence of pdc\_num in vinv if the same modif\_pdc exists in imap for a given inv\_recnum

Check existence of PDC\_NUM in vinv if the same MODIF\_PDC exists in imap for a given INV\_RECNUM

## Test-13: test for case sensitivity in Map\_InvOcc and Map\_InvKey tables

Tests for repeated keys in Map\_InvOcc and Map\_InvKey tables because the values are written in different cases. As an example “unknown” and “UnKnown.”

## Test-14: consistency of data between vcc and imap tables

## Test-15: consistency of data between vocc and imap tables

## Test-16: consistency of data between vgeo and imap tables

# Tool requirements

## Download and install R

## User inputs for validate\_VulnCache.bat

## Vulnerability database should be attached to a SQL server.

## NGGeography database should be attached to a SQL server.

## NAEQ\_PDC\_MAPPING database should be attached to a SQL server.

# Feature requests

1. List of state in each country (Done)
   1. Pair against NGGeography (a1geolookup for states, fieldlookup is very helpful)
2. Where country=xx for ALL queries (Done)
3. IMAP can contain records not in VCC or VOCC. Because of IFM and BR. IMAP can have these, but VCC/VOCC may not have those.
4. Reformat VGEO test output. Include sub sections for postcodes, admin codes etc. (Done)
5. VGEO: Check for wild card at the end (Done)
6. CGHS: coverage grade should either have 0 or 0-4 (Done)
7. Verify modif\_pdc in modifier binary files (this needs reading modifier binary files. We already have the code for that. Ref: VeenaK
8. Extend some tests to RL only (vdc etc.) (vdc0 test added)
9. Review the tool with Pasan/Agustin
10. Test-17: Integrity of IMAP with VCC, VOCC, VGEO, using PDC list received from modelers. (Proudly Done! Finally!!!!)
11. Test-18: CV key in IMAP should be in vcv table (DONE!)
12. Include case sensitivity for all KEYS
13. Test-19: Expand tests to fliinv tables. Keys in there too. (DONE!)
14. Test-20: Expand to MDSC\_ODSC tables. All modifiers should have options!
15. Test-21: DC\_Key between VHAZ and VGEO tables (DONE!)
16. Test-22: BI related tables:
    1. Pick up BI key from VGEO, flow it through all BI tables it shows up in!
    2. Note that this test will be different by model. New BI model will have a different set of BI tables than old models. This information should be coded and brought in from config file. (DONE!)
17. Test-23: Check for BI Zones between VGEO/VBI and Geohazard database’s Admin tables
18. Test-24. For every record in IMAP, there should be one record in VRS (DONE!)
19. TEST-25: Test in VDC if dam > 100%. Also test if the maximum damage of any given curve is very small ( < 1 or so) (DONE!)

Test on a Hurricane tables. POC: Peter Datin

1. Test for NULL values: There should be no null values in
   1. Any field of ppVOCCcc
   2. ppVCCcc except for RISK\_TYPE, MAPCCTIER2, MAPCCTIER3
   3. VGEO for DC\_KEY, INV\_KEY, BI\_KEY, BR\_KEY
   4. Any field of VINV
2. Wild card in IMAP and VGEO should always be the last record
3. Include a check for schemas. This should be done by creating an ‘expected schema’ list for each table for each peril model. To begin with, this can be confined to: VCC, VOCC, IMAP, VINV and VGEO
4. Compare Vulnerability with a baseline database
5. Alternative damage curves