

**Mobilink – OSS Expansion project**

**User Acceptance Testing**



**Innovise ESM**

**Keypoint House**

**High Street**

**Slough**

**SL1 1DY**

**Tel: +44 (0) 1753 513 902**

**Authors: Prev Datta**

**Title: Mobilink - UAT**

**Version: 2.3**

**Date: 6th August 2010**

Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reason** | **Author** |
| 1.1 | 06/08/2010 | Created | Prev Datta |
| 1.2 | 13/08/2010 | Updated | Prev Datta |
| 1.3 | 17/08/2010 | SME’s Add Cases | Prev Datta |
| 1.6 | 20/08/2010 | SME’s Add Cases | Prev Datta |
| 1.8 | 24/08/2010 | Added to document | Prev Datta |
| 2.0 | 31/08/2010 | Added TSRM Cluster Tests | Prev Datta |
| 2.1 | 31/08/2010 | Added New Test Summary | Prev Datta |
| 2.1 | 31/08/2010 | Added New Test Summary | Prev Datta |
| 2.2 | 01/09/2010 | Added New Test Summary | Prev Datta |
| 2.3 | 01/09/2010 | Cleaned/Add Missing TC’s | Prev Datta |
| 2.4 | 02/09/2010 | Updates Omnibus Test Cases | Fakhra Fiaz |
| 2.5 | 02/09/2010 | Add & Update Impact Test Cases | Chris Janes |

Table of Contents

[1.1. Objectives 5](#_Toc271136830)

[1.2. Definitions 5](#_Toc271136831)

[1.3. Acceptance Testing 7](#_Toc271136832)

[1.4. Configuration Validation 7](#_Toc271136833)

[1.5. Sample Test Scenarios 9](#_Toc271136834)

[1.6. Defect Tracking Process 10](#_Toc271136835)

[1.7. Problem Priorities 10](#_Toc271136836)

[1.8. TSRM High Availability 12](#_Toc271136837)

[1.9. Omnibus Environment 13](#_Toc271136838)

[1.10. Netcool Reporter Environment 21](#_Toc271136839)

[1.11. ITNM IP Environment 22](#_Toc271136840)

[1.12. Netcool/Impact Environment 23](#_Toc271136841)

[1.13. TBSM Environment 24](#_Toc271136842)

[1.14. TSRM Environment 24](#_Toc271136843)

[5.1 Omnibus 26](#_Toc271136844)

[Test Number: 5.1.1 26](#_Toc271136845)

[Test Number: 5.1.2 27](#_Toc271136846)

[Test Number: 5.1.2 27](#_Toc271136847)

[Test Number: 5.1.3 28](#_Toc271136848)

[Test Number: 5.1.4 28](#_Toc271136849)

[Test Number: 5.1.5 29](#_Toc271136850)

[Test Number: 5.1.6 29](#_Toc271136851)

[Test Number: 5.1.7 30](#_Toc271136852)

[Test Number: 5.1.8 30](#_Toc271136853)

[Test Number: 5.1.9 31](#_Toc271136854)

[Test Number: 5.2.0 31](#_Toc271136855)

[Test Number: 5.2.1 32](#_Toc271136856)

[Test Number: 5.2.2 32](#_Toc271136857)

[Test Number: 5.2.3 33](#_Toc271136858)

[Test Number: 5.2.4 33](#_Toc271136859)

[Test Number: 5.2.5 34](#_Toc271136860)

[Test Number: 5.2.6 35](#_Toc271136861)

[Test Number: 5.2.7 35](#_Toc271136862)

[Test Number: 5.2.8 36](#_Toc271136863)

[Test Number: 5.2.9 37](#_Toc271136864)

[Test Number: 5.3.0 37](#_Toc271136865)

[Test Number: 5.3.1 37](#_Toc271136866)

[Test Number: 5.3.2 38](#_Toc271136867)

[Test Number: 5.3.4 38](#_Toc271136868)

[Test Number: 5.3.5 39](#_Toc271136869)

[Test Number: 5.3.6 39](#_Toc271136870)

[Test Number: 5.3.6 40](#_Toc271136871)

[Test Number: 5.3.7 40](#_Toc271136872)

[Test Number: 5.3.8 41](#_Toc271136873)

[Test Number: 5.3.9 42](#_Toc271136874)

[Test Number: 5.4.0 42](#_Toc271136875)

[Test Number: 5.4.1 43](#_Toc271136876)

[6.0 ITM 43](#_Toc271136877)

[Test Number: 6.0.1 43](#_Toc271136878)

[Test Number: 6.0.2 44](#_Toc271136879)

[Test Number: 6.0.3 44](#_Toc271136880)

[Test Number: 6.0.4 45](#_Toc271136881)

[Test Number: 6.0.4 45](#_Toc271136882)

[Test Number: 6.0.5 45](#_Toc271136883)

[Test Number: 6.0.8 46](#_Toc271136884)

[Test Number: 6.0.9 46](#_Toc271136885)

[7.0 Impact 47](#_Toc271136886)

[Test Number:7.0.1 47](#_Toc271136887)

[Test Number: 7.0.2 48](#_Toc271136888)

[Test Number: 7.0.3 49](#_Toc271136889)

[Test Number: 7.0.4 50](#_Toc271136890)

[7.1 Cell Down 51](#_Toc271136891)

[Test Number: 7.1.0 51](#_Toc271136892)

[Test Number: 7.1.1 53](#_Toc271136893)

[Test Number: 7.1.2 54](#_Toc271136894)

[8.0 TSRM 54](#_Toc271136895)

[Test Number: 8.0.1 54](#_Toc271136896)

Introduction

## Objectives

The purpose of this document is to test the functionality of the components that make up the Mobilink OSS solution encompassing IBM Tivoli Service Request Manager, OMNIbus Probes, IMPACT, Tivoli Monitoring, Network Manager and Business Service Manager. The objective of the acceptance process is to ensure the product and processes meet Mobilink’s expectations and documented requirements.

The intent of this document is to outline an acceptance process for Mobilink. It includes

* Description of the testing processes
* Describes a test case and defect tracking
* Details the acceptance criteria and test cases

## Definitions

* ***Problem/Defect –*** a system response that does not meet the predicted test result and/or design, and therefore was unexpected. Problems require a change to the system (i.e., screen, table, report, etc.) and are specific to the application (i.e., TSRM).
* ***Issue -*** is an unresolved problem that needs to be identified, prioritized and the impact determined.
* ***Tested/Validated*** - functionality that has successfully passed a minimum of one round of system testing and acceptance testing.
* ***System Test*** - test performed by the technical team to verify the code/module was built as designed.
* ***Acceptance Testing -*** test performed by the business team to ensure the built system (and in turn, design) meets the business needs.
* ***Configuration Validation –*** series of tests and checks to validate the data loads, reports, and screen configurations prior to moving into Production.
* ***End-to-end Testing*** – Acceptance testing using multiple scenarios to ensure the system completes the overall business process.

Acceptance Process

## Acceptance Testing

The Mobilink OSS project is essentially a package implementation (i.e. TSRM/IMPACT) configured to meet Mobilink’s business needs. As such the testing process used will be somewhat simpler in nature than that used in typical development projects. Testing will primarily focus on verifying that the IBM solution meets the business requirements, i.e. acceptance testing. To verify this, the users will follow a number of scenarios which covers how the daily business will be conducted using the business processes for event and operational areas. These scenarios will then be executed using the base line configuration of TSRM and all defects/problems reported and tracked. To be effective, the testing must be structured in that all scenarios must be documented in detail so that they are repeatable, the execution results are tracked, and all defects/problems are recorded and reviewed.

Once scenario testing is completed, End-to-End (EtoE) tests will be run. The EtoE tests are essentially combining the various specific scenarios together so that a complete day in the life is simulated.

## Configuration Validation

The complete configuration (screens, value lists, tables, reports) needs to be verified prior to signing off to move to staging (i.e., code freeze) and again before moving to production. To ensure this review is thorough, each major component of the configuration will be assigned to Mobilink to review. They will test the following:

**Tables**

* Sample approx. 10% of data (i.e. 15-25 for 200 record table) and include the first and last record
* Compare the samples to the source data.
* Ensure the data is the same as the source, accurate, and complete

**Value Lists**

* Ensure all values in the list are accurate and complete

**Screens**

* Ensure the screen is useable and functions as expected, specifically:
* All the field lengths are correct and display the complete value
* Tab order is correct
* All the fields are available

**Work Flows**

* Ensure your test email is a recipient of the Work Flow.
* Follow the steps described in the Test Case
* Check to see if you have received the notification via email and the contents is correct.

**Reports**

* Ensure all the reports are readable and accurate, specifically:
* Data is correct
* All rows and columns are aligned on the report

Test Case Documentation and Tracking

The test cases are the basis of the testing exercise and should be devised to verify the application implementation & configuration supports the business processes. Tests will be designed with a predetermined objective (i.e., acceptance criteria) of testing the functionality of a specific component of the process. The design of the test cases must reflect a scientific method (i.e. tests must have an objective, prediction of outcome, controlled environment, and repeatable results).

*Example 1:*

***Acceptance Criteria***

*Mobilink require manual creation of Work Orders in some cases for vendor or internal operations.*

***Predicted Result***

*The User is able to manually create Work Orders from the WO module and assign it to a team or group of people and associate related Incident Records to it.*

***Test Description***

*1. Log on*

*2. Go to the WO module and select create NEW*

*3. Fill in the Work Order, Name, Description etc*

*4. Go to Related Records TAB and Select NEW under the Related Tickets section*

*5. Select the arrow up and browse for the Ticket you wish to associate.*

*6. Select the Ticket and SAVE*

## Sample Test Scenarios

|  |  |  |
| --- | --- | --- |
| **Test Case ID:** AW-01 | **Test Type:** Customer Service Desk, Functional Test | |
| **Test Name:** Standard work request. | | |
| **Description:** Create a standard work request. | | |
| **Data Setup:** Cost centers, GL, Active Org and Site, User needs to be assigned a default site. | | |
| **Steps to Execute:**   1. Enter request for work via web, the request should include; location, description, contact, date/time, cost center and a customer requested priority. 2. Once the above information has been correctly entered click “OK” to submit the work request to the Customer Service Desk 3. Log in to Maximo with the generic Work Controller account and select the above entered work request from the queue 4. Complete filling out the work request with asset, location code, responsibility/assignment, work type and GL account information. 5. Assign a status of **Unapproved** to the work request and route the work request to the correct work group. | | |
| **Expected Outcome:**  Work Order created and routed to the correct work group. | | **Actual Outcome:** |
| **Test Outcome: PASS - FAIL** | | **Test Date:** |
| **Tester:** Prev Datta  **Tester Sign Off:** | | **Test Lead:** Prev Datta  **Test Lead Sign Off:** |

## Defect Tracking Process

All problems/defects need to be recorded and reviewed prior to making any configuration changes to ensure that the fix will not have a negative impact and that the test result was correct. The following process will be used to review all defects.

|  |
| --- |
|  |

## Problem Priorities

* 1. The person opening the issue should assign a priority. At anytime throughout the process, the original submitter and whomever the issue is assigned to may change the priority with mutual agreement.

The priorities are as follows:

* ***P1 - Critical -*** System wide failure. No one can complete any test cases/business processes.  Everything must be dropped until fixed.  Cannot go live with this defect.  Example:  Maximo database crash and no one can logon
* ***P2 - Major*** - Major failure that stops that test case/business process from proceeding and there are no work around, but other work can continue. Bumps all scheduled work until resolved. Cannot go live with this defect.  Example:  SQL error when trying to create a work order on WOTRACK
* ***P3- Minor*** –. Defect that has work around and does not prevent completion of the test case/business process.  Resolution can be scheduled.  Can go live with the defect but will require that a work around also be implemented.  Example: GL string cannot be picked from the value list but the correct one can be typed in.

***E - Enhancements -*** Functionality that was not in the original scope.  May or may not be required to go live.

Test the Environment



The integration points that will be tested:

* IMPACT to TSRM
* LDAP to TSRM/TIP/ITM
* AGG to Gateway to Reporter
* Impact to TeleAlert
* AGG and Display Layers to TIP
* TBSM to CCMDB
* IMPACT to ITNM

## TSRM High Availability

The TSRM cluster solution will be tested in two parts. The Application Sever (HTTP) level will be tested by re-producing a server failure and monitoring the uptime of the user and the re-direction of the different components. The Database level will be tested by monitoring the functionality/capability of the TSA-MP to re-direct the connection from the Application Server.



Test Summary

The test results are summarized in the table below with each test dealt with in more detail in the following pages.

## Omnibus Environment

| ID | Functionality | **Test** | **Expected Result** | **Notes** |
| --- | --- | --- | --- | --- |
| 5.1.1 | Primary ObjectServer component in collection tier running under Process Control | Verify that ObjectServer is running by examining the server process  ps –ef | grep objserv  Verify that ObjectServer is running under Process Control by examining the various configuration files  vi $OMNIHOME/etc/nco\_pa.conf | ObjectServer running  Process Control configured correctly |  |
| 5.1.2 | Primary ObjectServer in collection tier available for user logins when a valid username and password are supplied | Attempt to login to the ObjectServer with a valid username and password  $OMNIHOME/bin/nco\_sql –server COL\_P\_1 –username root –password ‘’ | Login to ObjectServer successful |  |
| 5.1.3 | Backup ObjectServer in the collection tier running under Process Control | Verify that the Backup ObjectServer is running by viewing the process  Ps –ef | grep XXX  Verify that the Backup ObjectServer is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf | Process running as expected  Process Control configured as expected |  |
| 5.1.4 | Primary and Backup collection layer Objectserver Failover test | Verify that the failover is configured correctly by stopping the primary Objectserver under PA to see if backup server takes over shown in the Omnibus Event List. | Error message when primary server disconnects. Switch from Primary to Back shown in Event list |  |
| 5.1.5 | Backup ObjectServer in collection tier available for user logins when a valid username and password are supplied | Verify that the Backup ObjectServer is available to users by logging in to the ObjectServer  $OMNIHOME/bin/nco\_sql –server XXX –username root –password ‘’ | ObjectServer available for login |  |
| 5.1.6 | Aggregation tier Backup ObjectServer configuration | Verify that the configuration of the Primary ObjectServer is appropriately present in the Backup ObjectServer by viewing data records in the Administration client  View Trigger data in XXX | Backup ObjectServer data configuration matches Primary instance |  |
| 5.1.7 | Primary ObjectServer component in Aggregation tier running under Process Control | Verify that ObjectServer is running by examining the server process  ps –ef | grep objserv  Verify that ObjectServer is running under Process Control by examining the various configuration files  vi $OMNIHOME/etc/nco\_pa.conf | ObjectServer running  Process Control configured correctly |  |
| 5.1.8 | Primary ObjectServer in aggregation tier available for user logins when a valid username and password are supplied | Attempt to login to the ObjectServer with a valid username and password  $OMNIHOME/bin/nco\_sql –server COL\_P\_1 –username root –password ‘’ | Login to ObjectServer successful |  |
|  | Backup ObjectServer in the aggregation tier running under Process Control | Verify that the Backup ObjectServer is running by viewing the process  Ps –ef | grep XXX  Verify that the Backup ObjectServer is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf | Process running as expected  Process Control configured as expected |  |
|  | Primary ObjectServer component in display tier running under Process Control | Verify that ObjectServer is running by examining the server process table  ps –ef | grep objserv  Verify that ObjectServer is running under Process Control by examining the various configuration files  vi $OMNIHOME/etc/nco\_pa.cof | ObjectServer running  Process Control configured correctly |  |
|  | Primary ObjectServer in display tier available for user logins when a valid username and password are supplied | Attempt to login to the ObjectServer with a valid username and password  $OMNIHOME/bin/nco\_sql –server DIS\_P\_1 –username root –password ‘’ | Login to ObjectServer successful |  |
|  | Backup ObjectServer in the display tier running under Process Control | Verify that the secondary ObjectServer is running by viewing the process table  Ps –ef | grep XXX  Verify that the secondary ObjectServer is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf | Process running as expected  Process Control configured as expected |  |
|  | Display tier Backup ObjectServer available | Verify that the secpndary ObjectServer is available to users by logging in to the ObjectServer  $OMNIHOME/bin/nco\_sql –server XXX –username root –password ‘’ |  | ObjectServer available for login |
|  | Objectserver gateways – C\_to\_A\_GATE\_P\_1 functionality | Verify that the Gateway is running by viewing the process table  Ps –ef | grep objserv\_bi  Verify that the Gateway is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf  Verify all fields are mapped from in the gateway map file | Gateway process running as expected  Process Control configured as required  Gateway running and all Objectserver data is synchronized |  |
|  | Objectserver gateways – C\_to\_A\_GATE\_P\_2 Functionality | Verify that the Gateway is running by viewing the process table  Ps –ef | grep objserv\_bi  Verify that the Gateway is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf  Verify all fields are mapped from in the gateway map file | Gateway process running as expected  Process Control configured as required  Gateway running and all Objectserver data is synchronized |  |
|  | Objectserver gateways – AGG\_GATE Functionality | Verify that the Gateway is running by viewing the process table  Ps –ef | grep objserv\_bi  Verify that the Gateway is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf  Verify all fields are mapped from in the gateway map file | Gateway process running as expected  Process Control configured as required  Gateway running and all Objectserver data is synchronized |  |
|  | Objectserver gateways – A\_To\_D\_GATE\_P\_1 Functionality | Verify that the Gateway is running by viewing the process table  Ps –ef | grep objserv\_bi  Verify that the Gateway is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf  Verify all fields are mapped from in the gateway map file | Gateway process running as expected  Process Control configured as required  Gateway running and all Objectserver data is synchronized |  |
|  | Objectserver gateways – A\_To\_D\_GATE\_P\_1 Functionality | Verify that the Gateway is running by viewing the process table  Ps –ef | grep objserv\_bi  Verify that the Gateway is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf  Verify all fields are mapped from in the gateway map file | Gateway process running as expected  Process Control configured as required  Gateway running and all Objectserver data is synchronized |  |
|  | SNMP Probe Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘snmp’  Verify format of events in the Objectserver from the GUI are as expected.  Verify the probe is running as a primary instance by viewing the log file and props file configuration  Verify Failover by stopping the primary instance of the probe and checking if alarms are sent by failover instance to the Objectserver. Check in the probe log files to ensure failover probe has taken over | Process Control configured correctly  Relevant event data is shown  Valid event data seen after Primary Probe stopped |  |
|  | Alcatel Probe for OMC1 Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  *nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘OMC1’*  Verify format of events in the Objectserver from the GUI are as expected | Probe running as expected  Relevant event data is shown  Process Control configured correctly |  |
|  | Alcatel Probe for OMC2 Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  *nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘OMC2’*  Verify format of events in the Objectserver from the GUI are as expected | Probe running as expected  Relevant event data is shown  Process Control configured correctly |  |
|  | Alcatel Probe for OMC4 Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  *nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘OMC4’*  Verify format of events in the Objectserver from the GUI are as expected | Probe running as expected  Relevant event data is shown  Process Control configured correctly |  |
|  | Alcatel Probe for OMC5 Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  *nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘OMC5’*  Verify format of events in the Objectserver from the GUI are as expected | Probe running as expected  Relevant event data is shown  Process Control configured correctly |  |
|  | Motorola Probe Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  *nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘moto’*  Verify format of events in the Objectserver from the GUI are as expected | Probe running as expected  Relevant event data is shown  Process Control configured correctly |  |
|  | Siemens Probe Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  *nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘siemens’*  Verify format of events in the Objectserver from the GUI are as expected | Probe running as expected  Relevant event data is shown  Process Control configured correctly |  |
|  | Nokia NetAct Probe Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘nokia’  Verify format of events in the Objectserver from the GUI are as expected.  Verify the probe is running as a primary instance by viewing the log file and props file configuration  Verify Failover by stopping the primary instance of the probe and checking if alarms are sent by failover instance to the Objectserver. Check in the probe log files to ensure failover probe has taken over | Process Control configured correctly  Relevant event data is shown  Valid event data seen after Primary Probe stopped |  |
|  | Tellabs Probe Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  *nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘Tellabs’*  Verify format of events in the Objectserver from the GUI are as expected | Probe running as expected  Relevant event data is shown  Process Control configured correctly |  |
|  | Huawei M2000 Probe for BSS1 Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘BSS1’  Verify format of events in the Objectserver from the GUI are as expected.  Verify the probe is running as a primary instance by viewing the log file and props file configuration  Verify Failover by stopping the primary instance of the probe and checking if alarms are sent by failover instance to the Objectserver. Check in the probe log files to ensure failover probe has taken over | Process Control configured correctly  Relevant event data is shown  Valid event data seen after Primary Probe stopped |  |
|  | Huawei M2000 Probe for BSS2 Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘BSS2’  Verify format of events in the Objectserver from the GUI are as expected.  Verify the probe is running as a primary instance by viewing the log file and props file configuration  Verify Failover by stopping the primary instance of the probe and checking if alarms are sent by failover instance to the Objectserver. Check in the probe log files to ensure failover probe has taken over | Process Control configured correctly  Relevant event data is shown  Valid event data seen after Primary Probe stopped |  |
|  | Huawei M2000 Probe for CORE1 Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘CORE1’  Verify format of events in the Objectserver from the GUI are as expected.  Verify the probe is running as a primary instance by viewing the log file and props file configuration  Verify Failover by stopping the primary instance of the probe and checking if alarms are sent by failover instance to the Objectserver. Check in the probe log files to ensure failover probe has taken over | Process Control configured correctly  Relevant event data is shown  Valid event data seen after Primary Probe stopped |  |
|  | Huawei M2000 Probe for CORE2 Functionality | Verify the probe process is running under process control  Verify recent alarms are available in the Active Event list  nco\_p\_sql –server COL\_P\_1 select count(\*) from alerts.status where Manager like ‘CORE2’  Verify format of events in the Objectserver from the GUI are as expected.  Verify the probe is running as a primary instance by viewing the log file and props file configuration  Verify Failover by stopping the primary instance of the probe and checking if alarms are sent by failover instance to the Objectserver. Check in the probe log files to ensure failover probe has taken over | Process Control configured correctly  Relevant event data is shown  Valid event data seen after Primary Probe stopped |  |
|  | Advanced ObjectServer configuration completed | Verify that appropriate users, roles and groups have been created within the Display ObjectServer by looking at data records in the Administration client  Select \* from master.names;  Test user access for 3 of the created users by logging in to the Event List application using those users login credentials  $OMNIHOME/bin/nco\_sql –server WATEENP –username XXX –password ‘XXXX’ | Users created with correct rights for roles and groups. |  |
|  | Process Control – Start-Up Scripts | Verify process control is accessible on each server using right credentials  Verify scripts have been configured by viewing the script on each server /etc/init/nco  Restart servers on which start-up scripts have been configured  Upon start-up all processes should run under process control | Upon reboot all processes to start-up under process control |  |

## Netcool Reporter Environment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Functionality** | **Test** | **Expected Result** | **Notes** |
|  | Historical Gateway running under Process Control | Verify that the Gateway is running by viewing the process table  Ps –ef | grep nco\_g\_oracle.bin  Verify that the Gateway is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf |  | Oracle gateway is running as expected  Process Control configured correctly |
|  | Historical Gateway functioning as expected | Verify that the Gateway is transferring event data appropriately to the configured Oracle instance by manually correlating event data in both the ObjectServer and Oracle  Within ObjectServer:  Select \* from alerts.status where ServerSerial = XXX;  Within Oracle:  Select \* from reporter\_status where ServerSerial = XXX; |  | Event data present and correct within Oracle environment |
|  | Backup Historical Gateway running under Process Control | Verify that the Gateway is running by viewing the process table  Ps –ef | grep nco\_g\_oracle.bin  Verify that the Gateway is running under Process Control by viewing the relevant configuration files  Vi $OMNIHOME/etc/nco\_pa.conf |  | Process running as expected  Process Control configured as required |
|  | Backup Historical Gateway functioning as expected | Create a failover scenario by stopping the primary Gateway instance; ensure that the Gateway transfers event data appropriately to the configured Oracle instance by manually correlating event data in both the ObjectServer and Oracle |  | Historical Gateways continue to work as expected during failover scenario |
|  | Reporter Server running under Process Control | Verify that the Reporter Server is running by viewing the process table  Ps –ef| grep java  Verify that the Reporter Server is running under Process Control by viewing the relevant configuration files  Verify that Reporter instance is able to view appropriate historical data in the configured Oracle instance by running a pre-configured report, and viewing the result data |  | Reporter running as expected  Reports show apprpriate data |
|  | Standby Reporter Server ready to run | Verify that the Reporter Server is ready to run by examining the relevant configuration files  Vi $REPORTERHOME/etc/\*  Verify that Reporter instance is able to view appropriate historical data in the configured Oracle instance by running a pre-configured report, and viewing the result data |  | Reporter configuration as expected  Reports show correct content when run |
|  | Reporter server available for user logins when a valid username and password are supplied | Attempt to login to WebTop with a valid username and password  http://10.231.105.13:16310 | Login Successful |  |
|  | Accessible reports exported from Reporter. |  |  |  |

## ITNM IP Environment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Functionality** | **Test** | **Expected Result** | **Notes** |
|  | NMIP running and administrator able to connect | Verify that NMIP components are running by examining the process table  Ps –ef | grep nco\_p  Verify that users are able to login to the configuration and user sections of the NMIP interface | NMIP processes running as expected  Users are able to run NMIP Admin tools |  |
|  | NMIP is configured to send events to the ObjectServer | Verify that NMIP is creating suitable events in the Primary ObjectServer  Select count(\*) from alerts.status where Manager like ‘NMIP’;  Vi $NCHOME/logs/mttrapd.log | Suitable events are present within the ObjectServer |  |
|  | Backup NMIP running and administrator able to connect | Verify that NMIP components are running by examining the process table  Ps –ef | grep nco\_p  Verify that users are able to login to the configuration and user sections of the NMIP interface | NMIP processes running as expected  Users are able to run NMIP Admin tools |  |
|  | Backup NMIP is configured to send events to the ObjectServer | Verify that NMIP is creating suitable events in the Primary ObjectServer  Select count(\*) from alerts.status where Manager like ‘NMIP’ | Required events are present within ObjectServer |  |
|  | Backup NMIP communicates with Primary instance, and assumes functional responsibility when the Primary instance fails | Verify that the Backup instance is communicating with the Primary instance by viewing log files  Vi $NCHOME/log/ncp\*  Verify that the Backup instance assumes functional responsibility by deliberately failing the Primary instance | Primary & Backup NMIP instances are working together as expected |  |
|  | IP Core network discovered | Verify that the MPLS Core network has been discovered by NMIP by viewing network topology maps in the NMIP interface | Topology views available |  |

## Netcool/Impact Environment

| **ID** | **Functionality** | **Test** | **Expected Result** | **Notes** |
| --- | --- | --- | --- | --- |
|  | Impact application has been installed and is functioning correctly | Verify that the Impact components are running by examining the process table on the primary Impact server  Ps –ef | grep NCI | Impact server running as expected |  |
|  | Impact application is able to connect to ObjectServer components | Verify that Impact is able to connect to ObjectServer components by viewing an ObjectServer data type in the Impact web based GUI  <http://XXX:8080/nci> | Impact server available for user connections |  |
|  | Secondary Impact application has been installed and is functioning correctly | Verify that the Impact components are running by examining the process table on the secondary Impact server  Ps –ef | grep XXX | Impact server running as expected |  |
|  | Secondary Impact application is able to connect to ObjectServer components | Verify that the secondary Impact server is able to connect to ObjectServer components by viewing an ObjectServer data type in the Impact web based GUI  <http://XXX>:8080/nci | Impact server available for user connections |  |
|  | Primary and Secondary Impact servers are operating in cluster mode correctly | Verify that the two Impact servers are connected to the same cluster and both processing events as appropriate  Vi $IMPACT\_HOME/etc/server.props | Impact servers configured as required |  |
|  | StandardEnrichment policy functions as expected | Verify that the specified policy is xxx  Vi $IMPACT\_HOME/logs/StandardEnrichmentt\_Policy.log | Policy populating events as expected |  |
|  | MaintenancePolicies policy functions as expected | Verify that the specified policy is xxx  Vi $IMPACT\_HOME/logs/MaintenancePolicies\_Policy.log | Policy populating events as expected |  |
|  | IncidentRecordPolicies policy functions as expected | Verify that the specified policy is xxx  Vi $IMPACT\_HOME/logs/ | Policy populating events as expected |  |
|  | SyntheticEvents policy functions as expected | Verify that the specified policy is xxx  Vi $IMPACT\_HOME/logs/ | Policy populating events as expected |  |
|  | BSSEnvironmentalAlarms policy functions as expected | Verify that the specified policy is xxx  Vi $IMPACT\_HOME/logs/ | Policy populating events as expected |  |
|  | ParentChild policy functions as expected | Verify that the specified policy is xxx  Vi $IMPACT\_HOME/logs/ | Policy populating events as expected |  |

## TBSM Environment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Functionality** | **Test** | **Expected Result** | **Notes** |
|  | TBSM application running and users able to connect | Verify that TBSM application is running by examining the process table  ps –ef | grep nco  Verify that users are able to login to the configuration and user sections of the TBSM interface  http://XXX:8080 |  |  |
|  | BSC Hierarchical Overview Dashboard available | Verify that BSC Hierarchical Overview dashboard is available  http://XXX:8080 |  |  |

## TSRM Environment

| **ID** | **Functionality** | **Test** | **Expected Result** | **Notes** |
| --- | --- | --- | --- | --- |
|  | TSRM application installed and running, users able to connect | Verify that TSRM is running by connecting to <http://XXX/maximo>  Entering a valid username and password combination on the TSRM login screen will allow the user to login to the application and be presented with the appropriate dashboard view | TSRM available to connect, users able to login |  |
|  | New Incidents able to be created manually | Follow the supplied ‘User Guide’ to successfully create a new ticket within TSRM following the manual process | New ticket successfully created as expected |  |
|  | New Incidents able to be created from within Netcool | Generate a test event in the Netcool environment, and create a new Incident in TSRM by running the appropriate Netcool tool.  /tmp/create\_test\_event.sh  A new Incident will be created in TSRM that contains the Netcool ServerSerial reference, and also reflect the Netcool event status | Test event created in Netcool which is in turn used to create an Incident in TSRM |  |
|  | Clearance of Netcool event appropriately updates relevant Incident in TSRM | Update the Netcool event created in previous test so that it clears.  Update alerts.status set Severity = 0 where ServerSerial = XXX;  The TSRM Incident will reflect that that the state of the Netcool event has changed to a cleared state. | Netcool event updated as required  TSRM Incident reflects new Netcool event status |  |
|  | Clearance of TSRM Incident appropriately updates relevant event in Netcool | Generate a test event in the Netcool environment, and create a new Incident in TSRM by running the appropriate Netcool tool.  /tmp/create\_test\_event.sh  Once the new Incident has been created in TSRM, set the Incident status to resolved. The source Netcool event will by automatically cleared, and the Netcool Status field in TSRM will show that the event has cleared. | Test event created as required in Netcool  Netcool event cleared as required |  |
|  | TSRM is configured as per the HLD   * Base setup configuration * Organisation configuration * Security configuration * People configuration * Incident Management configuration | 1. Organisation of “Wateen Telecom” visible in Organisation application of TSRM. 2. Wateen “users and groups” visible in the person, person groups, users and security groups applications. 3. Assign incident to a user, check visibility of the incident as the user in the start center, update the incident record, change the status, and resolve and close the incident | TSRM configuration is included as required |  |
|  | TSRM reports available for viewing | Connect to TSRM as a user that is allowed to view report data. Run the ‘Incident List’ report to view detailed information about the Incidents that have been created. Verify that the report contains accurate data by viewing the content of various Incidents included in the report. | TSRM reports available |  |

Test Cases

## Omnibus

| Test Number: 5.1.1 | Revision: 1.0 | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus -- Objectserver | | Requirement Catalogue reference: Custom | |
| Test Title: | Primary ObjectServer component in collection tier running under Process Control | | |
| Test Purpose | To test the Primary ObjectServer component in collection tier running successfully under Process Control | | |
| Procedure | * Verify Objectserver process is running by ps-ef|grep COL\_P\_1 * Verify that ObjectServer is running under Process Control by examining the various configuration files * Verify ObjectServer is running under Process Control by accessing process control and checking the process status is ‘RUNNING’   Mark Schedule tasks "Complete". | | |
| **Results** | Complete. COL\_P\_1 server running under process control. Process ID also shown when ps-ef|grep COL\_P\_1 run. No errors found | | |
| **Reason for failure or comments** | None | | Pass  Fail |

| Test Number: 5.1.2 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus – Objectserver | | | Requirement Catalogue reference: Custom | |
| Test Title: | Backup ObjectServer in the collection tier running under Process Control | | | |
| Test Purpose | To test the failover Objectserver is running as expected | | | |
| Procedure | * Verify Backup Objectserver process is running by ps-ef|grep COL\_P\_2 * Verify that Backup ObjectServer is running under Process Control by examining the various configuration files * Verify Backup ObjectServer is running under Process Control by accessing process control and checking the process status is ‘RUNNING’   Mark Schedule tasks "Complete". | | | |
| **Results** | Complete. Ps-ef|grep shows process. Objectserver running under process control. No errors found. | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.1.2 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus – Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Primary and Backup collection Objectserver Failover Test | | | |
| Test Purpose | To test the failover between Primary and Backup Collection Objectservers | | | |
| Procedure | * User to login to EventList GUI / Desktop Client COL\_P\_1 * Check Objectserver is running under process control * Clean Shutdown of the Primary Collection Objectserver * User should receive a message showing Primary server is down * User EventList should display events from COL\_P\_2 * Start Primary Objectserver under process control   Mark Schedule tasks "Complete". | | | |
| **Results** | Complete. Successful login to GUI. Objectserver running under PA. After shutdown of Objectserver message received as expected and events displayed from failover server. | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.1.3 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Backup ObjectServer in collection tier available for user logins when a valid username and password are supplied | | | |
| Test Purpose | To test the Backup objectsever is accessible by users. | | | |
| Procedure | * Verify that the Backup ObjectServer is available to users by logging in to the ObjectServer Event list GUI via Desktop Client. * Verify that the Backup ObjectServer is available to users by logging in to the ObjectServer using nco\_sql. $OMNIHOME/bin/nco\_sql –server XXX –username root –password ‘   Mark Schedule tasks "Complete". | | | |
| **Results** | Complete. Able to access failover Objectserver GUI client and sql into Objectserver | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.1.4 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Aggregation tier ObjectServer configuration | | | |
| Test Purpose | To test the configuration of the Primary Objectserver is appropriately present in the Backup Objectserver. | | | |
| Procedure | * Verify that the configuration of the Primary ObjectServer configuration is appropriately present in the Backup ObjectServer by viewing data records in the Administration client * View Trigger data in XXX   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.1.5 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Primary ObjectServer component in Aggregation tier running under Process Control | | | |
| Test Purpose | To test the Primary ObjectServer component in aggregation layer running successfully under Process Control | | | |
| Procedure | * Verify Objectserver process is running by ps-ef|grep AGG\_P\_1 * Verify that ObjectServer is running under Process Control by examining the various configuration files * Verify ObjectServer is running under Process Control by accessing process control and checking the process status is ‘RUNNING’   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.1.6 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Primary ObjectServer in aggregation tier available for user logins when a valid username and password are supplied | | | |
| Test Purpose | To test Objectservers are accessible | | | |
| Procedure | * User to login to EventList GUI AGG\_P\_1 * Attempt to login to the ObjectServer SQL with a valid username and password * $OMNIHOME/bin/nco\_sql –server AGG\_P\_1 –username root –password ‘’   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.1.7 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Backup ObjectServer in the aggregation tier running under Process Control | | | |
| Test Purpose | To test the failover Objectserver is running as expected | | | |
| Procedure | * Verify that the Backup ObjectServer is available to users by logging in to the ObjectServer Event list GUI via Desktop Client. * Verify that the Backup ObjectServer is available to users by logging in to the ObjectServer using nco\_sql. $OMNIHOME/bin/nco\_sql –server XXX –username root –password ‘   Mark Schedule tasks "Complete". | | | |
| **Results** | Complete. Able to access failover Objectserver GUI client and sql into Objectserver | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.1.8 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Primary ObjectServer component in display tier running under Process Control | | | |
| Test Purpose | To test the Primary ObjectServer component in collection tier running successfully under Process Control | | | |
| Procedure | * Verify Objectserver process is running by ps-ef|grep DIS\_P\_1 * Verify that ObjectServer is running under Process Control by examining the various configuration files * Verify ObjectServer is running under Process Control by accessing process control and checking the process status is ‘RUNNING’   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.1.9 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Primary ObjectServer in display tier available for user logins when a valid username and password are supplied | | | |
| Test Purpose | To test Objectservers are accessible | | | |
| Procedure | * User to login to EventList GUI DIS\_P\_1 * Attempt to login to the ObjectServer SQL with a valid username and password * $OMNIHOME/bin/nco\_sql –server DIS\_P\_1 –username root –password ‘’   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.2.0 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Secondary ObjectServer in the display tier running under Process Control | | | |
| Test Purpose | To test the failover Objectserver is running as expected | | | |
| Procedure | * Verify that the Secondary ObjectServer is available to users by logging in to the ObjectServer Event list GUI via Desktop Client. * Verify that the Backup ObjectServer is available to users by logging in to the ObjectServer using nco\_sql. $OMNIHOME/bin/nco\_sql –server DIS\_P\_2 –username root –password ‘   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.2.1 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Display tier Backup ObjectServer available | | | |
| Test Purpose | To test Objectservers are accessible | | | |
| Procedure | * User to login to EventList GUI DIS\_P\_2 * Attempt to login to the ObjectServer SQL with a valid username and password * $OMNIHOME/bin/nco\_sql –server DIS\_P\_1 –username root –password ‘’   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.2.2 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Objectserver | | | Requirement Catalogue reference: | |
| Test Title: | Failover Display Objectservers | | | |
| Test Purpose | To test the failover between Primary and Backup Display Objectservers | | | |
| Procedure | * User to login to Webtop and open an active event list * Check Objectserver is running under process control * Clean Shutdown of the Primary Display Objectserver * Verify that the server backup server has taken over by viewing the webtop log file. * Users should be able to view events without errors * Start Primary Objectserver under process control | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | | None | | |

| Test Number: 5.2.3 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Gateway | | | Requirement Catalogue reference: | |
| Test Title: | Objectserver gateways – C\_to\_A\_GATE\_P\_1 functionality | | | |
| Test Purpose | To test the Objectserver gateway C\_to\_A\_GATE\_P\_1 is running as expected and all Objectserver data is synchronized. | | | |
| Procedure | * Verify that the Gateway is running by viewing the process table * Ps –ef | grep objserv\_bi * Verify that the Gateway is running under Process Control by viewing the relevant configuration files * Vi $OMNIHOME/etc/nco\_pa.conf * Verify all fields are mapped from in the gateway map file * Verify that data is synchronized between the ObjectServers by looking at data records in the Administration client Select count(\*) from alerts.status from each ObjectServer * Create a failover scenario by stopping the Primary ObjectServer; ensure that events continue to be added to the Backup ObjectServer, and that data is appropriately synchronized after re-starting the Primary ObjectServer   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.2.4 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus – Gateway | | | Requirement Catalogue reference: | |
| Test Title: | Objectserver gateways – C\_to\_A\_GATE\_P\_2 Functionality | | | |
| Test Purpose | To test the Objectserver gateway C\_to\_A\_GATE\_P\_2 is running and all Objectserver data is synchronized | | | |
| Procedure | * Verify that the Gateway is running by viewing the process table * Ps –ef | grep objserv\_bi * Verify that the Gateway is running under Process Control by viewing the relevant configuration files * Vi $OMNIHOME/etc/nco\_pa.conf * Verify all fields are mapped from in the gateway map file * Verify that data is synchronized between the ObjectServers by looking at data records in the Administration client Select count(\*) from alerts.status from each ObjectServer * Create a failover scenario by stopping the Primary ObjectServer; ensure that events continue to be added to the Backup ObjectServer, and that data is appropriately synchronized after re-starting the Primary ObjectServer   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.2.5 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus – Gateway | | | Requirement Catalogue reference: | |
| Test Title: | Objectserver gateways – AGG\_GATE Functionality | | | |
| Test Purpose | To test the Objectserver gateway AGG\_GATE is running and all Objectserver data is synchronized | | | |
| Procedure | * Verify that the Gateway is running by viewing the process table * Ps –ef | grep objserv\_bi * Verify that the Gateway is running under Process Control by viewing the relevant configuration files * Vi $OMNIHOME/etc/nco\_pa.conf * Verify all fields are mapped from in the gateway map file * Verify that data is synchronized between the ObjectServers by looking at data records in the Administration client Select count(\*) from alerts.status from each ObjectServer * Create a failover scenario by stopping the Primary ObjectServer; ensure that events continue to be added to the Backup ObjectServer, and that data is appropriately synchronized after re-starting the Primary ObjectServer   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.2.6 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus – Gateway | | | Requirement Catalogue reference: | |
| Test Title: | Objectserver gateways – A\_To\_D\_GATE\_P\_1 Functionality | | | |
| Test Purpose | To test the Objectserver gateway A\_To\_D\_GATE\_P\_1 is running and all Objectserver data is synchronized | | | |
| Procedure | * Verify that the Gateway is running by viewing the process table * Ps –ef | grep objserv\_bi * Verify that the Gateway is running under Process Control by viewing the relevant configuration files * Vi $OMNIHOME/etc/nco\_pa.conf * Verify all fields are mapped from in the gateway map file * Verify that data is synchronized between the ObjectServers by looking at data records in the Administration client Select count(\*) from alerts.status from each ObjectServer * Create a failover scenario by stopping the Primary ObjectServer; ensure that events continue to be added to the Backup ObjectServer, and that data is appropriately synchronized after re-starting the Primary ObjectServer   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.2.7 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus - - Gateway | | | Requirement Catalogue reference: | |
| Test Title: | Objectserver gateways – A\_To\_D\_GATE\_P\_2 Functionality | | | |
| Test Purpose | To test the Objectserver gateway A\_To\_D\_GATE\_P\_2 is running and all Objectserver data is synchronized | | | |
| Procedure | * Verify that the Gateway is running by viewing the process table * Ps –ef | grep objserv\_bi * Verify that the Gateway is running under Process Control by viewing the relevant configuration files * Vi $OMNIHOME/etc/nco\_pa.conf * Verify all fields are mapped from in the gateway map file * Verify that data is synchronized between the ObjectServers by looking at data records in the Administration client Select count(\*) from alerts.status from each ObjectServer * Create a failover scenario by stopping the Primary ObjectServer; ensure that events continue to be added to the Backup ObjectServer, and that data is appropriately synchronized after re-starting the Primary ObjectServer   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.2.8 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus - - Probe | | | Requirement Catalogue reference: | |
| Test Title: | SNMP Probe Functionality | | | |
| Test Purpose | To test the functionality of the primary snmp probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected. * Check the probe is running as a primary instance * Stop the primary instance of the probe * Check if alarms are sent by failover instance * Start-up primary instance   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.2.9 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus - - Probe | | | Requirement Catalogue reference: | |
| Test Title: | Alcatel Probe for OMC1 Functionality | | | |
| Test Purpose | To test the functionality of the primary snmp probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.0 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus - - Probe | | | Requirement Catalogue reference: | |
| Test Title: | Alcatel Probe for OMC2 Functionality | | | |
| Test Purpose | To test the functionality of the primary snmp probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.1 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus - - Probe | | | Requirement Catalogue reference: | |
| Test Title: | Alcatel Probe for OMC4 Functionality | | | |
| Test Purpose | To test the functionality of the primary snmp probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.2 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus - - Probe | | | Requirement Catalogue reference: | |
| Test Title: | Alcatel Probe for OMC5 Functionality | | | |
| Test Purpose | To test the functionality of the primary snmp probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.4 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus - - Probe | | | Requirement Catalogue reference: | |
| Test Title: | Motorola Probe Functionality | | | |
| Test Purpose | To test the functionality of the primary snmp probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.5 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus - - Probe | | | Requirement Catalogue reference: | |
| Test Title: | Siemens Switch Commander Probe | | | |
| Test Purpose | To test the functionality of the primary snmp probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.6 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Probe | | | Requirement Catalogue reference: | |
| Test Title: | Nokia Netact Probe Functionality | | | |
| Test Purpose | To test the functionality of the primary snmp probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected. * Check the probe is running as a primary instance * Stop the primary instance of the probe * Check if alarms are sent by failover instance * Start-up primary instance   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.6 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Probe | | | Requirement Catalogue reference: | |
| Test Title: | Huawei M2000 Probe for BSS 1 Functionality | | | |
| Test Purpose | To test the functionality of the primary Huawei M2000 probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected. * Check the probe is running as a primary instance * Stop the primary instance of the probe * Check if alarms are sent by failover instance * Start-up primary instance   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.7 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Probe | | | Requirement Catalogue reference: | |
| Test Title: | Huawei M2000 Probe for BSS 2 Functionality | | | |
| Test Purpose | To test the functionality of the primary Huawei M2000 probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected. * Check the probe is running as a primary instance * Stop the primary instance of the probe * Check if alarms are sent by failover instance * Start-up primary instance   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.8 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Probe | | | Requirement Catalogue reference: | |
| Test Title: | Huawei M2000 Probe for CORE 1 Functionality | | | |
| Test Purpose | To test the functionality of the primary Huawei M2000 probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected. * Check the probe is running as a primary instance * Stop the primary instance of the probe * Check if alarms are sent by failover instance * Start-up primary instance   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.3.9 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Probe | | | Requirement Catalogue reference: | |
| Test Title: | Huawei M2000 Probe for CORE 2 Functionality | | | |
| Test Purpose | To test the functionality of the primary Huawei M2000 probe. | | | |
| Procedure | * Check the probe process is running under process control * Check recent alarms are available in the Active Event list * Check format of events in the Objectserver are as expected. * Check the probe is running as a primary instance * Stop the primary instance of the probe * Check if alarms are sent by failover instance * Start-up primary instance   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.4.0 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Probe | | | Requirement Catalogue reference: | |
| Test Title: | Advanced ObjectServer configuration completed | | | |
| Test Purpose | To test the Advanced ObjectServer configuration settings on the Display Objectservers for user accessibility | | | |
| Procedure | * Verify that appropriate users, roles and groups have been created within the Display ObjectServer by looking at data records in the Administration client   Select \* from master.names;   * Test user access for 3 of the created users by logging in to the Event List application using those users login credentials   $OMNIHOME/bin/nco\_sql –server WATEENP –username XXX –password ‘XXXX’  Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 5.4.1 | Revision: 1.0 | | Author: Fakhra Fiaz | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** OMNIbus Environment | | | **Standard/Customization:** Standard | |
| **Product:** Netcool OMNIbus Process Control | | | Requirement Catalogue reference: | |
| Test Title: | Process Control – Start-Up Scripts | | | |
| Test Purpose | To test the all components are running under process control and test start-up scripts. | | | |
| Procedure | * Test process control is accessible on each server using right credentials * Restart servers on which start-up scripts have been configured * Upon start-up all processes should run under process control   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM

| Test Number: 6.0.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Windows Server monitoring | | | |
| Test Purpose | Require Windows server monitoring | | | |
| Procedure | From the ITM console, expand the windows computer systems and select a windows ITM agent.  Browse through the collected attributes for the server  Compare the attribute values with those manually collected from the server | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 6.0.2 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM server availability monitoring | | | |
| Test Purpose | Enable Poll on servers for availability | | | |
| Procedure | Simulate a server failure by simultaneously killing all ITM processes on a monitored server.  View the ITM event list to ensure that the relevant event is present  View the Omnibus event list to ensure that the relevant event is present | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 6.0.3 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM windows service monitoring | | | |
| Test Purpose | Enable Windows Service Monitoring | | | |
| Procedure | Simulate a windows service failure by manually stopping a service marked as automatic start  View the ITM event list to ensure that the relevant event is present  View the Omnibus event list to ensure that the relevant event is present | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 6.0.4 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM CPU monitoring test | | | |
| Test Purpose | Test CPU Monitoring | | | |
| Procedure | Simulate a CPU problem by lowering the monitored threshold in CPU situations to a lower value  Simulate a memory problem by lowering the monitored threshold for memory situations to a lower level | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 6.0.4 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM logfile monitoring test | | | |
| Test Purpose | Test Log File Monitoring | | | |
| Procedure | Simulate an error log problem by echoing some known problem text into a monitored log file  For each simulated problem, view the ITM and Omnibus event lists to ensure that the relevant event is present | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 6.0.5 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM process monitoring | | | |
| Test Purpose | Monitor Processor Application Information | | | |
| Procedure | Simulate application processes problems by adding a new process ‘test’ to each application process monitoring situation.  For each simulated problem, view the ITM and Omnibus event lists to ensure that the relevant event is present | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 6.0.8 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | Logfile monitoring functionality | | | |
| Test Purpose | Log file Monitoring | | | |
| Procedure | For each monitored logfile configuration, echo a known error string into the logfile.  Ensure that an event is generated both within ITM and Omnibus | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 6.0.9 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Coverage | | | |
| Test Purpose | In total there are 413 servers to monitor | | | |
| Procedure | Use the output of the tacmd listSystems command to count the number of servers being monitored by ITM. Compare this to the 413 value. | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | | Vendors have not yet provided authorization to install agents | | |

## Impact

| Test Number: 7.0.1 | Revision: 1.0 | | Author: Chris Janes | Date: 01/09/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Impact | | | **Standard/Customisation:** | |
| **Product:** Impact | | | **Requirement Catalogue reference:** | |
| Test Title: | Event Enrichment BSS Events | | | |
| Test Purpose | To Ensure that Events are being correctly enriched by impact after they are inserted into the Aggregation layer ObjectServers | | | |
| Procedure | Insert Test event using Script BSS\_Env\_Enrich.sh  Check that the following fields are correctly Enriched  CovCity  ManCity  Site  Domain  OCName  NePriority | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 7.0.2 | Revision: 1.0 | | Author: Chris Janes | Date: 01/09/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Impact | | | **Standard/Customisation:** | |
| **Product:** Impact | | | **Requirement Catalogue reference:** | |
| Test Title: | Event Enrichment Motorola Events | | | |
| Test Purpose | To Ensure that Events are being correctly enriched by impact after they are inserted into the Aggregation layer ObjectServers | | | |
| Procedure | Insert Test event using Script Motorola\_Enrich.sh  Check that the following fields are correctly Enriched  Node  CovCity  ManCity  Site  Domain  OCName  NePriority | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 7.0.2 | Revision: 1.0 | | Author: Chris Janes | Date: 01/09/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Impact | | | **Standard/Customisation:** | |
| **Product:** Impact | | | **Requirement Catalogue reference:** | |
| Test Title: | Event Enrichment Motorola Events | | | |
| Test Purpose | To Ensure that Events are being correctly enriched by impact after they are inserted into the Aggregation layer ObjectServers | | | |
| Procedure | Insert Test event using Script Motorola\_Enrich.sh  Check that the following fields are correctly Enriched  Node  CovCity  ManCity  Site  Domain  OCName  NePriority | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Maintenance

| Test Number: 7.1.1 | | Revision: 1.0 | | Author: Alex Silva | Date: 01/09/2010 |
| --- | --- | --- | --- | --- | --- |
| **Test Category:** ML\_Maintenance | | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | | **Requirement Catalogue reference:** | |
| Test Title: | ML\_Maintenance – Node In Maintenance: | | | | |
| Test Purpose | To test single node scheduled in maintenance | | | | |
| Procedure | 1. Open an event list and create a filter Identifier like ‘Maintenance’. 2. Check the maintenance time window for Node = ‘HYD498A\_\_CtyzenPlz\_2’ in the table MainWindow in MLDB 3. Login into omnibus1 (10.231.105.11) server as user netcool and insert the test event by executing the *in\_maintenance.sh* script that is located in the home directory (/home/netcool) 4. Check that the MaintFlag is set to 99 | | | | |
| Results |  | | | | Pass  Fail |
| **Reason for failure or comments** | | |  | | |

| Test Number: 7.1.2 | | Revision: 1.0 | | Author: Alex Silva | Date: 1/09/2010 |
| --- | --- | --- | --- | --- | --- |
| **Test Category:** ML\_Maintenance | | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | | **Requirement Catalogue reference:** | |
| Test Title: | ML\_Maintenance – End Time | | | | |
| Test Purpose | To test the end time of single node scheduled in maintenance | | | | |
| Procedure | 1. Open an event list and create a filter Identifier like ‘Maintenance’. 2. Check the maintenance time window for Node = ‘HYD498A\_CtyzenPlz\_2’ in the table MainWindow in MLDB 3. Login into omnibus1 (10.231.105.11) server as user netcool and insert the test event by running the in\_maintenance.sh script that is located in the home directory (/home/netcool) 4. Check that the MaintFlag is set to 99 5. Now run the script *update\_maintenance.sh*. This script will change the MaintEndTime to currentime. 6. Refresh the event list several times and check that the MaintFlag is set back to 0 in order to be reprocessed by the ML\_Maintenance policy and then set to 1. | | | | |
| Results |  | | | | Pass  Fail |
| **Reason for failure or comments** | | |  | | |

| Test Number: 7.1.3 | | Revision: 1.0 | | Author: Alex Silva | Date: 1/09/2010 |
| --- | --- | --- | --- | --- | --- |
| **Test Category:** ML\_Maintenance | | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | | **Requirement Catalogue reference:** | |
| Test Title: | ML\_Maintenance – In Service | | | | |
| Test Purpose | To test single node not scheduled in maintenance (In Service) | | | | |
| Procedure | 1. Open an event list and create a filter Identifier like ‘Maintenance’. 2. Check that the Node = ‘HYD469A\_CtyzenPlz\_2’ is not in the MainWindow table in MLDB 3. Login into omnibus1 (10.231.105.11) server as user netcool and insert the test event by running the not\_in\_maintenance.sh script that is located in the home directory (/home/netcool) 4. Check that the MaintFlag is set to 1 | | | | |
| Results |  | | | | Pass  Fail |
| **Reason for failure or comments** | | |  | | |

## Maintenance

| Test Number:7.2.1 | Revision: 1.0 | | Author: Chris Janes | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** BSS Environmental | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | BSS Environmental - Basic Functionality: | | | |
| Test Purpose | To test Basic Functionality of the BSS Environmental event Handling | | | |
| Procedure | pre-requisites   * + All Event need to be     - Domain = ‘BSS’     - EventType = ‘EnvironmentalAlarm’   + Event 1 - Temperature Alarm Node A     - Node = ‘HYD498A\_\_CtyzenPlz\_2’     - AddText = ‘BTS-EXTERNAL [9]External Temperature [89]’   + Event 2 - Temperature Alarm Node B     - Node = ‘HYD410A\_\_SiteArea\_2     - AddText = ‘BTS-EXTERNAL [9]External Temperature [89]’   + Event 3 - Voltage Alarm Node A     - Node = ‘HYD498A\_\_CtyzenPlz\_2’     - AddText = ‘BSC-EXTERNAL [50] Low Voltage [7]’   + Event 4 - Voltage Alarm Node B     - Node = ‘HYD410A\_\_SiteArea\_2     - AddText = ‘BSC-EXTERNAL [50] Low Voltage [7]’  1. Insert the test event (Event 1) 2. Check the event appears in WebTop 3. Wait for Synthetic event ( x Min) 4. Check Synthetic event is generated 5. Check Event Journal of synthetic event of entry about the original event 6. Check SyntheticServerSerial and Synthetic ServerName fields are populated in the original event 7. Wait for 1 minute check TTNumber is populated in the synthetic event 8. Check Incident Record in TSRM 9. Wait for a further minute and check TTNumber is populated in the original event 10. Clear the original event 11. Check that the synthetic event is set to clear 12. Check that the Incident Record is set to resolved | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 7.2.2 | Revision: 1.0 | | Author: Chris Janes | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** BSS Environmental | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | BSS Environmental - Single Nodes of the differing types: | | | |
| Test Purpose | To test Single Nodes of the different types of events within BSS Environmental event handling | | | |
| Procedure | 1. Insert the test event (Event 1) 2. Check that synthetic event and Incident record are created 3. Insert second test event (Event 3) 4. Check that it’s SyntheticServerSerial, Synthetic ServerName and TTNumber are populated 5. Check journal of synthetic event has details of both source events 6. Clear Event 1 7. Check that the synthetic event is NOT cleared 8. Clear Event 3 9. Check that the synthetic event is set to clear 10. Check that the Incident Record is set to resolved | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 7.2.3 | Revision: 1.0 | | Author: Chris Janes | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** BSS Environmental | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | BSS Environmental - Multiple Nodes with a single type: | | | |
| Test Purpose | To test Multiple Nodes with a single type within BSS Environmental event handling | | | |
| Procedure | 1. Insert the test event (Event 1) 2. Check that synthetic event and Incident record are created 3. Insert second test event (Event 2) 4. Check that an additionalsynthetic event and Incident record are created 5. Clear Event 1 6. Check that the synthetic event associate to Event 1 is not set to clear 7. Check that the Incident Record associate to Event 1 is not set to resolved 8. Clear Event 2 9. Check that the synthetic event associate to Event 2 is set to clear 10. Check that the Incident Record associate to Event 2 is set to resolved | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 7.2.4 | Revision: 1.0 | | Author: Chris Janes | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** BSS Environmental | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | BSS Environmental - Multiple Node of the multiple type: | | | |
| Test Purpose | To Test Multiple Nodes of the multiple type within BSS Environmental event handling | | | |
| Procedure | 1. Insert the test event (Event 1) 2. Check that synthetic event and Incident record are created 3. Insert second test event (Event 2) 4. Check that an additional synthetic event and Incident record are created 5. Insert third test event (Event 3) 6. Check that it’s SyntheticServerSerial, Synthetic ServerName and TTNumber are populated 7. Check journal of synthetic event associated with Events 1 and 2 has details of both source events 8. Insert forth test event (Event 4) 9. Check that it’s SyntheticServerSerial, Synthetic ServerName and TTNumber are populated 10. Check journal of synthetic event associated with Events 3 and 4 has details of both source events 11. Clear Event 1 12. Check that the synthetic event associate to Events 1 and 3 is not set to clear 13. Check that the Incident Record associate to Events 1 and 3 is not set to resolved 14. Clear Event 2 15. Check that the synthetic event associate to Events 2 and 4 is not set to clear 16. Check that the Incident Record associate to Events 2 and 4 is not set to resolved 17. Clear Event 4 18. Check that the synthetic event associate to Events 2 and 4 is set to clear 19. Check that the Incident Record associate to Events 2 and 4 is set to resolved 20. Clear Event 3 21. Check that the synthetic event associate to Events 1 and 3 is set to clear 22. Check that the Incident Record associate to Events 1 and 3 is set to resolved | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Cell Down

| Test Number: 7.3.0 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Cell Down | | | **Standard/Customization:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | Cell outage alarms | | | |
| Test Purpose | When all cell outage alarming then policy should create synthetic event as “Site Down” and raise TT. Also, when all cell resolution alarms come then synthetic event should be clear and incident record set to resolved. | | | |
| Procedure | Please ensure the following pre-requisites are met before starting these test cases.  All cell outage events need to be  AlertGroup = ‘Cell Out of Service’  MainFlag = 1  ImpactFlag = 5 (Set by Multisite down policy)  We need cell outage events for particular site. Let’s assume, we have 5 cells with ‘Cell Out of Service’ for particular site i.e. ‘LHR860H\_FRF\_4(BSC08).  Event 1 - “LHR19822\_\_S\_KachaRahimRoad” outage alarm  Event 2 - “LHR29822\_\_S\_KachaRahimRoad” outage alarm  Event 3 - “LHR39822\_\_S\_KachaRahimRoad” outage alarm  Event 4 - “LHR49822\_\_S\_KachaRahimRoad” outage alarm  Event 5 - “LHR59822\_\_S\_KachaRahimRoad” outage alarm  Also, based on scenario, we need resolution event for these cells.  AlertGroup = ‘Cell Out of Service’  MainFlag = 1  ImpactFlag = 0  Event 6 - “LHR19822\_\_S\_KachaRahimRoad” resolution alarm  Event 7 - “LHR29822\_\_S\_KachaRahimRoad” resolution alarm  Event 8 - “LHR39822\_\_S\_KachaRahimRoad” resolution alarm  Event 9 - “LHR49822\_\_S\_KachaRahimRoad” resolution alarm  Event 10 - “LHR59822\_\_S\_KachaRahimRoad” resolution alarm   1. Insert the test events (Event 1,Event2, Event3,Event4,Event5) 2. Check the events appears in WebTop 3. Wait for Synthetic event ( x Min) 4. Check Synthetic event is generated and it should be “Site Down” 5. Check Event Journal of synthetic event should have entries of cell outage events details. 6. Check SyntheticServerSerial and Synthetic ServerName fields are populated in the original cell outage events. 7. Also, check for any existing “Environmental Alarm” for this site, they should be cleared and their details added into journal entry of synthetic event. 8. Wait for 1 minute check TTNumber is populated in the synthetic event 9. Check Incident Record in TSRM 10. Wait for a further minute and check TTNumber is populated in the original cell outage events 11. Insert the resolution events for all cells(i.e. Event6, Event 7, Event8,Event9,Event 10) 12. Check that the synthetic event is set to clear 13. Check that the Incident Record is set to resolved | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 7.3.1 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Cell Down | | | **Standard/Customization:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | Partial site down | | | |
| Test Purpose | When cell of the site alarming then policy should raise synthetic event or update it based on number of cell outage alarms to “Site Partial Down” or “Site Down”. Also, based on cell resolution alarms, it should clear synthetic event or downgrade it to “Site Partial Down”. | | | |
| Procedure | 1. Insert the test event (Event 1,Event2, Event3) 2. Check the events appeared in WebTop 3. Check that synthetic event creates and should be “Site Partial Down”. 4. Its journal has entry of all three cell outage alarm details. 5. Incident record is created 6. Check SyntheticServerSerial & SyntheticServerName fields are populated in the original cell outage alarms. 7. Also, check for any existing “Environment Alarm” for this site, they should be cleared and their details should be added into journal of synthetic event. 8. Insert remaining cell outage event (Event4,Event5) 9. Check the synthetic event should update to “Site Down” and its journal has entry of newly cell outage alarms. 10. Check that SyntheticServerSerial & SyntheticServerName fields are populated in the original cell outage alarms. 11. Check journal of synthetic event has details of both source events 12. Wait for some time (x Min) 13. Insert resolution event for some cells (i.e. Event6, Event7,Event8). 14. Check that synthetic event should be downgraded to “Site Partial Down” and its journal has entry of resolution events. 15. Insert resolution event for remaining cell (i.e. Event9, Event10) 16. Check that synthetic event should be clear and its journal has entry of resolution events. 17. Check that the Incident Record is set to resolved | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

| Test Number: 7.3.2 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Cell Down | | | **Standard/Customization:** Custom | |
| **Product:** IMPACT | | | Requirement Catalogue reference | |
| Test Title: | Cell outage alarms resolve during sleep time | | | |
| Test Purpose | When cell outage alarms occur first time then based on sleep time, if alarm resolved during that period then does not raise TT. | | | |
| Procedure | 1. Insert the test event (Event 1,Event2,Event3) 2. During sleep time, insert resolution events for cell outage (Event6, Event7 & Event8). 3. Check that synthetic event should not be created as we don’t have any cell outage alarms. | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## TSRM

| Test Number: 8.0.1 | Revision: 1.0 | | Author: Prev Datta | Date: 26/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** TSRM HA | | | **Standard/Customization:** Standard | |
| **Product:** TSRM | | | Requirement Catalogue reference: System Testing | |
| Test Title: |  | | | |
| Test Purpose |  | | | |
| Procedure | * Login onto URL <https://10.231.105.21:9043/ibm/console/logon.jsp> * Select CONTINUE TO THE WEBSITE * Type in User ID wasadmin and password * Expand the Server drop down on the left * Select Application Servers * Select the TICK BOX by MXSERVER1 * And press STOP * Confirm the Stop by pressing OK * Log onto <http://10.231.105.25/maximo/ui/login> * Go back to <https://10.231.105.21:9043/ibm/console/logon.jsp> * Select the tick box MXServer1 | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |
|  | |  | | |