**Microcloud Health Check Test**

Contents

[Introduction 1](#_Toc358715738)

[Test Name: 1320\_UCA\_Setup\_Wizard\_7.x\_auto 2](#_Toc358715739)

[Test Name: 0310\_UCA\_Ilo\_Register\_Ilo\_auto 2](#_Toc358715740)

[Test Name: 0810\_UCA\_Ilo\_Service\_Event\_auto 3](#_Toc358715741)

[Test Name: 0715\_UCA\_Ilo\_L2\_collections\_Irs\_initiated\_auto 4](#_Toc358715742)

[Test Name: 1895\_Metrics\_Disable\_Device\_auto 4](#_Toc358715743)

[Test Name: 0320\_UCA\_iLO4\_Unregister\_iLO 5](#_Toc358715744)

[Test Name: 1395\_UCA\_Unregister 5](#_Toc358715745)

[Test Name: 9700\_HealthcheckEmailer\_auto 6](#_Toc358715746)

**Author:** Mike Healy 10/6/2013

**Last Update:** Mike Healy 16/7/2013

## Introduction

ITG is a complicated test environment with many interdependencies across multiple systems. The test activities of the Microcloud project are dependent on the ITG environment having as much up time as possible. The Microcloud Health Check (MHC) is a test automation initiative to monitor ITG features at periodic intervals.

To achieve that the following automated tests have been provided which observe data propagating end to end through the ITG environment.

1. Register an IRS host (IRS – RSDC – SADB - HPSC)
2. Register a gen8 endpoint device (Endpoint - IRS – RSDC – SADB - HPSC)
3. Generate a service event (Endpoint – IRS – RSDC – SADB -HPSC)
4. Generate a collection event (IRS – RSDC – SADB - HPSC) \*
5. Generate a metric event (IRS – RSDC – SADB)
6. Disable an endpoint (Endpoint – IRS – RSDC – SADB)
7. Disable a host (IRS – RSDC – SADB)

Test execution time is circa 90 minutes but can take in excess of 120 minutes where the tests encounter multiple failures (this is to allow for test timeouts).

This document bullet points the test steps for each of the above tests. This is a living document as the test steps are liable to change on account of constant review.

*\*Note: The collection is generated using the IRS interface. However, the IRS client connects to the endpoint under test to generate the collection.*

|  |  |  |
| --- | --- | --- |
| Test Name: 1320\_UCA\_Setup\_Wizard\_7.x\_auto | | |
| **Test Step** | **Checkpoints applied** | **Additional information** |
| Test Preparation   * Import test data tables * Note test start time |  |  |
| Logon to IRS  Run the IRS registration wizard   * Web Proxy data filled * Customer details data filled * Site Details data filled * Register Host device data filled * Service providers data filled * Ensure host visibility in HPSC * Note data centre name * Note registration token from XML * Note GUID of Metric Generated with Client Registration in IRS DB | * All screens successfully data filled. * Check expected growler messages appear * Check HP Connection Status successful * Check HP support details validated * Check partner details are updated * Check Connectivity and Registered icons | Supports IRS7.0.5 and 7.0.8  Runs pgSQL\_CollEvent\_Metric\_LastGUID immediately after updating partners |
| Logon to RSDC   * Run customer search to find host device. * Note host most recent timestamp * Note host GDID | * Check host visibility is enabled * Check host timestamp has been updated * Check Partner details |  |
| Logon to SADB   * Perform device search using GDID * Note MOOS key | * Check Remote Support is enabled * Check Registration is complete * Check host timestamp has been updated * Check Partner details | Blocker:  An excessive amount of old data for pdehost9 makes it difficult for a browser to load all the events data to check the DEV\_REG\_GUID |
| Logon To HPSC with HPP\_ID   * Perform device search for the host | * Check host timestamp update * Check partner details |  |

|  |  |  |
| --- | --- | --- |
| Test Name: 0310\_UCA\_Ilo\_Register\_Ilo\_auto | | |
| **Test Step** | **Checkpoints applied** | **Additional information** |
| Test Preparation   * Import test data tables * Logon to ILO device and ensure it is unregistered * Ensure host visibility in HPSC * Note test start time | * Check ILO unregistered |  |
| Logon to IRS   * Add Gen8 device protocol * Discover Gen8 device * Override product and serial number for gen8 device * Add ILO device protocol * Discover ILO device * Note GUID of Metric Generated with Client Registration in IRS DB | * Check Gen8 discovery is successful * Check ILO discovery is successful | Runs pgSQL\_CollEvent\_Metric\_LastGUID  Discovery is verified in the IRS interface by checking that the newly discovered devices are listed in the “Devices” list, are enabled and have the correct protocol assigned to them. |
| Logon to ILO device | * Check ILO device is registered |  |
| Logon to RSDC   * Perform a device search for the gen8 device * Note Gen8 GDID | * Check Gen8 device timestamp update * Check Partner details |  |
| Logon to SADB   * Perform a device search using the Gen8 GDID * Perform device search for the endpoint * Note DEV\_REG GUID | * Check Gen8 device timestamp update * Check DEV\_REG GUID timestamp update * Check Partner details * Check SADB activity log\*\* | Add timestamp check to DEV\_REG GUID |
| Logon To HPSC with HPP\_ID   * Perform a device search for the Gen8 device | * Check Gen8 device timestamp update * Check Partner details |  |
| Logon To HPSC with ServicePartner\_ID   * Perform customer device search for the endpoint | * Check Gen8 device timestamp update |  |
| Logon To HPSC with SalesPartner\_ID   * Perform customer device search for the endpoint | * Check Gen8 device timestamp update |  |

|  |  |  |
| --- | --- | --- |
| Test Name: 0810\_UCA\_Ilo\_Service\_Event\_auto | | |
| **Test Step** | **Checkpoints applied** | **Additional information** |
| Test Preparation   * Import test data tables * Logon to ILO device and ensure it is registered * Reset ILO device * Note test start time * Generate Service event | * Check that a new Service Event is received by the IRS client by looking for a new event to appear in the IRS DB. When it is received the GUID is noted. | If a new Service Event is not received within the specified timeout (circa 8 minutes) then the test ends as we don’t have a GUID to search for in the various interfaces. |
| Logon to IRS   * Look for the event in the IRS interface under “Service Events” | * Check is visible in IRS |  |
| Logon to RSDC   * Search for service event GUID * Note the WMF case ID * Note the service event description in event activity log | * Check service event GUID is found * Check service event GUID activity log event processing\* | All entries are checked to make sure that they have a status of “OK”. Also, 2 entries in particular are looked for: “DPSCN has been queued to be sent to SADB” and “DataPackage: <GUID> has been sent to Sadb.” Where <GUID> is replaced by the event GUID. |
| Logon to SADB   * Search for service event GUID | * Check service event GUID is found |  |
| Logon To HPSC with HPP\_ID   * Search for the service event using event description and WFM case ID | * Check the service event exists   Check event description and WMF case ID |  |

|  |  |  |
| --- | --- | --- |
| Test Name: 0715\_UCA\_Ilo\_L2\_collections\_Irs\_initiated\_auto | | |
| **Test Step** | **Checkpoints applied** | **Additional information** |
| Test Preparation   * Import test data tables * Logon to ILO device and ensure it is registered * Note test start time |  |  |
| Logon to IRS   * Generate a collection by creating and running a collection schedule * Note the collection date and GUID from the IRS DB when it is received * Search for collection event in IRS interface based on its timestamp. | * Collection can be found * Check collection GUID in the IRS interface. |  |
| Logon to RSDC   * Search for event GUID | * Check GUID is found * Check GUID activity log event processing\* | All entries are checked to make sure that they have a status of “OK”. Also, 2 entries in particular are looked for: “DPSCN has been queued to be sent to SADB” and “DataPackage: <GUID> has been sent to Sadb.” Where <GUID> is replaced by the event GUID. |
| Logon to SADB   * Search for event GUID | * Check event GUID is found |  |
| Logon To HPSC with HPP\_ID | * Check timestamp update for host device |  |

|  |  |  |
| --- | --- | --- |
| Test Name: 1895\_Metrics\_Disable\_Device\_auto | | |
| **Test Step** | **Checkpoints applied** | **Additional information** |
| Test Preparation   * Import test data tables |  |  |
| Logon to IRS   * Generate a metric event by disabling then re-enabling the host device. * Note the enable metric GUID generated in the IRS DB | * Check event GUID generated successfully |  |
| Logon to RSDC   * Search for event GUID | * Check event GUID is found * Check GUID activity log event processing\* | All entries are checked to make sure that they have a status of “OK”. Also, 2 entries in particular are looked for: “DPSCN has been queued to be sent to SADB” and “DataPackage: <GUID> has been sent to Sadb.” Where <GUID> is replaced by the event GUID. |
| Logon to SADB   * Search for event GUID | * Check event GUID is found |  |

|  |  |  |
| --- | --- | --- |
| Test Name: 0320\_UCA\_iLO4\_Unregister\_iLO | | |
| **Test Step** | **Checkpoints applied** | **Additional information** |
| Test Preparation   * Import test data tables |  |  |
| Logon to IRS   * Disable the Gen8 device * Check Endpoint reports itself as disabled   (to be provided)   * Note the event GUID in the sessions xlm file * Note GUID of Metric Generated with Client Registration in IRS DB | * Check both GUIDS are the same (cross reference check). * Check the event state of the session xml file is ‘disabled’ | Runs pgSQL\_CollEvent\_Metric\_LastGUID |
| Logon to RSDC   * Search for event GUID | * Check GUID is found * Check GUID activity log event processing * Check GUID timestamp |  |
| Logon to SADB   * Search for event GUID | * Check GUID is found * Check GUID activity log event processing\*\* * Check GUID timestamp | Consider checking the DEV\_REG event also |

|  |  |  |
| --- | --- | --- |
| Test Name: 1395\_UCA\_Unregister | | |
| **Test Step** | **Checkpoints applied** | **Additional information** |
| Test Preparation   * Import test data tables |  |  |
| Logon to IRS   * Disable the Gen8 device * Note the GUID in the IRS DB * Note the GUID in the sessions xlm file | * Check both GUIDS are the same (cross reference check). * Check the event state of the session xml file is ‘disabled’ | Runs pgSQL\_CollEvent\_Metric\_LastGUID |
| Logon to RSDC   * Search for GUID | * Check GUID is found * Check GUID activity log event processing * Check GUID timestamp |  |
| Logon to SADB   * Search for GUID | * Check GUID is found * Check GUID activity log event processing\*\* * Check GUID timestamp | Consider checking the DEV\_REG event also  Blocker:  An excessive amount of old data for pdebld9 makes it difficult for a browser to load all the events data to check the DEV\_REG\_GUID |

|  |
| --- |
| Test Name: 9700\_HealthcheckEmailer\_auto |
| This is not an ITG test. It generates the MHC email.   * Logon to SADB * Capture an image of the SADB Health. * Captures an image of the QueueMonitor. * Generates and sends the MHC email. |

**\*** RSDC activity log checks the following..

(a) Results column contains only ‘OK’ result

(b) Error Code colume contains no error codes.

(c) Check that there is a component called ‘DataCollectionSender’ containing the description "DataPackage: <GUID> has been sent to Sadb." Where <GUID> is the actual event GUID.

**\*\*** Check SADB activity log checks the following...

1. The term “End Success:” is found in any message text column entries (to be updated to also look for the term “DELETE”.

The frist qtp TEST:

[Test ID: 47640](td://support_automation.ts_dev.qc1d.atlanta.hp.com/qcbin/TestPlanModule-00000000395028973?EntityType=ITest&EntityID=47640) 0200\_Delete\_HC\_Export\_Data\_auto

This is used for to delete to export data we had used before,

The export data we used is the data we used to generate the report.