



Windows Cloud Server Test Utility Documentation

Chassis Validation Utility

Date: Feb 10, 2014  
Version: V1.01

**RESTRICTED**

THIS DOCUMENT AND ANY ASSOCIATED DATA CONTAIN RESTRICTED INFORMATION THAT IS MICROSOFT PROPERTY. DO NOT DISCLOSE TO OR DUPLICATE FOR OTHERS EXCEPT AS AUTHORIZED BY MICROSOFT CORPORATION.

|  |  |
| --- | --- |
| **Revision/ Date** | **Notes** |
| **V1.0** **1/27/15** | Initial document |
| **V1.01** **2/10/15** | Added App.Config Setup Details #2 and #3  Added General Hardware Setup Detail #2 |

Contents

[1 Overview 4](#_Toc410655856)

[2 Setup 4](#_Toc410655857)

[1 General Environment Setup Diagram 4](#_Toc410655858)

[2 General Hardware Setup Diagram 5](#_Toc410655859)

[3 General Hardware Setup Details 5](#_Toc410655860)

[4 Asset Management Test Setup Details 6](#_Toc410655861)

[5 FPGA Blade Mezzanine Test Setup Details 6](#_Toc410655862)

[6 PSU Firmware Test Setup Details 7](#_Toc410655863)

[7 App.Config Setup Details 7](#_Toc410655864)

[3 Running the Utility 8](#_Toc410655865)

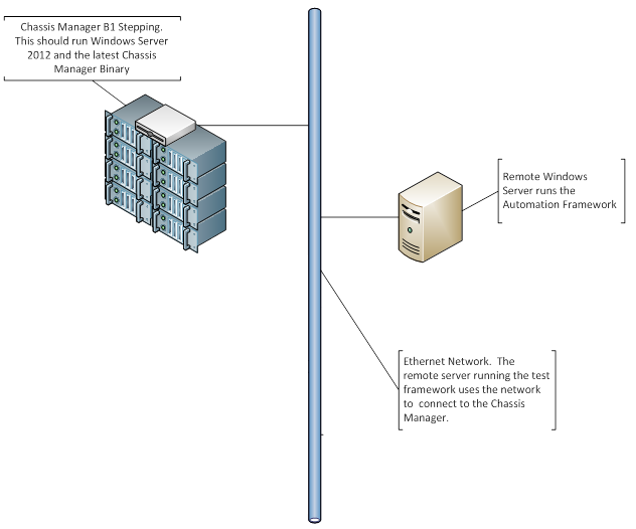
[4 Analyzing the Results 8](#_Toc410655866)

# Overview

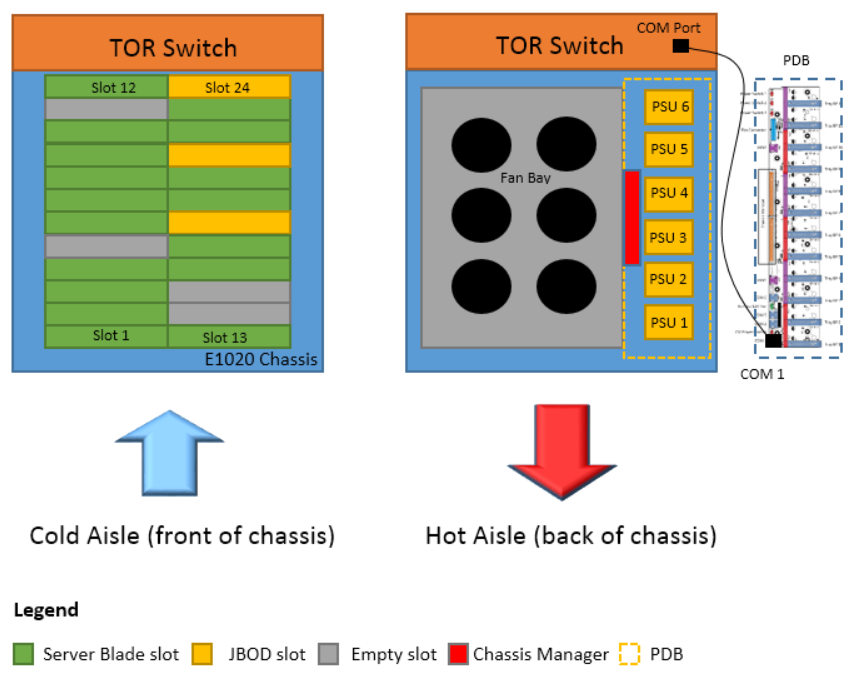
The Chassis Validation Utility is an automation utility that provides Functional Verification Testing of the Chassis Manager RESTful Web Service. The utility also tests the functionality of the Chassis Manager in its interaction with the rest of the chassis components. The utility can be run by using the provided Command Line interface.

# Setup

## General Environment Setup Diagram



## General Hardware Setup Diagram



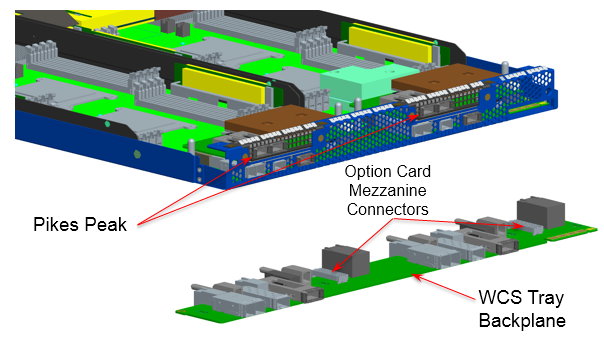
## General Hardware Setup Details

1. A functional Chassis Manager that is connected with a network connection is required.
2. Add “Admin” with password “$pl3nd1D” to be local admin in Chassis Manager under test.
3. Although the above diagram is the recommended setup, the number of server blades, JBODs, and empty slots do not have to be as shown. A minimum of 2 is required for each.
4. Server blades do not have to be connected to the network for this validation.
5. Tests “StartStopSendReceivePortSerialConsoleByAllUsersTest” and “StartStopSerialPortConsoleTest” require CAT-6 cable connected between TOR switch and COM 1 as shown in the diagram above.
6. Tests “StartStopBladeSerialSessionTest” and “StartStopSendReceiveBladeSerialSessionByAllUsersTest” require EMS enabled on all blades.

## Asset Management Test Setup Details

* 1. Test Cases include “GetChassisManagerAssetInfoTest”, “GetPdbAssetInfoTest”, “GetBladeAssetInfoTest”, “SetChassisManagerAssetInfoTest”, “SetPdbAssetInfoTest”, “SetBladeAssetInfoTest”.
  2. Chassis Manager FRU, PDB FRU, and Blade FRU all need to have MultiRecord Area Offset set to a non-zero value.
     1. FRU tools are placed in the “TestResources\FRU\Tools” folder, in case they are need to set the MultiRecord Area Offset to a non-zero value.
  3. Before running “GetChassisManagerAssetInfoTest” or “SetChassisManagerAssetInfoTest”, populate “TestResources\TestData\ChassisManagerFruSample.xml” by calling REST API GetChassisManagerAssetInfo.
  4. Before running “GetPdbAssetInfoTest” or “SetPdbAssetInfoTest”, populate “TestData\PdbFruSample.xml” by calling REST API GetPdbAssetInfo.
  5. Before running “GetBladeAssetInfoTest” or “SetBladeAssetInfoTest”, populate “TestResources\TestData\BladeFruSample.xml” by calling REST API GetBladeAssetInfo?bladeId=<aValidBladeIndex>. This only has to be done for one of the blades.

## FPGA Blade Mezzanine Test Setup Details



* 1. Test Cases include “SetBladeMezzPassThroughModeTest”, “GetBladeMezzPassThroughModeTest”, and “GetBladeMezzAssetInfoTest”.
  2. “GetBladeMezzAssetInfoTest” requires a Pikes Peak card with FRU contents populated as per Pikes Peak FRU V0.2 Specifications.
  3. FPGA Pikes Peak Blade Mezzanine attaches to the back of the WCS TBP as shown above.
  4. Verify Blade Mezzanine FPGA card is imaged. If not, use the image placed in “TestResources\Blade Mezz\Image” folder.
  5. No more than 1 or 2 Mezz cards is required for functional verification. Make sure to specify the slots in which these blades are installed in the App.Config file.

## PSU Firmware Test Setup Details

* 1. Test Cases include “UpdatePsuFirmwareTest” and “GetPsuFirmwareStatusTest”.
  2. Chassis should only be populated with either LES PSUs (Artesyn PL1600H) or Non-LES PSUs (Artesyn PS1600H) for these tests.
  3. All PSUs must have an updated bootloader that allow for APIs “UpdatePsuFirmware” and “GetPsuFirmwareStatus” to function.
  4. Sample PSU firmware image files are provided in the “PSU FW” folder.

## App.Config Setup Details

1. Files “App.Config” and “ChassisValidationUtility.exe.Config” need to have the same information.
2. Key “AdminUserName” should have value “<CmHostName>\Admin” where <CmHostName> is the Chassis Manager name under test.
3. Key “Password” should have value “$pl3nd1D”.
4. Keys “LabDomainTestUser” and “LabDomainName” do not need to be updated.
5. Server Serial Number Keys are server blade slots, and the Values are the corresponding serial numbers.
6. “JbodCount” is number of JBODs in the chassis.
7. “ProcessorCount” is number of processors per blade.
8. “DIMMsCount” is number of DIMMs per blade.
9. “PCIeCount” is number of PCIe slots per blade.
10. “NicCount” is number of MAC addresses or Nic Info objects returned per blade.
11. Key “SpecifiedBladeLocations” is only used by tests “SetBladeMezzPassThroughModeTest”, “GetBladeMezzPassThroughModeTest”, and “GetBladeMezzAssetInfoTest”. Value for this key are all server blade slots that have an attached Blade Mezzanine FPGA card (space delimited).
12. Key “PriLesFwFilePath” refers to the file path in the Chassis Manager where a Primary LES Psu Fw Image is stored. “SecLes\*” refers to the Secondary LES Psu Fw Image, “PriNonLes\*” for Primary Non-LES Psu FW Image, and “SecNonLes” for Secondary Non-LES Psu FW Image.
13. Key “InvalidPsuFwFilePath” refers to an invalid file (of different extension such as .bin) that is stored at the specified path in the Chassis Manager.

# Running the Utility

* 1. Copy binaries to the system running that will run the automation utility.
  2. Open command prompt with administrator rights.
  3. Go to the directory where binaries are located.
  4. Update App.Config accordingly.
  5. Run “ChassisValidationUtility.exe /?” to view help and options for running the utility.

# Analyzing the Results

1. Wait for all tests to finish. Expect for there to be many outputs from the command prompt.
2. The outputs of each test run are logged into a file, which has a naming convention “TestResults\_<Date>\_<Time>.txt”.
3. All failures will need to be reviewed individually (such as the example shown in red above), and the failing test scenario will need to be identified.
4. Test Cases can be rerun individually using the “-t” parameter, or they can be run in a batch using the “-b” parameter.