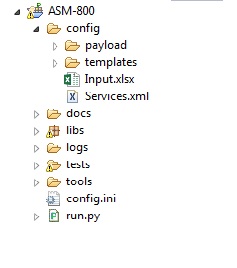
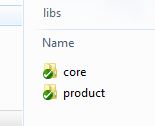
**ASM Automation Framework**

# Key Documentation:

This documents the Active System Manager Automation Framework native to Python 2.6.

# Implementation Structure:

**Fig 1.1The diagram shows the folder structure of ASM automation framework**

**config:** The ‘config’ folder contains all the files related to the configuration required for running the automation.

**Payload -** The ‘payload’ folder contains the template for payload for operations like Creating Credential, Network, Discovery, and Login etc.

**Templates -** The ‘templates’ folder contains the templates for the Deployment.

**Input.xlsx -** The Input.xls contains information about Credentials, Networks, Devices to be discovered and Test cases. Any New Device, Network or Credential to be added has to be added to the corresponding sheet in the Input.xls file.

**Services.xml -** This file contains the relative URLS for all ASM Operations.

**docs:** This folder contains all the documents related to the Automation Suite

**libs:** This folder contains two sub folders

**core –** This folder contains dellunit framework modules

**product –** This folder contains ASM related Framework modules

**requests** – This is python package used for making REST Calls

**templates** – This folder contains templates for reporting.

**BaseClass.py** – This module contains all the API for performing all ASM operations

**globalVars.py** – This module contains all the global variables which are used across automation suite.

**hooks.py** – This module contains Setup and Teardown methods to execute at automation suite level.

**utility.py** - This module contains all the methods related to the utilities

**logs:** This folder contains all the logs. A ‘report.html’ will be generated post execution of all test cases and the other logs generated while executing the test cases. The report will contain the list of test cases executed along with test result.

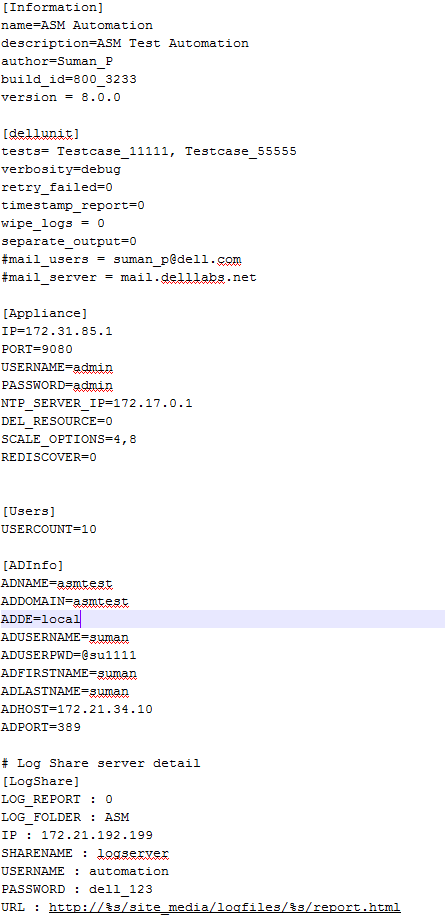
Hyperlinks “Steps” and “Execution Log” will be provided for each test case in the report page. The link “Steps” will display a list of all logical points of the test case and the status of that case. The link “Execution Log” will display a page containing all the operations that were carried out during the execution of the test case.

**tests:** The ‘tests’ folder contains all the test cases related to ASM. This folder must contain the target test files.

**Testcase\_11111.py** – This is a Pre-Requisite Test which should be included in every run. This test performs Discovery of the Devices provided in Input.xls

**tools:** The ‘tools’ folder contains tools like plink.exe and pscp.exe which are used to run remote commands on ASM appliance for enabling Debugging.

**config.ini:**



Config.ini has various sections like

**Information**: Project level Information has to be provided here

**dellunit**: Provide which tests to execute in ‘tests’ attribute.

**Appliance**: Provide Appliance Information under Appliance Section

**ADInfo**: Active Directory Information has to be provided here

**Execution**:

Download the latest code from github and copy to any Windows machine.

Launch console and change directory to the automation folder and run the following command after providing all the required inputs.

**“python run.py config.ini”**