



**Eclipse APERI Storage Management Project  
Integrated Development Verification Test Plan  
*-Phase 2-***

## Table of Contents

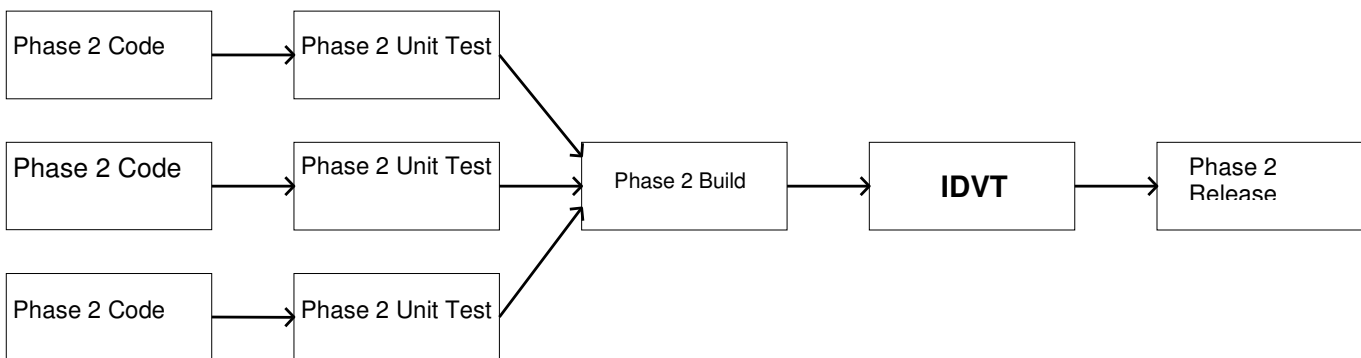
<b>1.</b>	<b><i>Introduction</i></b>	<b>3</b>
1.1	<b>Objectives</b>	<b>3</b>
1.2	<b>Scope</b>	<b>3</b>
1.3	<b>IDVT Ecosystem Overview</b>	<b>4</b>
1.4	<b>IDVT Entry Criteria</b>	<b>4</b>
1.5	<b>IDVT Exit Criteria</b>	<b>4</b>
1.6	<b>Domains of Testing</b>	<b>5</b>
1.7	<b>Device Support</b>	<b>Error! Bookmark not defined.</b>
<b>2.</b>	<b><i>Approach</i></b>	<b>6</b>
2.1	<b>Testing</b>	<b>6</b>
2.1.1	Acceptance Testing	6
2.1.2	Performance Testing	6
2.1.3	Test Automation	7
<b>3.</b>	<b><i>Environmental Requirements</i></b>	<b>8</b>
3.1.1	Hardware	8
3.1.2	Software	8
3.1.3	Tools	8
3.1.4	Risks and Assumptions	8
<b>4.</b>	<b><i>Status Reporting</i></b>	<b>10</b>
4.1	<b>Status Reporting</b>	<b>10</b>
<b>5.</b>	<b><i>IDVT Test Cases</i></b>	<b>11</b>
5.1	<b>Phase 2</b>	<b>11</b>
5.1.1	Required Environment	11
5.1.2	Required Staff	11
5.1.3	Target Dates	11
5.1.4	Test Cases	11
5.1.4.1	Install and Configuration Tests	11
5.1.4.2	GUI Common & Base Tests	12
5.1.4.3	Disk	12
5.1.4.4	Agent	13
5.1.4.5	CIMOM/Subsystem Discovery	13
5.1.4.6	Fabric Alerts and Alert Logs	14
5.1.4.7	Topology	14
5.1.4.8	Tape Manager	16
5.1.4.9	Reporting	17
5.1.4.10	Help	18

## 1. Introduction

This document specifies the Integrated Development Verification Test (IDVT) Plan for APERI, Phase 2.

This IDVT is performed by development after all unit testing is completed for Phase 2. IDVT concentrates on “good path” testing. Summarized test results are tracked via a document in the APERI External Development wiki. Individual test cases are tracked via TTT (Test Tracking Tool) – an external Lotus Notes database where individual test cases are defined.

IDVT for APERI is performed as follows:




---

### 1.1 Objectives

The purpose of this IDVT is to test the end-to-end integration of APERI components and to verify that all phase two requirements are completed with only a limited number of minor bugs.

IDVT will test the Aperi Storage Manager as a complete application. This includes functions of the platform layer, such as monitoring and discovery; as well as application layer functions, such as reporting and zone control. IDVT will not directly test the extensibility or componentization, implemented in phase two, product vendors will eventually leverage to create features, plug-ins, or plug-in frames.

---

### 1.2 Scope

IDVT will cover the Aperi Storage Manager new component architecture and its new Eclipse RCP based user interface. Some effort will also be made to test and benchmark Aperi’s performance.

Normally, IDVT would occur when the entire stage code is completed, unit tested, and checked into the repository. However, because the user interface implementation differs from the other phase two tasks, testing for the GUI will be a moving target. The functionality of Aperi will be tested using the legacy GUI that was included in the initial contribution. Testing of the new Eclipse RCP based GUI will begin once the testing of functionality for Aperi Storage Manager is completed, and will extend until the GUI design iterations are finished. The GUI test cases will be susceptible to changes, and will be defined in more detail as the design of the GUI stabilizes.

Development of the UI is based on an iterative development model, as opposed to the waterfall approach being used to componentized the servers and agent. Much of the details of the UI tests will develop as the UI itself develops. The legacy GUI will be used for the first part of IDVT, therefore, many of the tests will appear to be

regression tests, following some of the same testing performed for the initial contribution. Additionally, some of the same functionality tested earlier will again get re-tested using the new UI.

---

## 1.3 IDVT Ecosystem Overview

For most test situations, the stage build is installed on existing acceptance test hardware, and IDVT test cases are performed in the acceptance test environment. In some cases, access to specialized hardware devices may require IDVT to be performed in a different site.

IDVT test cases are normally executed by members of the Aperi development team.

Defects found during IDVT will be logged in Bugzilla at Eclipse.

---

## 1.4 IDVT Entry Criteria

IDVT for phase 2 can be broken into two categories – functionality testing and user interface testing. The IDVT entry criteria differ between the two.

The entry criteria for the first part of IDVT (functional testing) are:

- The IDVT Plan is approved by all mandatory approvers.  
For a list of the mandatory approvers, see [http://wiki.eclipse.org/index.php/Talk:Aperi/Test/Test\\_Plans/IDVT](http://wiki.eclipse.org/index.php/Talk:Aperi/Test/Test_Plans/IDVT)
- The phase 2, code (not including UI changes) is checked into the CVS repository.
- If special workarounds are required, they are specified in a handover document provided by the appropriate development area owner.
- The IDVT build passes an acceptance test (see below for details of the acceptance test).

The entry criteria for part two (GUI test) are:

- Part one (functional testing) of phase 2 IDVT is completed and meets the exit criteria.
- The “new” GUI is integrated into the Storage Manager and is being built as part of the nightly build.

Any deviation from these entry criteria requires the approval of the Release Manager.

Code review is desirable but not required by this process. Successfully exiting one phase is **not** an entry criterion for entering the next phase.

Specific environmental requirements are identified below.

---

## 1.5 IDVT Exit Criteria

The IDVT exit criteria for this phase are:

- 100% of IDVT test cases attempted
- 95% of all IDVT test cases passed
- No severity 1 or 2 defects or viable workarounds provided for any severity 1 or 2 defects.

- A “handover document” is produced to describe any manual procedures, workarounds, or limitations that the community needs to be aware of for this phase. This includes configuration instructions.

Any deviation from these exit criteria requires the approval of the Release Manager, Development Lead, Lead Architect, and the Test Lead.

---

## 1.6 Domains of Testing

IDVT testing emphasizes use of the GUI and (externally visible) CLI. These interfaces are used to test functions as the user would execute them. No special testing at the API level is performed during IDVT. It is expected that the nucleus of the IDVT test cases is the GUI and test cases.

In addition, install/configuration testing is performed as part of IDVT.

IDVT concentrates on the following scenario:

- All servers located on a single machine
- Windows platform
- Derby as repository
- Fresh install
- Limited number of tested hardware devices

Some IDVT test cases specifically address the limitations of this scenario; that is, there will be some cases to test:

- Linux platform
- Replacing the Derby repository with DB2

IDVT does not thoroughly test all supported hardware devices. Since IDVT concentrates on “good path” testing, any devices beyond the core set we concentrate on are not required to be tested by IDVT if they would exercise the same code paths as tested devices. But any hardware device that requires its own code path is tested during IDVT.

IDVT does not include publications, globalization, or accessibility.

## 2. Approach

---

### 2.1 Testing

#### 2.1.1 Acceptance Testing

In order to insure that IDVT starts as scheduled, at least two comprehensive acceptance tests will be performed the week before IDVT is scheduled to begin. Acceptance tests (smoke test) will be conducted to determine whether or not Aperi Storage Manager satisfies basic requirements. Successful completion of the final test will allow for entry into the more detailed integration development verification testing.

A successful attempt of all of the following tasks constitutes a successful acceptance test:

1. Use of the `cfgaperi` script to install and configure the Data Server, Device Server, Data Agent, and the GUI.
2. Verification that the Data Agent starts and is able to successfully register with the Data Server.
3. Verification that the GUI comes up with no noticeable problems.
4. The initial agent probe works.
5. No errors appear in the Data Server log.
6. No errors appear in the Device Server log
7. Verification that the Device Server is up via the web page: <http://localhost:9000/ServiceManager>
8. Verification that the Device is running via the GUI.
9. Verifying discovery works
10. Verifying default OS ping job executes
11. Verify the default OS probe job executes
12. Verifying out of band fabric discovery works with no errors in the log files
13. Verifying discovered subsystems appear via the Topology Viewer
14. Verifying discovered volumes appear via the Topology Viewer
15. A random run through of various reports (system reports, Disk Manager reports, Data Manager reports) to verify their accessibility.

#### 2.1.2 Performance Testing

Some testing will be conducted to ensure that Aperi Storage Manager response time and scalability performs to expectations. Several performance issues were discovered in the initial contribution of Aperi. The most notable being derby performance issues.

The following results indicates some of the performance issues discovered in reporting while testing the initial contribution:

*Disk Manager*

- >Reporting
  - >Storage Subsystems
    - >LUN to HBA assignment
    - >By Storage Subsystem
    - >By Capacity

*It may take 2 minutes to finish the report for a total 123 discovered volumes  
 It may take 3 minutes to finish the report for a total 781 discovered volumes  
 It may take 7 minutes to finish the report for a total 1378 discovered volumes*

*The second one is to generating report for Aperi Storage Manager*

- >My Reports
  - >System Reports
    - >Fabrics
      - >Port Connections
      - >SAN Assets(Connected Devices)

*It also may take a while to get the report ( 2 mins for a 20 connected devices )*

In an attempt to improve Derby performance, several database changes were implemented in phase 2. One change that appeared to make a significant impact was increasing the heap size to 1 GB. IDVT will continue to test for database performance issues. Some of these test will include:

- Testing reporting in phase 2 against reporting in the initial contribution
- Testing the topology viewer in phase 2 against the topology viewer in the initial contribution
- Comparing tasks in Aperi Storage Manager against similar tasks in the initial contribution
- Using DB2 in place of Derby and comparing the differences
- Possibly trying to fine tune the heap size

In addition to the above tests the SAN Simulator will be utilized to create a baseline that will be used to compare Aperi builds against a “known good”.

Some other desirable performance tests include:

- Testing discoveries with a large number of storage subsystems
- Testing discoveries with a large number of volumes in a storage subsystem
- Testing discoveries with a large number of agents
- Testing probes on a large number of file systems
- Testing control operations

### 2.1.3 Test Automation

Several software testing automation packages were researched in an effort to try and add automation testing to Aperi. The desire is to add an open source, multi-platform, automation to an Aperi test-bed. Abbot is an open source, GUI test framework, which lets you programmatically drive UI components. The plan is to use the Abbot framework to create a set of tests that will emulate the acceptance test (smoke test) and can be included as part of the build verification.

## 3. Environmental Requirements

This section lists both the necessary and desired properties required for the test environment.

### 3.1.1 Hardware

- Two Windows servers to host APERI server components
- One Windows machine, without an HBA, running the Data and Fabric agents (and not running the server)
- One Windows machine, with an HBA, running the Data and Fabric agents (and not running the server)
- One Linux machine running APERI server and agent components
- One NAS server
- One tape device
- At least one subsystem visible to test environment
- Brocade switch
- McData switch
- Cisco switch
- A large number of storage subsystems (desired for performance testing)
- Storage subsystems with a large number of volumes (desired for performance testing)
- A large number of hosts (desired for performance testing)

### 3.1.2 Software

- Abbot for automation testing
- DB2 to test as a replacement to derby

### 3.1.3 Tools

- SAN Simulator for performance testing
- Tape CIMOM

### 3.1.4 Risks and Assumptions

The approach being used to design and code the new GUI will

- **Performance** - In some cases it will be difficult to create an environment required to emulate some of the desired performance tests. For example, testing with an actual large number of subsystems or a large number of agents will not be possible due to the limited hardware available. Efforts will be made to simulate environments for performance tests where the actual physical environment will not be possible to create.
- **GUI** – The new Eclipse RCP based GUI poses the greatest challenge for IDVT. As the design of the GUI develops and the coding progresses, the IDVT plan will be amended to account for the expected metamorphosis the UI will undergo.





- **Automation** – Since the desired automation testing is user interface driven, and the design of the new Aperi GUI is an ongoing process that will undergo many changes, it will be a huge challenge to create automation tests that will not require constant maintenance. We therefore have two options for creating automation tests:
  1. Use the legacy GUI to drive automation tests.
  2. Wait until the design of the “new” GUI is stable before creating automation tests.

## 4. Status Reporting

---

### 4.1 Status Reporting

Testing progress will be formally reported during the weekly Status Meetings. The status will include all critical defects and all defects that impede the completion of IDVT.

A status report will be prepared by the test team to facilitate the meeting and inform management of IDVT status. This report will be posted on the Aperi external wiki by the end of each Friday until the exit of IDVT. The report will contain the following information:

#### Test Cases

- The projected number of exposed test cases for each week in IDVT
- The projected number of successful test cases expected each week
- The actual number of exposed test cases each week
- The actual number of successful test cases each week
- The number of test cases being blocked due to defects

#### Defects

- The number of open defects
- The number of defects being worked on
- The number fixes to defects waiting to be verified
- The number of transferred defects
- The number of closed defects
- The number of returned defects
- The number of canceled defects
- The number of defects categorized by severity
- The number of open and working severity 1 and severity 2 defects

## 5. IDVT Test Cases

---

### 5.1 Phase 2

*Note: Additional scenarios to test the “new” GUI will be added as the design of the GUI is solidified.*

#### 5.1.1 Required Environment

- One Windows server to host APERI server components
- One Windows machine running the Data agent (and not running the server)
- One Linux machine running APERI server and agent components
- At least one subsystem visible to test environment

#### 5.1.2 Required Staff

Two IDVT testers.

#### 5.1.3 Target Dates

- November 2, 2006 – start testing (part 1) using the old legacy GUI
- December 21, 2007 – end testing (part 1) using the old legacy GUI
- December 22, 2006 – start testing (part 2) using new Eclipse RCP based GUI
- March 16, 2007 – end testing (part 2) using new Eclipse RCP based GUI

Note: Testing of the new Eclipse RCP based GUI will be dependent on the iterative development process of the UI, and therefore, the testing of this piece will itself be an iterative process.

#### 5.1.4 Test Cases

##### 5.1.4.1 Install and Configuration Tests

Source: Initial Contribution

IDVT installation and configuration for the Initial Donation will be done using the Aperi build procedure and will be used to verify that the Data Server, Device Server, Data Agent, Fabric Agent, database, and GUI are all configured and started without errors.

- Install Data Server, Device Server, Data Agent, Fabric Agent, database and GUI all on a single windows machine.
- Install Data Server, Device Server, Data Agent, Fabric Agent, database and GUI all on a single LINUX machine.

- Verify the agents running on one machine can connect and communicate with a Data Server running on a separate machine.
- Install all components using “easy” mode.
- Install all components using “regular” mode.
- Install the GUI, by itself, on a separate machine then the servers.

#### 5.1.4.2 GUI Common & Base Tests

Source: Aperi GUI Development Design Document, Aperi Internal Development Executive Review with Functional Scope plan

- Test all nodes in the tree correspond to the new Aperi node structure (six top-level nodes – “Administrative Services”, “Aperi Storage Manager”, “Data manager”, “Disk Manager”, “Fabric Manager”, “Tape Manager”).
- Test that all subnodes correspond to the agreed donation items and are functioning properly.
- Verify that all references to legacy names are removed.
- Test that jobs (discovery, probe, ping) appear beneath the correct subnode.
- Test that you are allowed to save jobs (discovery, probe, ping) under different names with different types.
- Test that “Do you want to save your changes” dialog gets displayed when a user opens another report.
- Test CIMOM jobs appear beneath the correct discovery subnode

#### 5.1.4.3 Disk

Source: IDVT Initial Contribution test plan

- Subsystems
  - Verify Storage Subsystems window
  - Verify Subsystem Details for a subsystem
- Volumes
  - Verify Volumes window
    - Filter by host
    - Filter by array
    - No filtering
  - Verify Volume Details
  - Create a volume, with and without Fabric integration
  - Delete a volume
  - Change a volume’s host assignments
- VDisks
  - Verify VDisks window
    - Filter by host

- Filter by array
  - No filtering
- Verify VDisks Details
- Create a vdisk, with and without Fabric integration
- Delete a vdisk
- Change a vdisk's host assignments
- Show Servers Affected by A Storage Subsystem Alert
  - Launch the Alert panel by selecting “Alerting->Storage Subsystem Alerts” or “Alert->Alert Log->Storage Subsystem”.
  - Click “Show affected hosts and Applications”

#### 5.1.4.4 Agent

Source: Aperi Agent Development Design Document, IDVT Initial Contribution test plan

- On the “Out of Band Fabric” panel, add an agent that can be contacted and one that cannot, see that they are updated correctly.
- On the “Out of Band Fabric” panel, remove an agent and see that it is removed from the GUI.
- On the “Out of Band Fabric” panel, add advanced properties to an agent and check that it appropriately then can receive sensor and events
- On the “Out of Band Fabric” panel, change the advanced properties and see that the change is reflected in the UI.
- Manually add a NAS server and verify it is added and can be probed.
- Test that a NAS server can be discovered and then probed.
- Verify the NAS server appears in reports.
- Verify remote scripts can be executed on the agent.
- Test that outband agents can be discovered and probed.

#### 5.1.4.5 CIMOM/Subsystem Discovery

Source:

- Run CIMOM discovery; create Probes against discovered subsystems; verify that Probes work.
- Manually add a CIMOM and test that it gets connected.
- Test a scheduled CIMOM discovery.
- Test storage subsystem discovery and that the subsystem is displayed in the topology viewer.

### 5.1.4.6 Fabric Alerts and Alert Logs

Source: IDVT Initial Contribution test plan

- Verify that alert panels can be produced for Fabric alerts and switch alerts.
- Test for Cisco blade switch alerts.
- Verify change alerts for McData switches.
- Verify switch\_property\_change alert for Brocade, Cisco, and McData switches.
- Test fabric connection state changes.
- Verify that the selection list used to determine which entities may be chosen for an alert definition is correct for each of the three alert categories.
- Verify that selections can be chosen and that alert definitions can be saved.
- Verify that nodes exist for viewing alert logs.
- Verify the contents of the alert log panels for each of the alert logs.
- Verify that alert log entries can be cleared for each alert log.
- Verify that alert log entries can be deleted for each alert log.
- Verify that details can be viewed for entries in each of the alert logs.

### 5.1.4.7 Topology

Source: IDVT Initial Contribution test plan

Note: no support for details panels, launch element manager, or tape libraries in this stage. Fabric discovery only in this stage.

Common Features

- Minimap
- Navigation
  - Dragging the mouse over the minimap reposition the view.
  - Clicking outside the view centralize the view.
- Manipulation
  - The minimap can be repositioned over the screen.
  - Representation of the minimap must be synchronized with current opened view.
    - Minimap can be minimized and resized.

Overlays

- Health Overlay
  - Entity status
    - Shown by elements which are composed of other entities (or no other entity at all), however all of the entities are visible on the view (or belong to a group which is visible).
  - Aggregated status
    - Shown by elements that contain other entities, and this entities are hidden either because the group is collapsed or the other entities can not be shown. For example, collapsed groups (host groups, subsystems groups, file systems, volumes groups, etc). Groups elements do not have their own status, consequently the aggregated status is shown for that entities.

- Colors
  - The use of colors, as described in the external SDD1, applies to connections, as well as elements when health overlays are turned on.

## Groups

- Elements can be logically grouped, as defined in the external SDD1.
- When groups are collapsed, the number of elements of a particular type, as well as the number of selected elements is shown on the representation of each group.
- Groups can be collapsed/expanded through context menu or by click in the upper right corner.
- Groups can be sorted, and the elements within a group can be regrouped into multiple groups depending on the grouping algorithm.

## Navigation

- On the Graphical View
  - Refreshing the screen
  - Opening/Closing a new tab
  - Expanding/Collapsing groups
- On the Tabular View
  - Locate elements on the tabular view
    - Elements meeting the criteria are highlighted on the graphical view as well as on the tabular view
  - Sorting
    - Elements can be sorted within groups, or in a particular column
- User defined properties
  - The user can specify user defined properties for elements such as:
    - Switch, hosts, subsystems
  - Changing user defined properties column names for a type of entity
  - Changing the value associated to a user defined property of a particular element
  - Grouping by
    - Grouping on some entity groups can be done by:
      - Predefined grouping methods that apply to specific entities
      - User defined properties
- Navigation on the Graphical View
- Selecting an element
  - Elements can be added/removed to/from an existing selection group.
  - Group of elements can be selected, selecting all the entities contained on them.
  - Rubber band can be used to add multiple elements to the selection.
  - Selecting a row in the tabular view refocus the view on the selected element
  - Selecting an element on the graphical view selects the element on the tabular view
- Sorting
  - Sorting can be applied on elements within a group

- For details about sorting algorithm are covered on the external SDD1
- Grouping by
  - Predefined grouping methods that apply to specific entity types
  - Grouping by user defined properties
- Printing and print preview
  - The print option provides the mechanism to print the existing view
- Remove Missing entity
  - Through the context menu, entity marked as missing can be removed from the database. See below on Selective Delete Missing.
- Synchronization of tabular view and graphical view
- Selection of elements
  - See *selecting an element* on section “*Navigation on the Graphical View*”
- Highlighting elements
  - Identifying elements directly associated to the element on focus, such as zones, zone sets.
  - Physical or logical connections from a particular element can be shown on demand by the user by using the ALT+CLICK operation.
  - Locate operation elements on the tabular view, or selecting zones/zone sets highlight elements on the current view
  - Highlights on the elements can be removed
- Pin list
  - Elements can be added/removed from the pin list
  - Elements on the pin list can be sorted
- Refreshing the graphical/tabular views
  - Adjusting the refresh interval
  - Auto-refresh operation in background
    - Refresh a view, where the focus element has been removed from the database.

#### Specific Views

Include testing of one example of each kind of view:

- SAN L0
- SAN L1
- SAN L2
- Computer L0
- Computer L1
- Computer L2

#### 5.1.4.8 Tape Manager

Source: IDVT Initial Contribution test plan



- Perform tape discovery. Expand “Tape Manager/Tape Libraries” node. Verify list of displayed tape libraries and attributes.
- Probe a tape device.
- Launch Element Manager.
- Verify a tape library gets displayed in the Topology Viewer.
- Click on “Details” button for displayed libraries. Verify data displayed.
- Click on “Launch Element Manager” button for selected library. Verify the correct element manager is launched.
- Click on “Drives” button. Verify displayed information is correct for selected library.
- Click on “Media Changers” button. Verify displayed information is correct for selected library.
- Click on “Frames” button. Verify displayed information is correct for selected library.
- Click on “Cartridges” button. Verify displayed information is correct for selected library.
- Click on “I/O Ports” button. Verify displayed information is correct for selected library.
- Click “Remove” button. Verify the library is removed from both the tape libraries table as well as from the corresponding database table.

#### 5.1.4.9 Reporting

Source: Brian Delaire, IDVT Initial Contribution test plan

- **Batch Reporting of Charts**
  - Batch Reporting of Graphs
    - Verify batch reporting of graphs on disk capacity report in HTML format
    - Verify batch reporting of graphs on computer uptime report in HTML format
    - Verify batch reporting of graphs on access time report in HTML format
    - Verify batch reporting of graphs on network-wide ping report in HTML format
    - Verify batch reporting of graphs on filesystem capacity report in HTML format
    - Verify batch reporting of graphs on disk capacity report in PDF format
    - Verify batch reporting of graphs on computer uptime report in PDF format
    - Verify batch reporting of graphs on access time report in PDF format
    - Verify batch reporting of graphs on network-wide ping report in PDF format
    - Verify batch reporting of graphs on filesystem capacity report in PDF format
    - Verify batch reporting of graphs on file type report in PDF format
- **Data Manager Reporting**
  - Reporting
    - Verify Assets -By Computer -one agent
    - Verify Assets\_ByComp\_Controller for one agent
    - Verify Assets\_By Comp\_Disks for one agent
    - Verify Assets\_By Comp\_Exports/shares for one agent
    - Verify Assets\_By Filesystem for one agent
    - Verify Assets\_By Sys Wide\_Users for one agent
    - Verify Assets By OS Type for one agent
    - Verify Availability\_Ping for one agent reports correct.
    - Verify Capacity\_Disk\_Capacity for one agent report correct
    - Verify Capacity\_FS\_Capacity for one agent report correct
    - Verify Capacity\_FS\_FreeSpace for one agent report correct
    - Verify Capacity\_FS\_UsedSpace for one agent report correct

- Run scan with only "Largest Files" profile, verify only that profile is reported - 1 Agent
- Verify UsageViolation\_ConstraintViolations are reported correctly.
- 

Note: Additional reporting tests will be added to the TTT database.

#### 5.1.4.10 Help

Source: Chris King, IDVT Initial Contribution test plan

- Verify the link between "help" and function is working by pressing F1 for various items.
- Ensure that the correct help topic is appearing for the product window that invoked the help. checking the links between the entries in the help's table of contents and the actual topics in the help. For example, pressing F1 from the Data Manager > Monitoring > Groups >Computer > Create Computer group window, should cause the appropriate help topic to appear.
- Verify links from one help topic to another. For example, does the link that appears in the Data Manager > Monitoring > Groups >Computer > Create Computer group > help window correctly link to the Monitoring - Groups window?
- Check the links between the entries in the help's table of contents and the actual topics in the help.
- Test the Back, Forward, and print buttons in the tool/menu bar.
- Verify the cheat sheets: Ensure that the correct cheat sheet appears for the product window, check any links within the cheat sheet to confirm that they link correctly, etc.