# Eclipse APERI Storage Management Project
# Integrated Development Verification Test Plan
## *-Phase 3-*

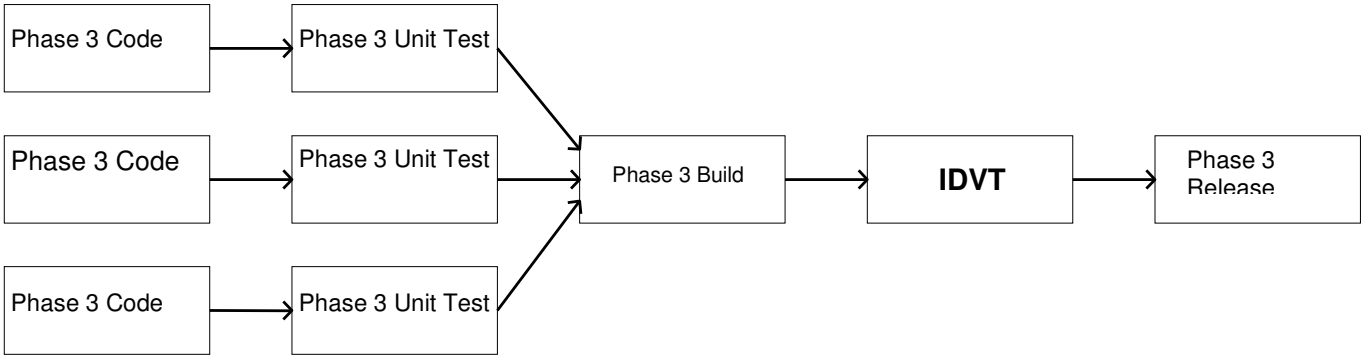
# Table of Contents

# 1. Introduction

This document specifies the Integrated Development Verification Test (IDVT) Plan for APERI, Phase 3. It identifies the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources required to complete testing, and the risks associated with this plan.

This IDVT is performed by development after all unit testing is completed for Phase 3. IDVT concentrates on "good path" testing. Summarized test results are tracked via a document in the APERI External Development wiki (http://wiki.eclipse.org/images/2/20/Aperi_IDVT_Status%2C_Phase_3_-_Phase_3_IDVT_Status.pdf). Individual test cases are tracked by the test team using an external database, where individual test cases are defined in detail. A complete summary of the test cases are listed in this document.

IDVT for APERI is performed as follows:




## 1.1 Objectives

The purpose of this IDVT is to test new features added to APERI since the phase 2 release and to verify that all phase three requirements are completed with only a limited number of minor bugs.

Testing includes, but is not limited to the following items:

- APERI's new RCP Framework, including the new login panel, the toolbar, buttons, menus, and help.
- The RCP encapsulation of the legacy GUI (classic view), including popup dialogs and reports.
- The BIRT reporting tools, including the Report Designer and the Report Viewer.
- The integration of new functions and features into APERI.
- Performance
- Help – content and branding

IDVT will test the Aperi Storage Manager as a complete application. This includes functions of the platform layer and application layer. IDVT will not directly test any of the extensibility or componentization product vendors will eventually leverage to create features, plug-ins, or plug-in frames.

## 1.2 Scope

IDVT will cover new features migrated into the Aperi Storage Manager and its new Eclipse RCP based user interface. Some effort will also be made to test and benchmark Aperi's performance.

Phase 3 development called for the encapsulation of the Legacy UI into the RCP GUI as a new perspective/view. This encapsulated *Classic View* will replace the Legacy GUI in the final product (at R3). Most of the underlying code will be unchanged, so phase two testing completed on the Legacy GUI will, theoretically, not be invalidated, but regression testing will be required. Along with this change, the RCP based topology viewer moves out of R3 as a formal deliverable (since there will be a topology viewer in the classic view). Therefore, the key new functionality to focus on for the R3 release will be the RCP encapsulated Classic View, the BIRT based report viewer, the BIRT based report designer, and the handful of reports that will be implemented at that time.

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

The entry criteria for Phase 3 of IDVT are:

- The IDVT Plan is approved by the Project Lead.
- The phase 3 user interface code 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).
- Phase 2 IDVT is completed and meets exit criteria
- The "new" GUI can be used with the Storage Manager and is being built in a nightly build.

Any deviation from these entry criteria requires the approval of the Project Lead.

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 ‘blocked’, ‘critical’ or ‘major’, severity defects, or viable workarounds provided for any ‘blocked’, critical’ or ‘major’ severity 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 Project Lead, and the Test Lead.

## 1.6 Domains of Testing

IDVT testing emphasizes use of the new RCP GUI. This interface will be 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 will be the GUI.

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. Download and install the RCP GUI build.
2. Use of the cfgaperi script to install and configure the Data Server, Device Server, Data Agent, and the GUI.
3. Verification that the Data Agent starts and is able to successfully register with the Data Server.
4. Verification that the GUI comes up with no noticeable problems.
5. The initial agent probe works.
6. No errors appear in the Data Server log.
7. No errors appear in the Device Server log
8. Verification that the Device Server is up via the web page: http://localhost:9000/ServiceManager
9. Verification that the Device is running via the GUI.
10. Verifying discovery works
11. Verifying default OS ping job executes
12. Verify the default OS probe job executes
13. Verifying out of band fabric discovery works with no errors in the log files
14. Verifying discovered subsystems appear via the Topology Viewer
15. Verifying discovered volumes appear via the Topology Viewer
16. 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. In some cases, mainly reporting, the new RCP GUI will run side by side with then legacy GUI in order to benchmark the performance of the RCP GUI.

### 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 Linux machine running APERI server and agent components
- At least one subsystem visible to test environment
- Brocade switch
- McData switch
- Cisco switch

### 3.1.2 Software

- DB2 to test as a replacement to derby

### 3.1.3 Risks and Assumptions

**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.

- **Performance** - In some cases it will be difficult to create an environment required to emulate some of the desired performance tests. For example, create volume in large configurations 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.

# 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 3

### 5.1.1 Required Environment

- Two Windows server to host APERI server components
- 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

- January 26, 2007 – start IDVT testing
- March 16, 2007 – end IDVT testing

### 5.1.4 Test Cases

#### 5.1.4.1 Install and Configuration Tests

Source: Hans Lin, Brian Delaire

IDVT installation and configuration for Phase 3 will be done using the Aperi build procedure and will be used to verify that the Data Server, Device Server, Data Agent, Fabric Agent, and database are all configured and started without errors. The RCP GUI is considered an independent application, and will therefore have its own install, independent of the current server/GUI install.

- Install Data Server, Device Server, Data Agent, Fabric Agent, database and RCP GUI all on a single windows machine.
- Install Data Server, Device Server, Data Agent, Fabric Agent, database and RCP GUI all on a single LINUX machine.
- Install the GUI, by itself, on a separate machine then the servers.
- Install all components using “easy” mode.
- Install all components using “regular” mode.
- Verify the RCP GUI running on one machine can connect and communicate with a Data Server running on a separate machine.

#### 5.1.4.2 RCP GUI - Classic View

Source: Hans Lin, Brian Delaire

The following items will be tested using the new RCP GUI

- Storage Manager login popup panel
- Menu bar “File” button
- Menu bar “View” button
- Menu bar “Preferences” button
- Menu bar “Help” button
- General “Print Branch pop-up panel
- Test Administrative Services -> Agents -> CIMOM “Add CIMOM” pop-up dialogue
- Test Administrative Services -> Agents -> Out of Band Fabric “Add” pop-up dialogue
- Test Administrative Services -> Configuration -> Role-to-Group mappings pop-up dialogue
- Test Administrative Services -> Configuration -> Probe Agent Administration pop-up dialogue
- Test Administrative Services -> Configuration -> Manager Element Manager pop-up dialogue
- Test Administrative Services -> Configuration -> Manage NAS Server pop-up dialogue
- Test Administrative Services -> My Reports -> Batch Reports – Create batch report pop-up dialogue
- Test Data Manager -> Monitoring -> Group -> Computer – Create computer group pop-up dialogue
- Test Data Manager -> Monitoring -> Group -> Filesystem – Create filesystem group pop-up dialogue
- Test Data Manager -> Monitoring -> Group -> Ping – Create Ping group pop-up dialogue
- Test Data Manager -> Alerting -> Computer Alerts -> Create computer alert pop-up dialogue
- Test Data Manager -> Alerting -> Filesystem Alerts -> Create filesystem alert pop-up dialogue
- Test Data Manager -> Alerting -> Scheduled Actions -> Scripts -> Create script pop-up dialogue
- Test Data Manager -> Reporting -> Groups -> Computer - Create computer group pop-up dialogue
- Test Data Manager -> Reporting -> Groups -> Filesystem - Create filesystem group pop-up dialogue
- Test Disk Manager -> Monitoring -> Groups -> Storage Subsystems - Create storage subsystem group pop-up dialogue
- Test Disk Manager -> Alerting -> Storage Subsystems Alerts - Create storage subsystem group alert pop-up dialogue
- Test Fabric Manager -> Monitoring -> Groups -> Fabric - Create fabric group pop-up dialogue
- Test Fabric Manager -> Alerting -> Fabric Alerts - Create fabric alert pop-up dialogue
- Test Fabric Manager -> Alerting -> Swtich Alerts - Create switch alert pop-up dialogue
- Test Fabric Manager -> Alerting -> Endpoint Device Alerts - Create EP device alert pop-up dialogue
- Test Tape Manager -> Monitoring -> Groups -> Tape Library - Create tape library group pop-up dialogue
- Test Menu bar “ BACK” button
- Test Menu bar “FORWARD” button

- Test Menu bar “SAVE” button
- Test Menu bar “Close the Current Panel” button
- Test Menu bar “Stop the Current Server Request” button
- Aperi Storage Manager -> My Reports -> System Reports -> Data -> Disk Capacity
- Aperi Storage Manager -> My Reports -> System Reports -> Data -> Disk Defects
- Aperi Storage Manager -> My Reports -> System Reports -> Data -> Storage Availability
- Aperi Storage Manager -> My Reports -> System Reports -> Data -> Storage Capacity
- Aperi Storage Manager -> My Reports -> System Reports -> Data -> Total Freespace
- Aperi Storage Manager -> My Reports -> System Reports -> Fabric -> Port Connections
- Disk Manager -> Reporting -> Storage Subsystems -> Computer Views -> By Computer
- Disk Manager -> Reporting -> Storage Subsystems -> Storage Subsystem Views -> By File Systems
- Disk Manager -> Reporting -> Storage Subsystems -> Storage Subsystem Views -> By SS
- Disk Manager -> Reporting -> Storage Subsystems -> Storage Subsystem Views -> By LUN
- Disk Manager -> Reporting -> Storage Subsystems -> Storage Subsystems Views -> Disks
- Disk Manager -> Reporting -> Storage Subsystems -> LUN to HBA Assignment -> By SS
- Disk Manager -> Reporting -> Storage Subsystems -> LUN to HBA Assignment -> By Capacity
- Disk Manager -> Reporting -> Storage Subsystems -> LUN to HBA Assignment -> with no data agent
- Disk Manager -> Reporting -> Storage Subsystems -> LUN to HBA Assignment -> with not visible data agent

#### 5.1.4.3 BIRT Reporter Viewer

Source: Hans Lin, Brian Delaire

- Verify standard report “Disk Capacity by Disk”
- Verify standard report “Storage Subsystems: By Storage Subsystem”
- Test the dialog panel that lets you add and remove columns to configure the standard reports
- Test the “Filter” functionality for the standard reports
- Save a standard to the repository
- Import a standard report
- Export a standard report
- Test drill-down/drill-up functionality or reports (tentative)

#### 5.1.4.4 BIRT Reporter Designer

Source: Hans Lin, Brian Delaire

- Design two or more of the standard reports that are in the legacy GUI
- Save a new report to the repository

- Delete a report from the repository
- Delete a report from the workspace
- Open a saved report, modify the report in the workspace and resave to the repository
- Promote a report to “Installed Reports”
- Demote an “Installed Report”
- Verify that “Installed Reports” show up in the correct place in the navigation tree
- Import a newly created report
- Export a newly created report

#### 5.1.4.5 GUI Common & Base Tests

Source: Hans Lin, Brian Delaire

- 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) appear beneath the correct subnode.
- Test that you are allowed to save jobs (discovery, probe) 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

#### 5.1.4.6 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 and Forward, 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.