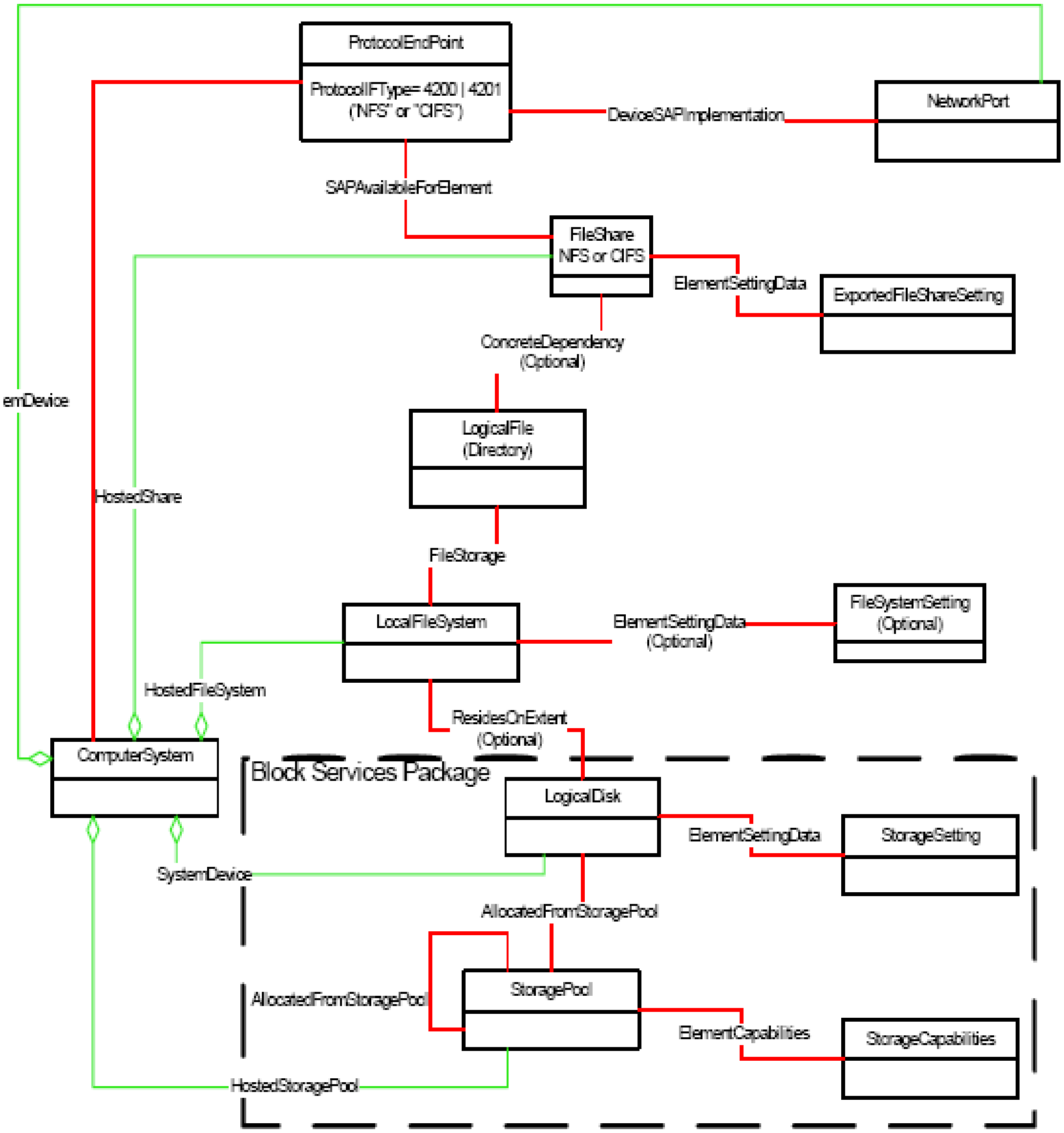
Support the Self-Contained NAS Profile (reporting only)

Main Components of the Architecture

The Self-contained NAS profile defines NAS systems that are self-contained in that all the storage they use to store the NAS data is part of the NAS System (and not exposed). As a result, the Self-contained NAS profile needs to be able to address aspects of physical storage. However, the physical storage aspects are already implemented as part of the Array Profile and will not be included into this document.



As with Arrays, the “top level” ComputerSystem of the Self-Contained NAS typically isn’t a real ComputerSystem. It is merely the ManagedElement upon which all aspects of the NAS offering are scoped.

Everything above the LogicalDisk is specific to NAS (and does not appear in the Array Profile). LocalFileSystems are created on the Logicaldisks, LogicalFiles within those LocalFileSystens are shared (FileShare) through ProtocolEndpoints associated with NetworkPorts.

The ResidesOnExtent is optional, but is shown here to illustrate that a LocalFileSystem may map to a LogicalDisk. However, other mappings to storage are also possible.
The FileSystemSetting (and the corresponding ElementSettingData) are also optional.

For Self-Contained NAS, LogicalDisks are the ElementType that is supported for storage allocation functions (e.g. CreateOrModifyElementFromStoragePool and ReturnToStoragePool) and LogicalDisk creation is optional. NAS also supports (optionally) the Pool manipulation functions (e.g. CreateOrModifyStoragePool and DeleteStoragePool) of the Block Services Package.

This document contains the design of the following aspects from the Self-Contained NAS Profile:

- Reporting on port connectivity to the Self-Contained NAS
- Reporting on the file systems and file shares that are configured out of the storage of the Self-Contained NAS

The new functionality will enable Aperi to gather information about file systems and file shares from a SMI-S Agent that implements the SMI-S 1.1.0 Self-Contained NAS System profile.

The new functionality is not supposed to be an isolated component but shall be integrated to make use of existing functionality to present the user a uniform experience. This shall be achieved by the reuse of existing database tables for conceptually analogous entities to let them appear in existing reports.

Some of the new reports, like the port connectivity report, will be designed using the BIRT technology and will be executed using the Aperi Report Server. Other ones (which are analogue to other existing reports) will be implemented in the classic way.

Network Appliance devices will be displayed in the GUI as both a ‘computer’ and a ‘subsystem’ if probed by both a proxy data agent and a Network Appliance SMI-S Agent. We need to correlate this SMI-S Agent information to the information from the data agent. The code will be analyzed to identify areas of the code that need to be adjusted to ensure that data is not duplicated and that data is properly displayed for all affected reports.

These are the topics covered:

- Database schema and the mapping of the object properties
- Probe design for data retrieval
- Correlation mechanisms to identify file systems and file shares to the ones reported by data agents
- Reporting
- Detectability, Removed resource retention

Data Model: Schema Mapping from SMI-S objects to Database

A NAS File Server will appear as a storage subsystem. Hence its assets will be reported just as assets on a storage subsystem. To facilitate this with minimum impact we will try to use existing tables, as much as possible, to store the data we get from the SMI-S Agent.
There will be several modifications to existing tables. The majority of the changes are made in order to add the additional attributes collected using the SMI-S Agent and to be able to put these tables under the control of the detectability service. UPDATE_TIMESTAMP is mandatory in any case. DETECTABLE is only for those entities that we don't want to be auto deleted by detectability service.

New columns are marked in green.

T_RES_ETH_PORT

Primary Key: SYSTEM_NAMES_ID, DEVICE_ID

Unique Constraint: ETH_PORT_ID

Column	Object Attribute or Description
ETH_PORT_ID	autogen
SYSTEM_CREATION_CLASS_NAME_ID	The ID of the scoping System's CreationClassName.
SYSTEM_NAMES_ID	The ID of the scoping System's Name.
CREATION_CLASS_NAME_ID	The ID of the name of the class or the subclass used in the creation of an instance.
DEVICE_ID	An address or other identifying information to uniquely name the LogicalDevice.
OPERATIONAL_STATUS	The current operational status of the port.
CONSOLIDATED_STATUS	calculate from operational status
PERMANENT_ADDRESS	The size of a block in bytes for certain file systems that use a fixed block size when creating file systems.
DETECTABLE	
UPDATE_TIMESTAMP	current probe timestamp

T_RES_FILESYSTEM

Primary Key: none

Unique Constraint: FILESYSTEM_ID

Index: GROUP_ID, LOGICAK_DISK_ID, COMPUTER_ID

Column	Object Attribute or Description
FILESYSTEM_ID	autogen
COMPUTER_ID	Subsystem id or equivalent computer id
GROUP_ID	-1
LOGICAL_DISK_ID	-1
LOG_DISK_ID	-1
MAXFILES	-1

USED_INODES		-1
FREE_INODES		-1
PHYSICAL_SIZE	capacity	
CAPACITY	capacity	
USED_SPACE	capacity - freespace	
FREE_SPACE	freeSpace	
FILE_COUNT		-1
DIRECTORY_COUNT		-1
LAST_SCAN_TIME	current probe timestamp	
FILESYSTEM_TYPE	type	
USE_COUNT		1
MOUNT_POINT	path	
DISCOVERED_TIME	timestamp of first probe	
SCANNING_COMP_ID		-1
EXPORT_NAME	blank	
OPERATIONAL_STATUS	The current operational status of the LocalFileSystem.	
CONSOLIDATED_STATUS	calculate from operational status	
BLOCK_SIZE	The size of a block in bytes for certain file systems that use a fixed block size when creating file systems.	
CASE_PRESERVED	Whether this file system preserves the case of characters in filenames when saving and restoring.	
CASE_SENSITIVE	Whether this filesystem is sensitive to the case of characters in filenames.	
MAX_FILE_NAME_LENGTH	The length of the longest filename.	
IS_FIXED_SIZE	Indicates that the filesystem cannot be expanded or shrunk.	
DETECTABLE		
UPDATE_TIMESTAMP	current probe timestamp	

T_RES_SHARE

This table relates various kinds of resources to a computer and in this case a storage subsystem.

Primary Key: none
Unique Constraint: COMPUTER_ID + RESOURCE_TYPE + RESOURCE

Column	Object Attribute or Description
COMPUTER_ID	id of storage subsystem or equivalent computer
RESOURCE_ID	id of physical vol, logical disk or filesystem
RESOURCE_TYPE	according type
SCAN_TIME	Current probe timestamp
REMOVED_TIME	epoch 0
PATH	path of physical vol or logical disk
NAME	
UPDATE_TIMESTAMP	current probe timestamp

This is table is filled for physical volumes, logical disks and file systems. Path and name are filled similar to what the data agent provides for computers.

T_RES_EXPORT

This table will contain file share information.

Primary Key_	none
Unique Constraint:	LOGICAL_DISK_ID, PARENT_LOGICAL_DISK_ID

Column	Object Attribute or Description
EXPORT_ID	autgen
COMPUTER_ID	Subsystem id
PROTOCOL	Protocol (CIFS or NFS)
PATH	path
EXPORT_NAME	name
DISCOVERED_TIME	timestamp of first probe
SHARING_DIRECTORY	Indicates if the shared element is a file or a directory
DETECTABLE	
UPDATE_TIMESTAMP	current probe timestamp

Probe design for data retrieval

The probe will collect data about network ports, file systems and file shares.

A Filesystem shall be represented in the model as a LocalFileSystem instance. A LocalFileSystem instance may have exactly one ResidesOnExtent association to exactly one LogicalDisk.

The FileSystem shall have a HostedFileSystem association to a NAS ComputerSystem. Normally this will be the top level ComputerSystem of the NAS profile. However, if the Multiple Computer System Subprofile is implemented, the HostedFileSystem may be associated to a component ComputerSystem.

The LocalFileSystem instance may also have an ElementSettingData association to the FileSystemSetting for the Filesystem. However, the FileSystemSetting is optional and may not be present.

The NAS Profile shall model any File Shares that have been exported to the network. A File Share shall be represented as a FileShare instance with associations to the ComputerSystem that hosts the share (via HostedShare), to the ExportedFileShareSetting (via ElementSettingData) and to the ProtocolEndpoint through which the Share can be accessed (via SAPAvailableForElement). Optionally, there may also be an association between the FileShare and the LogicalFile that the share represents (via ConcreteDependency).

The probing algorithm can thus be described as follows:

1. define the traversal and retrieve the data
2. process the data
    a. Port connectivity to the Self-Contained NAS
        1. iterate over the Ethernet Ports

        2. persist data
            a. T_RES_ETH_PORT
    b. File systems configuration
        1. iterate over Local File Systems for details
        2. evaluate the file system type and set the type property
        3. persist data
            a. T_RES_FILESYSTEM
            b. T_RES_SHARE
    c. File shares on local file systems that can then be accessed by remote clients
        1. Iterate over File Shares and Exported FileShare Settings
        2. persist data
            a. T_RES_EXPORT

A new class ProbeGenericNASSubsystem Process extending ProbeGenericSubsystemProcess will be defined.

For the probe the following methods will be defined:

- public IStep getStepCollectEthernetPortsFromComputerSystem(DiskCIMProcessor pProcessor, LogTraceHelper pLTH)
- public IStep getStepCollectFileSystemsFromComputerSystem(DiskCIMProcessor pProcessor, LogTraceHelper pLTH)
- public IStep getStepCollectFileSharesFromComputerSystem(DiskCIMProcessor pProcessor, LogTraceHelper pLTH)

The following mappers will be defined:

- SMISCIM_EthernetPortToDBMapper
- SMISCIM_LocalFileSystemToDBMapper
- SMISCIM_FileShareToDBMapper
- SMISCIM_ ExportedFileShareSettingToDBMapper
- SMISONTAP_LocalFSToDBMapper
- SMISONTAP_FileShareToDBMapper
- SMISONTAP_ExportedFileShareSettingToDBMapper

Correlation mechanisms to identify file systems and file shares to the ones reported by data agents

The correlation mechanism will be based on a new table storing the equivalent computers and storage subsystems (i.e. for a NAS probed by both the Data Agent and SMI-S Agent, the COMPUTER_ID found by the Data Agent for the NAS filer and the STORAGE_SUBSYSTEM_ID found by the SMI-S Agent).

This table will be populated by both Data Agent and SMI-S Agent at probe time based on the storage subsystem serial number or IP address.

The NAS File Server serial number is sent to the server by the proxy data agent at probe time and is stored into T_STAT_COMPUTER.SERIAL_NUMBER. The IP address is stored into T_RES_HOST.IP_ADDRESS.

The same serial number is also reported by the SMI-S Agent in the T_RES_STORAGE_SUBSYSTEM object repository as the SERIAL_NUMBER property. The IP address is stored into T_RES_STORAGE_SUBSYSTEM.IP_ADDRESS.

The computer / storage subsystem serial number and IP address will be analyzed during agent /storage subsystem probe and the equivalence table will be maintained. During reports generation, the COMPUTER_ID / SUBSYSTEM_ID from the equivalence table will be used.

At SMI-S Agent probe time, we will query the T_RES_FILESYSTEM if there is a row with the same name or with an equivalent computer and the same MOUNT_POINT.

If a row is found, the row will be updated, otherwise a new row will be inserted and the COMPUTER_ID set with the corresponding T_RES_STORAGE_SUBSYSTEM.SUBSYSTEM_ID.

Also at the data agent registration time we will check and update the T_RES_FILESYSTEM if a match is found based on the serial number or IP address and mount point.

Note: Allowing simultaneous usage of Data Agent and SMI-S Agent for the same NAS device will complicate very much the implementation: correlation issue, entities removal, reports generation. We can consider a partial solution which allow, for each device, only one type of probe. For example, if the device has already been probed as SMI-S storage subsystem and now someone is trying to probe it through a Data Agent, the probe will be rejected (device already probed through a different method). This simplification would make the implementation much easier and would avoid the complications mentioned above, but in the same time would prevent the system to make use of the features that are specific to only one agent.

Reporting

Both Data server and Report server reporting capabilities shall be leveraged to query and visualize the collected data.

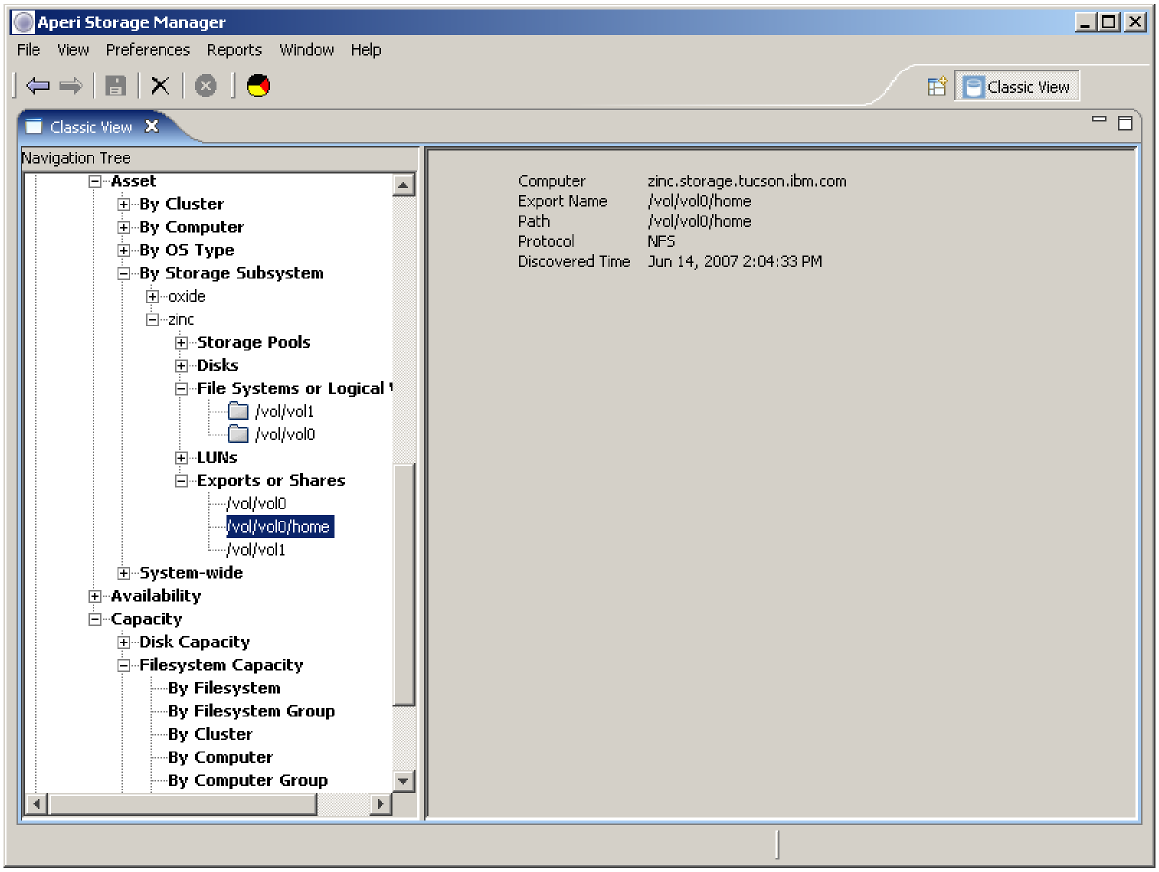
The existing file system asset reports will be enhanced to display the additional attributes as well.

New asset reports will be defined:

Data Manager->Reporting-Asset->By Storage Subsystems->File Systems or Logical Volumes

Data Manager->Reporting-Asset->By Storage Subsystems->Exports or Shares

These will be similar to the By Computer asset reports.



Computer	zinc.storage.tucson.ibm.com
Export Name	/vol/vol0/home
Path	/vol/vol0/home
Protocol	NFS
Discovered Time	Jun 14, 2007 2:04:33 PM

Affected reports

The following reports should be checked and fixed if not working properly:

Dashboard
Data Manager
System-wide
File Systems or Logical Volumes
By Freespace
By Probe Time
By Scan Time
By Discovered Time
Removed File Systems
Logical Volumes without File Systems
Exports or Shares
Capacity
Filesystem Capacity
By Filesystem
By Filesystem Group
By Cluster

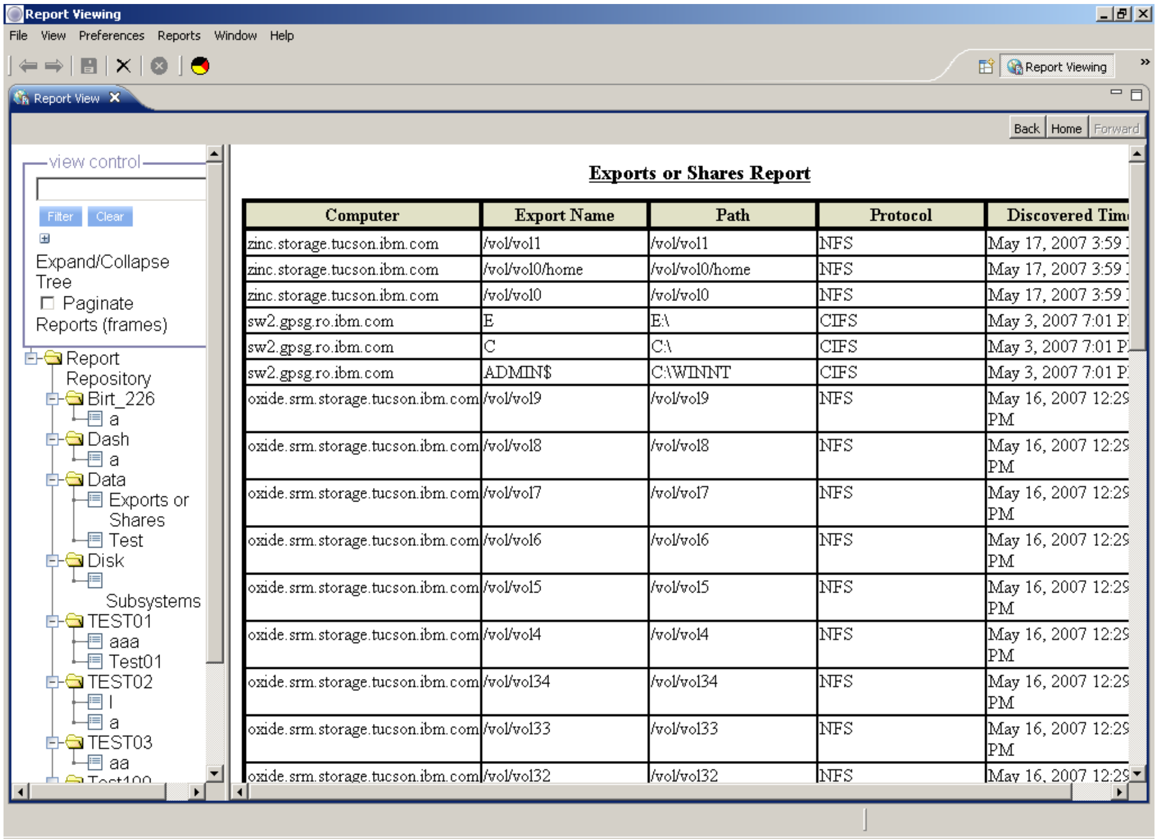
By Computer
By Computer Group
By Domain
Network-wide
Filesystem Used Space
By Filesystem
By Filesystem Group
By Cluster
By Computer
By Computer Group
By Domain
Network-wide
Filesystem Free Space
By Filesystem
By Filesystem Group
By Cluster
By Computer
By Computer Group
By Domain
Network-wide
By Computer Group
By Domain
Network-wide

The reporting on file shares was experimentally implemented using BIRT as well.

BIRT supports web oriented report design and has extensive customization and reuse capabilities.

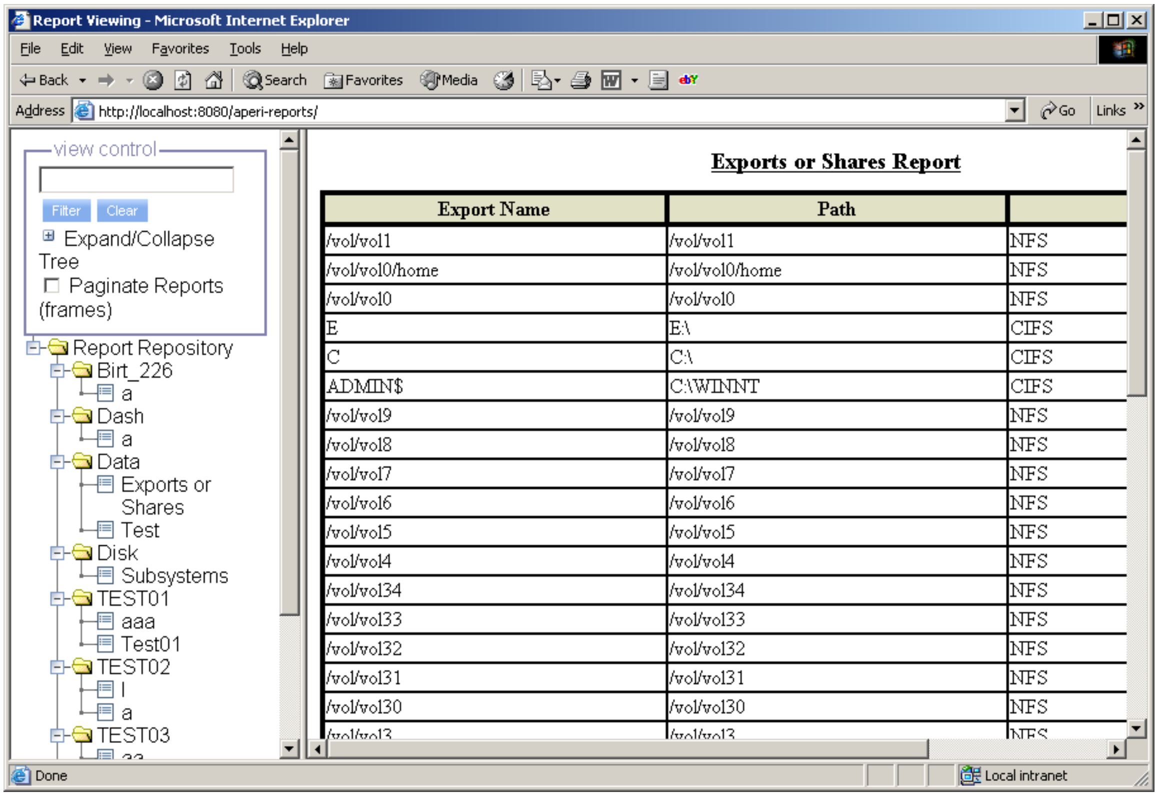
The following report was executed into the Aperi Report Server using the new aperi-reports web application.

The report can be displayed using the Aperi RCP GUI application or a browser.



Exports or Shares Report

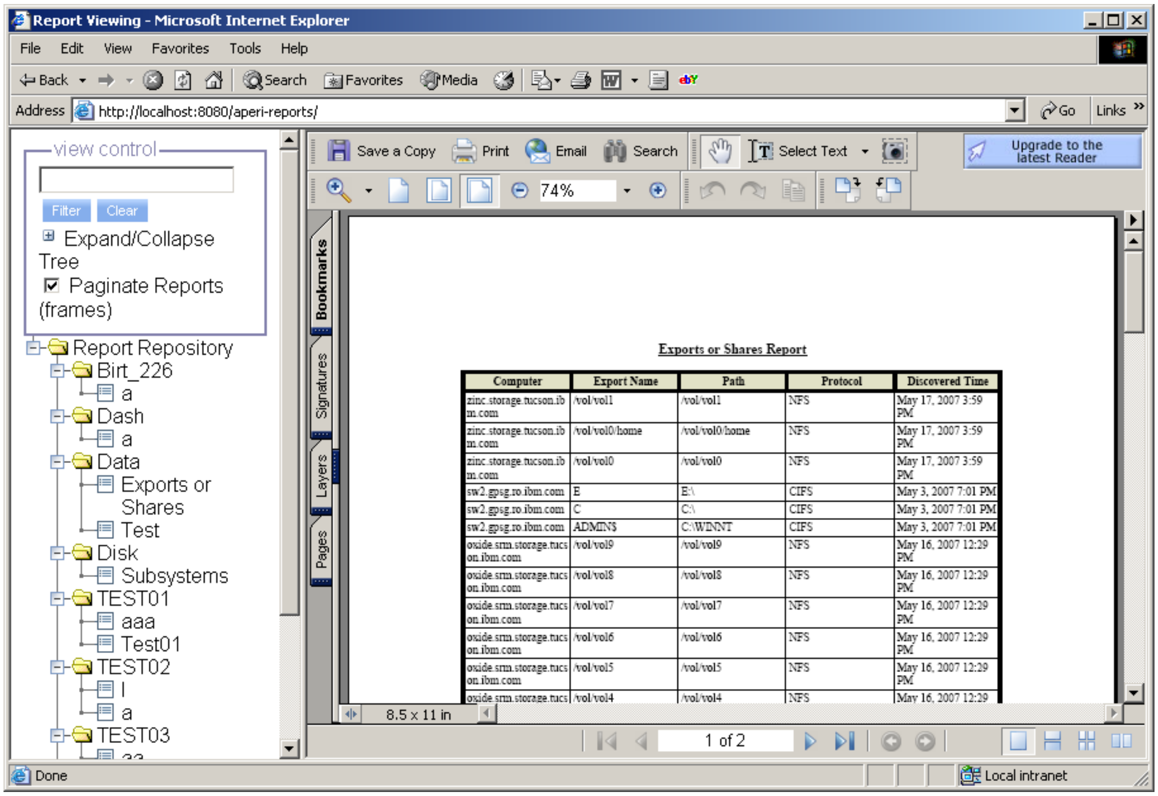
Computer	Export Name	Path	Protocol	Discovered Tim
zinc.storage.tucson.ibm.com	/vol/vol1	/vol/vol1	NFS	May 17, 2007 3:59
zinc.storage.tucson.ibm.com	/vol/vol0/home	/vol/vol0/home	NFS	May 17, 2007 3:59
zinc.storage.tucson.ibm.com	/vol/vol0	/vol/vol0	NFS	May 17, 2007 3:59
sw2.gpsg.ro.ibm.com	E	E:\	CIFS	May 3, 2007 7:01 P
sw2.gpsg.ro.ibm.com	C	C:\	CIFS	May 3, 2007 7:01 P
sw2.gpsg.ro.ibm.com	ADMIN$	C:\WINNT	CIFS	May 3, 2007 7:01 P
oxide.srm.storage.tucson.ibm.com	/vol/vol9	/vol/vol9	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol8	/vol/vol8	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol7	/vol/vol7	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol6	/vol/vol6	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol5	/vol/vol5	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol4	/vol/vol4	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol34	/vol/vol34	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol33	/vol/vol33	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol32	/vol/vol32	NFS	May 16, 2007 12:29


Report Viewing - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/aperi-reports/
Go Links
view control
Filter Clear
Expand/Collapse Tree
Paginate Reports (frames)
Report Repository
Birt_226
a
Dash
a
Data
Exports or Shares
Test
Disk
Subsystems
TEST01
aaa
Test01
TEST02
l
a
TEST03
aa
Done
Local intranet

Exports or Shares Report

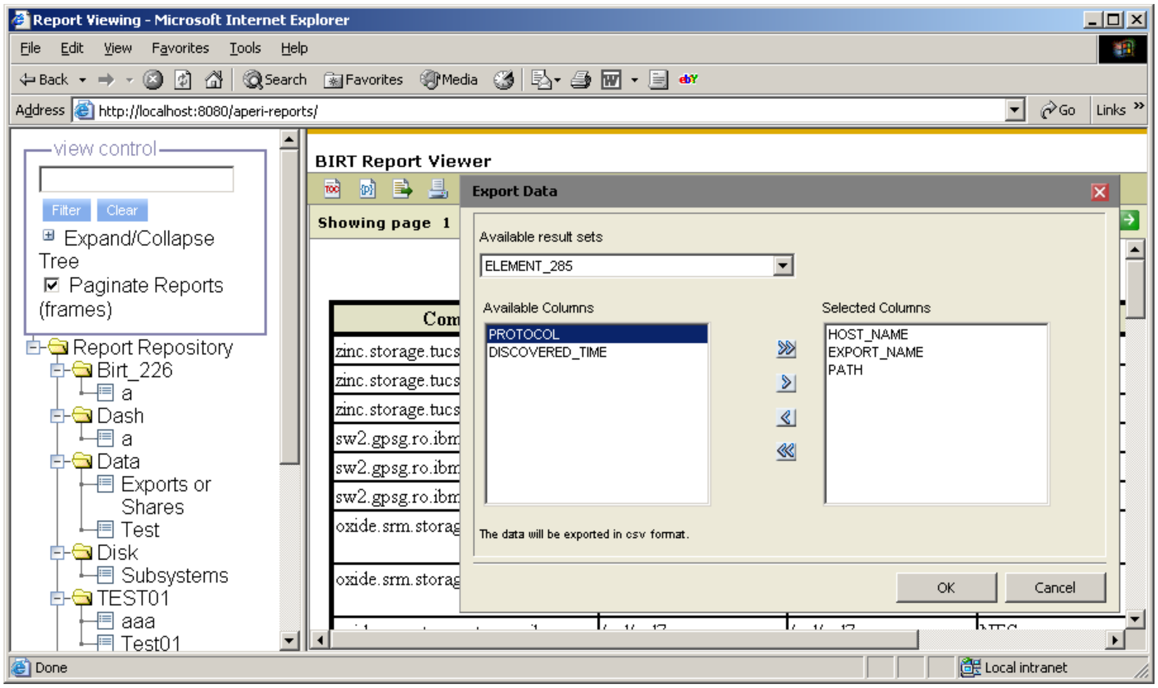
Export Name	Path	
/vol/vol1	/vol/vol1	NFS
/vol/vol0/home	/vol/vol0/home	NFS
/vol/vol0	/vol/vol0	NFS
E	E:\	CIFS
C	C:\	CIFS
ADMIN$	C:\WINNT	CIFS
/vol/vol9	/vol/vol9	NFS
/vol/vol8	/vol/vol8	NFS
/vol/vol7	/vol/vol7	NFS
/vol/vol6	/vol/vol6	NFS
/vol/vol5	/vol/vol5	NFS
/vol/vol4	/vol/vol4	NFS
/vol/vol34	/vol/vol34	NFS
/vol/vol33	/vol/vol33	NFS
/vol/vol32	/vol/vol32	NFS
/vol/vol31	/vol/vol31	NFS
/vol/vol30	/vol/vol30	NFS
/vol/vol3	/vol/vol3	NFS

The report can be printed in pdf format and has export capabilities.


Report Viewing - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/aperi-reports/
Go Links
view control
Filter Clear
Expand/Collapse Tree
Paginate Reports (frames)
Report Repository
Birt_226
a
Dash
a
Data
Exports or Shares
Test
Disk
Subsystems
TEST01
aaa
Test01
TEST02
l
a
TEST03
Save a Copy Print Email Search Select Text
Upgrade to the latest Reader
74%
Bookmarks
Signatures
Layers
Pages
8.5 x 11 in
1 of 2
Done
Local intranet

Exports or Shares Report

Computer	Export Name	Path	Protocol	Discovered Time
zinc.storage.tucson.ibm.com	/vol/vol1	/vol/vol1	NFS	May 17, 2007 3:59 PM
zinc.storage.tucson.ibm.com	/vol/vol0/home	/vol/vol0/home	NFS	May 17, 2007 3:59 PM
zinc.storage.tucson.ibm.com	/vol/vol0	/vol/vol0	NFS	May 17, 2007 3:59 PM
sw2.gpsg.ro.ibm.com	E	E:\	CIFS	May 3, 2007 7:01 PM
sw2.gpsg.ro.ibm.com	C	C:\	CIFS	May 3, 2007 7:01 PM
sw2.gpsg.ro.ibm.com	ADMIN$	C:\WINNT	CIFS	May 3, 2007 7:01 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol9	/vol/vol9	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol8	/vol/vol8	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol7	/vol/vol7	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol6	/vol/vol6	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucson.ibm.com	/vol/vol5	/vol/vol5	NFS	May 16, 2007 12:29 PM
oxide.srm.storage.tucs	/vol/vol4	/vol/vol4	NFS	May 16, 2007 12:29



Export Data

Available result sets: ELEMENT_285

Available Columns:
- PROTOCOL
- DISCOVERED_TIME

Selected Columns:
- HOST_NAME
- EXPORT_NAME
- PATH

The data will be exported in csv format.

Detectability, Resource Retention/Removal

Detectability and Remove Resource Retention device server components will be used to track the lifecycle of these entities.
Detectability columns will be added to the data agent related tables that are reused for storing file system and file share information. Detectability will not touch the content updated by the data agent since the update timestamp for data agent content is always null. This is how it already works today for the table T_RES_PHYSICAL_VOLUME. Detectability and Remove Resource Retention components will handle removal of entities from the tables populated by the probe.

Entity	Table	Auto-delete	Retention	Authoritative
Filesystem	T_RES_FILESYSTEM	No	Filesystems	yes
File Share	T_RES_EXPORT	No	Filesystems	yes

Unit Test

The reporting on port connectivity, file systems and file shares should work for any SMI-S 1.1.0 Agent that implements the Self-Contained NAS Profile.

Discover the SMI-S agent, create and run a probe job against it using the GUI.

The following set of reports should be validated:
TBD

If the NAS device is discovered and probed simultaneously using a data agent, some data could appear twice if no correlation logic is added.

Correlation logic will be added for the special case of Network Appliance devices.