

SANscreen

Capacity Manager 5.1

User Guide

NetApp, Inc.
495 East Java Drive
Sunnyvale, CA 94089 U.S.A.
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: <http://www.netapp.com>

Part number 215-04639_A0

April 2009

Copyright and trademark information

Copyright information

Copyright © 1994-2009 NetApp, Inc. All rights reserved. Printed in the U.S.A.

The third-party copyright statements and licenses that apply to open source software components distributed with the product are contained in the section “Third-Party Copyrights and Licenses.”

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S.A. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp, the Network Appliance logo, the bolt design, NetApp—the Network Appliance Company, Cryptainer, Cryptoshred, DataFabric, DataFort, Data ONTAP, Decru, FAServer, FilerView, FlexClone, FlexVol, Manage ONTAP, MultiStore, NearStore, NetCache, NOW NetApp on the Web, SANscreen, SecureShare, SnapDrive, SnapLock, SnapManager, SnapMirror, SnapMover, SnapRestore, SnapValidator, SnapVault, Spinnaker Networks, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, StoreVault, SyncMirror, Topio, VFM, VFM (Virtual File Manager), and WAFL are registered trademarks of NetApp, Inc. in the U.S.A. and/or other countries. gFiler, Network Appliance, SnapCopy, Snapshot, and The evolution of storage are trademarks of NetApp, Inc. in the U.S.A. and/or other countries and registered trademarks in some other countries. The NetApp arch logo; the StoreVault logo; ApplianceWatch; BareMetal; Camera-to-Viewer; ComplianceClock; ComplianceJournal; ContentDirector; ContentFabric; EdgeFiler; FlexShare; FPolicy; Go Further, Faster; HyperSAN; InfoFabric; Lifetime Key Management, LockVault; NOW; ONTAPI; OpenKey, RAID-DP; ReplicatorX; RoboCache; RoboFiler; SecureAdmin; Serving Data by

Design; SharedStorage; Simplicore; Simulate ONTAP; Smart SAN; SnapCache; SnapDirector; SnapFilter; SnapMigrator; SnapSuite; SohoFiler; SpinMirror; SpinRestore; SpinShot; SpinStor; vFiler; Virtual File Manager; VPolicy; and Web Filer are trademarks of NetApp, Inc. in the U.S.A. and other countries. NetApp Availability Assurance and NetApp ProTech Expert are service marks of NetApp, Inc. in the U.S.A.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. A complete and current list of other IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Apple is a registered trademark and QuickTime is a trademark of Apple, Inc. in the U.S.A. and/or other countries. Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the U.S.A. and/or other countries. RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the U.S.A. and/or other countries.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

NetApp, Inc. is a licensee of the CompactFlash and CF Logo trademarks. NetApp, Inc. NetCache is certified RealSystem compatible.

Table of Contents

Preface	6
NetApp Products	7
Documentation Set	8
Quick Start Instructions	9
Capacity Manager: An Example	11
The Plan	11
The Players/Roles	11
Assumptions	11
The Project	11
The Storage Request	12
The Reservation	12
Completed Request	14
Audit	14
Consequences and Advantages	15
Chapter 1 Overview and Licensing	16
1.1 Product Overview	16
1.2 Sequence of Steps for Using Capacity Manager	16
1.2 State Transitions of a Request	18
1.3 License Installation	17
Chapter 2 Placing a Request	27
2.1 Create a New Request	27
2.1.1 Target Tab	29
2.1.1.1 Add Existing	30
2.1.1.2 Add by App	31
2.1.1.3 Add New	31
2.2 Set Up Requirements	32
2.2.1 Requirements Tab	46
2.2.1.1 Add Ports	32
2.2.1.2 Add LUNs	33
Chapter 3 Using Rules: Service Quality	48

Chapter 4 Reserving Optimal Resources	36
4.1 Make the Reservation	36
4.2 Satisfying a Requirement	37
4.2.1 Volume Requirements	38
4.2.2 Port Requirements	39
Chapter 5 Cancelling a Request	41
5.1 Perform a Deallocation of a Reservation	41
5.2 Remove a Request	42
Chapter 6 Communicating the Status of Requests	44
6.1 View Pending Tasks	44
Chapter 7 Reports	45
7.1 Cost Accounting	45
7.2 Storage Utilization	47
7.2 Chargeback	50
7.2 Allocation Audit Report	51
Chapter 8 External System Integration	52
Integration - Capacity Manager	53
Integration - Annotations, Applications, Business Units	63

Table of Figures

Figure 1: Request View	27
Figure 2: Detail Icons.....	27
Figure 3: Requests with Detail Icons Maximized.....	28
Figure 4: Creating a Request	28
Figure 5: Request Fields Completed	29
Figure 6: Adding Existing Hosts	30
Figure 7: Adding Hosts by Application.....	30
Figure 8: Adding New Hosts.....	31
Figure 9: Adding Ports to Requirements	32
Figure 10: Ports Shown on Requirements Tab	33
Figure 11: 3 LUNs of 60 GB Added to Requirements	33
Figure 12: 2 LUNs of 10GB Added to Requirements	34
Figure 13: LUNs Added to Requirements Tab	34
Figure 14: Reservations Tree.....	36
Figure 15: Using General Reservations	37
Figure 16: Adding a Reservation	37
Figure 17: Volume Reservation	38
Figure 18: Reservation Switch Filter.....	39
Figure 19: Port Reservation.....	39
Figure 20: Satisfying Requirements in Reservation	40
Figure 21: Reservation Violations.....	40
Figure 22: An Allocated Request.....	40
Figure 23: Confirmation of Cancelled Request.....	40
Figure 24: Remove Request Option	41
Figure 25: Removing a Request.....	41
Figure 26: Pending Tasks View	43

Preface

The *SANscreen Capacity Manager User Guide* describes SANscreen Capacity Manager's functionality and usage, giving you a detailed and comprehensive understanding of how to automate your storage request and delivery process. The scope of this user guide assumes familiarity with the SANscreen Service Insight product.

The SANscreen Capacity Manager provides real-time visibility into global resource allocations, rule-based automated service tier management, and utilization and chargeback reports.

Capacity Manager uses the service information discovered by the innovative SANscreen Service Insight software to manage the end-to-end resource request process. This allows companies to accelerate application provisioning, manage reservations of different resources, forecast based on future demand and simplify the provisioning process--enabling front-line support engineers to create and use accurate service provisioning action plans.

The two aspects complement each other: visibility, utilization and capacity related decisions are enhanced by a better request process and reservation-based visibility into future demand. The request delivery process is enhanced by accurate real-time resource allocation visibility and assisted decision process for resource selection.

Capacity Management Challenges

Successful capacity management involves management of the life cycle of capacity allocation, ensuring the most accurate data on past current and future needs and facilitating educated decisions based on this information.

Given the complexity of today's data center environments, capacity management becomes an increasingly difficult task. Even the simple task of understanding resource current allocation can be very difficult. In addition, most organizations have a nonstructured provisioning process and lack robust reservation capabilities. This can hinder the process of making the right provisioning decisions based on current and future capacity usage and can prevent gaining insight into future demand.

As a result, VPs of infrastructure and storage teams are facing continuous inefficiencies and exposure to risk around capacity issues, purchasing decisions and the provisioning process due to:

- The inability to understand the mapping between applications and SAN resources or to interpret historic usage trends.
- The challenge of effectively managing tiered services beyond the basic "by vendor/model" to further optimize the usage of existing resources.
- Making provisioning decisions based on information maintained in spreadsheets. In most cases, this information is not up-to-date--leading to mistakes, inefficient provisioning plans, and a lower quality of provisioning service and delays.
- Lacking a standard reservation system for SAN resources, so that unavailable or inaccurate information can cause organizations to over-invest in standby capacity or risk delays, brownouts and outages.
- Using a complex unstructured process combined with the need to use front line IT engineers to reduce costs--different skill levels can lead to incorrect provisioning, latent quality issues or outages.

As a result, in order to deal with the uncertainties, organizations may resort to expensive over-provisioning and inaccurate guesswork for capacity management decisions. In addition, these organizations are exposed to a risky and long turnaround time for new business requirements.

The critical task of assigning storage to applications in your SAN is often a slow, difficult and expensive process, involving a series of steps that must be followed in an assigned order to be effective (the provisioning process lifecycle). These steps progress from: 1) a request being sent with the specific service requirements; 2) optimal resources being reserved at the right location to satisfy the request; all the way to 3) planning non-disruptive configuration changes of hosts, volumes, zone sets, LUNs, and switches; and 4) execution of these changes by distributed operational teams. The process ends with 4) validation of the correct execution and 5) an email or report to the user on the availability of the storage.

The provisioning process allows a greater end-to-end management of all these resources. As this user guide will describe, the SANscreen Capacity Manager uses provisioning to:

- Enable an integrated process for off-shoring or between storage operation groups
- Enable integrated provisioning of different tiers of service
- Provide resource reservation data for planning of future purchases
- Reduce risks of making configuration errors.
- Reduce time to provision a service.
- Reduce operational costs.

Using this product the storage manager can perform common and customized storage provisioning requests to:

- Allocate capacity to new hosts or applications.
- Add capacity to existing hosts or applications.

NetApp SANscreen Products

The following products comprise the NetApp SANscreen suite of solutions.

Product Name	Description
Service Insight	Offers visibility into your SAN environment, including host-to-storage access paths, storage availability and change management.
Service Assurance	Enables real-time monitoring of storage services and risk-free migrations.
Capacity Manager	Capacity Manager manages the end-to-end resource order and allocation process to allow companies to accelerate application provisioning, manage resource reservations and forecast based on past trends and future demand, enabling educated capacity management decisions.
VM Insight	Enables virtual machine to array path visibility for organizations deploying VMware ESX technology combined with VMware Virtual Center.
Application Insight	Uses SAN traffic data to reclaim underutilized resources, manage tiers, identify multipath risks and troubleshoot ongoing performance bottlenecks. Discovers, collects and maintains performance information measured by critical array metrics throughout the SAN.

Documentation Set

The following table lists the full documentation set for NetApp's suite of products. These manuals are available from the client Help menu.

User Guide	Description of Content
SANscreen Installation and Administration User Guide	Overview of the suite's architecture with instructions to get the system up and running, discover the SAN and perform administrative tasks.
SANscreen Service Insight User Guide	Information about the tools and features available to help you gain visibility into your SAN environment.
SANscreen Service Assurance User Guide	Overview of how to use Service Assurance to analyze and validate your storage network environment and to automate operational monitoring, as well as procedures to make safe changes without disrupting availability.
SANscreen VM Insight	Overview of Virtualization in SANscreen.
SANscreen Application Insight User Guide	Overview of how to use Application Insight to reclaim underutilized resources, manage tiers, identify multipath risks and troubleshoot on-going performance bottlenecks. Provides information about how the Array Performance Module for Application Insight discovers, collects and maintains performance information measured by critical array metrics throughout the SAN.
SANscreen Capacity Manager User Guide	Overview of how to use Capacity Manager to enable educated capacity management decisions by managing the end-to-end resource order and allocation process.
SANscreen Connect API User Guide	Overview of how to use the API interface to enable integration with other applications, such as reporting and monitoring systems.

Quick Start Instructions

Follow these steps to get SANscreen Capacity Manager up and running as quickly as possible:

Task	For details see...
1. Install the SANscreen software.	<i>SANscreen Installation and Administration User Guide</i>
2. Install the license. Make sure you have at least the following products enabled: Service Insight or Service Assurance.	<i>SANscreen Installation and Administration User Guide</i>
4. Deploy SANscreen data sources, discover the SAN and identify the Hosts.	<i>SANscreen Installation and Administration User Guide</i>
5. Access the SANscreen Portal and configure users, mail and SNMP if needed.	<i>SANscreen Installation and Administration User Guide</i>
6. Launch the SANscreen client.	<i>Service Insight User Guide</i>

Capacity Manager: An Example

The following example illustrates how the application of a structured workflow and validation can facilitate a proactive approach to process automation and provide optimum utilization within the SAN. The core of this process lies in the analysis of all resources in the relevant context and cost — hosts, switches, ports and storage.

Starling Enterprises, Inc., a mid-sized organization with 1,200 employees and based in the United Kingdom, has a critical ERP system managing and synchronizing many different departments. The ERP became the most critical application affecting almost all employees. This example will describe a project to extend the ERP and explain how the storage portion of it was managed.

The Plan

After a year of successful integration, the decision was made to extend the implementation of ERP to the entire financial division.

The Players/Roles

ERP Application Manager – In charge of the ERP application

System Administrator – Manages the server resources

Storage Administrator – Manages day-to-day operations

Storage Engineer – Manages storage architecture and projects

SANscreen Capacity Manager views – Demonstrates the impact of using the product

Assumptions

- ♦ The storage engineer has designed a system for storage management and tiers in the SAN.
- ♦ Best practice guidelines for selecting resources for reservation have been established.

The Project

The ERP applications manager in Inverness begins planning a major project for an ERP extension upgrade to accommodate the need. To that end, he calls on his system administrator to build the server portion of the planning cycle. The new resources must be allocated by October, the deadline for the ERP extension to be made available.

The Storage Request

1. The system administrator in turn sends several requests over to the storage administrator, each requiring resources allocated by the assigned deadline. His priority is to get the resources delivered on time, to ensure no delay for the server build and to get the best performance he needs at the cheapest price.

Each such request includes the following:

- Identification data:
 - Physical location for the resources, such as the data center , floor, room , etc.
 - Business unit or entity that will pay for the resources
 - Tabs to set up requirements:
 - Number and size of LUNS required
 - Number of ports required
2. In this example, one of these requests includes the following requirements: Two new hosts connected to the SAN, each requiring 4 4Gb ports and 10 Tier-1 60GB LUNs.

The storage administrator decides to use SANscreen Capacity Manager to manage the provisioning process. For initial setup, he:

- Installs the SANscreen suite.
- Installs the license keys for three NetApp products: Service Insight and Capacity Manager.
- Uses the *SANscreen Installation and Administration User Guide* and the *SANscreen Service Insight User Guide* to configure SANscreen for general operation.

The storage administrator who needs to perform the actual task of adding the necessary storage waits for verification of the request before he moves to the next step. The request has an ID and project number when verified and then the storage administrator is ready to move to the next step of the process and reserve the resources.

The Reservation

1. The storage administrator reviews all open storage requests by project and review requirements and cost. He sees that the system administrator had originally asked for 10 LUNs at Tier 1, but that was seen as excessive cost relative to the application needs. The storage administrator responds to the system administrator with a recommendation to change the requirement to 4 Tier-1 60GB LUNs and 6 60GM Tier-2 LUNs, which cost a quarter of the price.

It is the job of the storage engineer to recommend the tier to use from a storage perspective. It is not uncommon for resources to be overprovisioned — so this is something that needs to be monitored. This is a crucial stage where storage engineers need to step in and make sure the right decisions are made. The system administrator approves the recommendation.

2. The storage administrator starts the process of reserving resources to satisfy the requirements of the request. Based on the original request, the storage administrator needs essential information to do this, such as:
 - What volumes are available in what physical location that meet the LUN requirements?

- Which arrays are least utilized to use for the reservation?
- What connections to the fabric exist?
- What switch ports are available?

He continues to make requests until all requirements of the request are satisfied.

With answers to these questions, he then understands the capacity and requirements he needs and how to obtain them. With this information, he is able to reserve resources, logically carving up disk space, ensuring storage available in the space that is connected to the fabric, and specifying a switch that has enough ports in the fabric. The resulting reservations will display in the Pending Tasks view.

This stage of the plan has been executed successfully. But are the reservations valid?

To ensure successful provisioning, there must be validation of the request to guarantee that the resources are sufficient and correctly reserved. Have there been any violations as a result of the reservation?

3. In this case, the storage administrator made a mistake and reserved the storage space on a hot array, unknowingly creating future performance issues for the ERP application. The other aspect was that the tier assignment was incorrect: Volumes were assigned to the wrong tier. Lastly, not enough volumes were reserved to match the LUN request.
4. The storage administrator investigates the different reservation-related violations, identifies the problem and easily corrects it using the views in SANscreen Capacity Manager. For example, a request for a LUN of Tier Gold is erroneously reserved on a Tier Bronze volume. The storage administrator edits the request from the Reservation Violations screen and selects an appropriate volume in the Tier Gold, or changes the requirement to Tier Bronze. The revision appears immediately in the system and on the GUI. The violation is resolved, and the request is reserved.

Note

Note: The SAN administrator has the responsibility and authority to most effectively meet the requirements—thus he can adjust those requirements or even reject them if he sees that action as benefiting the provisioning process.

5. As a result of the resources reserved for this project, a capacity planner receives an automatic report saying a threshold has been identified in Tier-1 storage on the 4th floor in the Inverness data center. It is the capacity planner's job to use business plans and forecasts to help determine future needs. He requests that more capacity be purchased for that data center to be on time for the storage allocation.

The Request is now reserved, but the process is not yet complete. The resources have not been allocated. What has happened? Unexpected problems/issues have shown up:

6. Over the weekend someone carried out an emergency allocation without checking on the status of the storage space, and in the process, that person allocated an item that was already reserved.

The storage administrator sees the resulting violation in the SANscreen Capacity Manager and needs to find new resources. While this is not a difficult procedure, it can be tedious. He creates new reservations, which prevents the situation from becoming more serious—but what if those resources had not been available? The willy-nilly assignment of storage, even for emergencies, can damage

and /or degrade applications running on the SAN: it can create outages when volumes are shared by applications, or delay storage assignments if storage device is running out of space.

The storage administrator has to ensure that the reservation was not taken out of the provisioning life-cycle, to prevent such process flaws from emerging.

7. The storage administrator who has to implement the request builds a complete configuration plan, a process that is absolutely required to provision the request. In the course of executing this plan, the storage administrator only provisions 50% of what was reserved. This problem triggers a violation stating that only 2 or 3 of the total number of reservations have been allocated and completed. The storage administrator reviews the violation and completes the storage allocation for the entire request.
8. Suddenly, and after the complete project resources were allocated, the ERP application manager calls the system administrator and reports that half the project has been cancelled. This information is handed down to the storage administrator, who must process the cancellations.

Requests that were reserved but never allocated will simply disappear.

Requests that were allocated now have to be de-allocated. The storage administrator in charge of provisioning carries this out. The administrator validates the plan and removes the configuration associated with the reservation.

The SAN administrator can use the utilization and threshold reports and the automated tiering to make ongoing decisions about capacity. This information can reveal what tiers are running low and what thresholds have been crossed, what information the data centers are able to present to the Business Units, what resources are being used and what chargeback has been accounted for and reported. Such information helps the SAN administrator to make the best decisions when ordering more capacity.

Completed Request

Finally, the physical equipment has been set up and connected, all configurations have been set and storage is available to the new host. At this stage, the storage administrator verifies the request and marks it as Complete in the State column (by right-clicking on the field).

Note

NOTE: The company auditor receives a report that proves all storage allocations are accounted for and approved and no compliance issues were detected.

Audit

As part of the report, the auditor sees an audit compliance problem in the emergency allocation that occurred over the weekend. He tells the person who performed that allocation not to use the system in that haphazard way. The auditor emphasizes that resources must be allocated according to the architecture and design planning that has been set up for the SAN.

Consequences and Advantages

Value has been added to every step, as shown in the table at the end of this user guide, with cost savings added in. Alternatives, options and improvements in the process have resulted in a smooth path to completing storage requests successfully.

Chapter 1 Overview

This chapter covers the prerequisites necessary to use Capacity Manager successfully. The rest of this chapter is set up as follows:

- **Section 1.1** *Product Overview*
- **Section 1.2** *State Transitions of a Request*
- **Section 1.3** *License Installation*

1.1 Product Overview

The SANscreen Capacity Manager is designed to assist SAN managers in automating their storage request and delivery process. This helps determine how resources are assigned and implemented in the SAN to enable end-to-end availability.

Using the NetApp SANscreen API, a customized, flexible plan can be enabled and integrated using the steps for just three principal features: requests and requirements, reservations and violation reports.

1.2 Sequence of Steps for Using Capacity Manager

You can handle the ongoing collection of information by looking at the capacity utilization reports to make capacity purchase decisions for storage capacity in different tiers and ports. Automation of the provisioning process allows a greater end-to-end management of all resources.

The first two steps described here are usually carried out by the system administrator. The remaining steps are generally completed by the storage administrator.

See also the capacity example in the previous section of this user guide.

1. Set up the Rules for your tiering strategy.

Note

*Before setting up Rules, you must set up Annotations. See the **Service Insight User Guide**.*

2. Create a new Request, providing its profile data. A ticket number and ID are assigned to that Request.
3. Set up the Requirements in the Request for the resources that need to be allocated. These include LUNs and ports.

4. Observe the cost associated with each Requirement. The storage administrator who manages the Request will assess whether the cost is appropriate to the application and may suggest a change in the Requirement via email.
5. If the Request has been approved (and the SAN administrator has not rejected it) the storage administrator begins the process of reserving the resources to satisfy the Requirements of that Request. This process is not finished until all Requirements have been met.
6. The storage administrator reserves the necessary storage space and number of switch ports.
7. Since reserved resources cannot be accessible or made available for other requests, the storage administrator has to validate the Request and verify that there are no violations, such as assignment to the wrong tier, to a device that is already operating at peak volume, or to resources that have already been allocated.
8. Using Capacity Manager views, the storage administrator can discover a problem and correct it.
9. A list of Pending Tasks is maintained by the Capacity Manager to ensure proper and timely implementation of provisioning of the reserved resources.
10. Finally, the storage administrator has to verify that the resources have been allocated as planned. If they have, then the Request is changed from Open to Completed.
11. It is possible for a Cancellation to occur because the original terms of the Request have changed. If allocation of resources has occurred, as well, then the Requests must be de-allocated using the Capacity Manager. This de-allocation is validated and the resources become available again.

The physical equipment is set up and connected. The Request can now be manually assigned as Completed.

You can assign different costs to options that are defined using switch levels and tiers from Annotation Types. Capacity Manager allows the SAN administrator to partition this information into three reports:

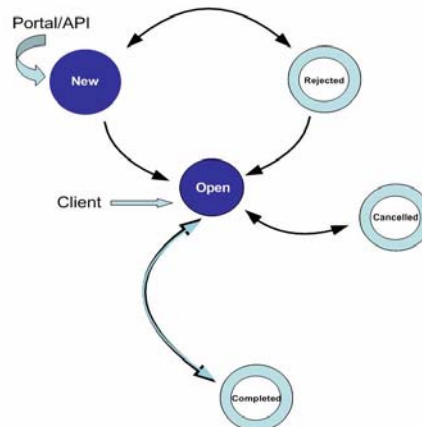
- "Utilization report, per tier and data center, for storage and switch that enables purchase planning.
- "Chargeback report showing historical storage consumption per BU with charge per GB.
- Additional value can be created by using the API. See *Chapter 8*.

The benefit of implementing the request process using the Capacity Manager is that mistakes related to overbooking resources are prevented and the repository of requests creates visibility into the demand pipeline of the organization. This lets the storage administrator know what resources are needed and forecast when to buy them.

When the SAN does not have visibility into the demand pipeline, it is typically necessary to maintain a large safety buffer of resources. Being able to see into the demand pipeline gives the SAN administrator the ability to reduce this buffer. This dramatically reduces capital expenditures related to the buffer, as well as to the cost of overhead.

1.3 State Transitions of a Request

The following state diagram illustrates the valid transitions for Requests. The illustration shows the manual process of steps a user must perform to push Requests through the system and so obtain the necessary resources to manage storage in the SAN:



The transitions that are permitted in the process define the choices you must make when configuring the Requests for Capacity Manager. These paths represent the possible behavior of Requests as they are processed through the SAN.

A New Request goes automatically into the Open state. By right-clicking on the Open state the single option of Cancelled appears. To reach the Completed state, the normal end, other actions must occur first.

A Request is in the Open state when it has not been acted upon—it has not been Rejected, Cancelled or Completed, all of which are viable exit options, depending on what stage of provisioning you are configuring that cause significant changes in the outcome.

Chapter 2 Using Rules: Service Quality

Creating a tiering strategy for your SAN can directly increase its utilization and optimize service levels. The tiering Rules in Capacity Manager let you classify certain data to specific tiers, and the automatic features ensure that the tiered infrastructure can be easily established and managed.

Using the Volume and Storage Rules, you are able not only to set up tiering for different types of resources and vendor-specific devices, but also to facilitate the transfer of data between them.

The rest of this chapter is set up as follows:

- **Section 2.1** *Configure Annotation Rules*
- **Section 2.2** *Storage Rule*
- **Section 2.3** *Volume Rule*

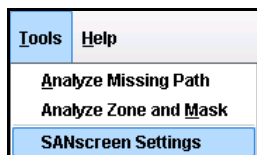
2.1 Configure Annotation Rules

You can set the Annotation Rules in any systematic way you want to manage the SAN. This saves time because you do not need to manually categorize your resources into tiers. The criteria you use could be based on storage models, fast disks, more expensive equipment, etc. For example, using Annotation Rules helps you set thousands of volumes under the same description.

Note

NOTE: *Tier Annotation options described in the Service Insight User Guide must be configured prior to creating Rules.*

1. Click **Tools** in the SANscreen client menu and then select **SANscreen Settings**:



2.2 Storage Rule

Name	Description	Result
Foo		Gold
EMC is silver		Silver

- Click on the **Add** button to open the next view. *This is where tier values are set.*

Model

- Enter a **Name** and **Description** for the Rule.
- Click on **Result Value** to assign a tier to the rule from the drop-down box. The tiers have been created in Annotation Types.
- Click on the drop-down box for **Vendor**. Select a vendor to be assigned that tier level for this Rule. Note that the vendor information can be filtered by model. If you select more than one model, the rule will be matched so long as the storage is any of the selected models.
- Click on **OK**. The vendor name will show up in the first view with the assigned tier value. You can then edit the rule again and using different names and descriptions, assign multiple vendors to the same tier value, or divide them according to your SAN classification requirements.

Note

NOTE: Rules evaluate in order. The first rule that matches for a storage determines the result. Rules can be re-ordered using the **Up** and **Down** buttons.

- Once all assignments have been made, click **OK** on the main Storage Tier Quality view. The storage tiers will appear immediately in the SANscreen Client.

Note

NOTE: Rules are constantly being evaluated; they are not cached or updated. Whenever a tier is being shown, SANscreen runs the rules and checks whether any rules apply if a tier has not been manually set.

2.3 Volume Rule

The Volume Tier Quality Rule allows you to perform the same classification as shown in the steps for Storage Tier Quality. However, for the Volume Rule, you can also filter the rule by tabs that include Volume Type, Redundancy, Disk Type, Disk Speed and Disk Size.

Once a rule is set, values assigned by the rule show up in the SANscreen Client in italics. If you hover the mouse over an italic value, the rule that provided that value is displayed in a tool tip.

- Highlight the **Volume Tier Quality** option under Annotation Rules and click the **Edit** button to get the following view.

Manage Annotation Rules

Manage Annotation Rule Set

Define the rules for automatically annotating a Volume with a Tier

ID: Volume Tier Quality

Name:

Description: Classification of volume resources by quality

Name	Description	Result
IBM is Gold		Gold
EMC Bronze		Bronze
EMC is silver		Silver

Buttons: Add..., Edit..., Remove, Up, Down

Buttons: OK, Cancel

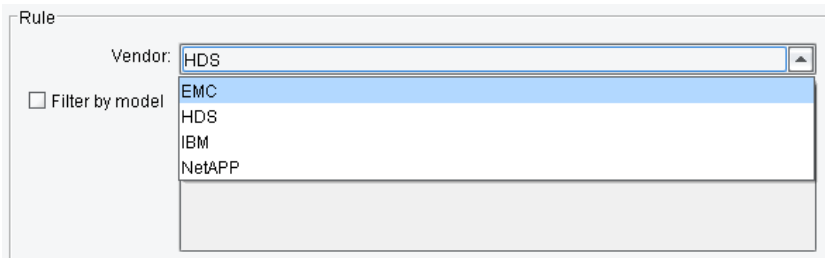
- Click on the **Add** button to open the next view, where you assign volume tier values.

The screenshot shows a window titled "New Rule" with a subtitle "Manage Annotation Rule". Below the subtitle is a description: "Define the rule for automatically annotating a Volume with a Tier". The form contains several fields: "Name:" (text input), "Description:" (text input), and "Result Value:" (a dropdown menu). Below these is a "Rule" section with a "Vendor:" dropdown menu and a checkbox labeled "Filter by model". To the right of the checkbox is a table with a header "Model" and an empty body. Below the "Rule" section is a "Filter Volumes by:" section with four tabs: "Volume Type", "Redundancy", "Disk Type", "Disk Speed", and "Disk Size". The "Volume Type" tab is selected. Below the tabs is a checkbox labeled "Filter Volumes by Volume Type" and a table with a header "Volume Type" and an empty body. At the bottom of the window are "OK" and "Cancel" buttons.

3. Enter a **Name** and **Description** for the Rule.
4. Click on **Result Value** to assign a tier to the rule from the drop-down box. The tiers have been created in Annotation Types.
5. Click on the drop-down box for **Vendor**. Select a vendor to be assigned that tier level for this Rule. Note that the vendor information can be filtered by model number.
6. View the tabs listed below: **Volume Type**, **Redundancy**, **Disk Type**, **Disk Speed** and **Disk Size**. Note that each one is automatically filled with associated data according to the vendor you have selected. If you want the volumes to be classified more specifically, you can use these tabs as filters to determine how the volume assignments to the tiers should occur.
7. Click **OK**. The vendor name will show up in the first view with the assigned tier value. You can then edit the Rule again and using different names and descriptions, assign multiple vendors to the same tier value, or divide them according to your SAN classification requirements.
8. Once all assignments have been made, click **OK** on the main Volume Tier Quality view. The volume tiers will appear immediately in the SANscreen Client.

2.3.1 Filtering Volumes

Select a vendor from the vendor list.



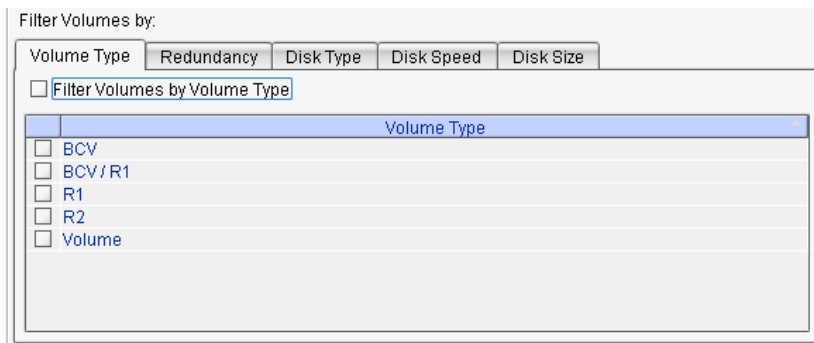
You can also check **Filter by Model** to further narrow the scope of the Rule to specific models of devices from that vendor.

The five tabs shown in Figure 24 let you filter volumes for the specified vendor and model.

For each tab:

- If you enable the filter, you must select at least one volume.
- If there are no volumes selected, you will see an error message at the top of the dialog and a red checkmark on the tab header.
- If there are no options to select, it means that the vendor and model do not have volumes that apply to that tab.
- If you need to temporarily disable a filter, your filter selections will stay intact.

Volume Type:



Redundancy

Filter Volumes by:

☒ Volume Type ☒ Redundancy ☐ Disk Type ☐ Disk Speed ☐ Disk Size

☐ Filter Volumes by Redundancy

	Redundancy
☐	[Data, Data, N/A, N/A]
☐	[Data, N/A, N/A, N/A]
☐	[Remote R1, RAID-5, RAID-5, N/A]
☐	[Remote R2, RAID-5, RAID-5, N/A]

Disk Type

Filter Volumes by:

☐ Volume Type ☒ Redundancy ☒ Disk Type ☐ Disk Speed ☐ Disk Size

☐ Filter Volumes by Disk Type

	Disk Type
☐	SATA

Disk Speed

Filter Volumes by:

☐ Volume Type ☐ Redundancy ☐ Disk Type ☒ Disk Speed ☐ Disk Size

☐ Filter Volumes by Disk Speed

	Disk Speed
☐	4800

Disk Size

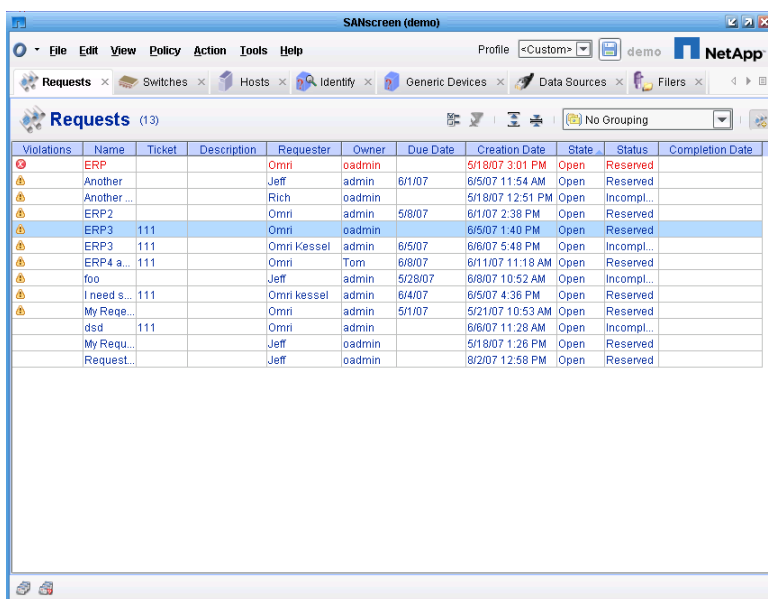
Volume Type	Redundancy	Disk Type	Disk Speed	Disk Size
☐ Filter Volumes by Disk Size				
				Disk Size
☐ 1000,2000				

Chapter 3 Placing a Request

This chapter explains the specific procedures required to make a Request, including the setup of Requirements.

3.1 Create a New Request

1. Click on Requests in the Capacity Manager section of the launch menu to get the following view.



Violations	Name	Ticket	Description	Requester	Owner	Due Date	Creation Date	State	Status	Completion Date
ERP				Omri	oadmin		5/18/07 3:01 PM	Open	Reserved	
Another				Jeff	admin	6/1/07	6/5/07 11:54 AM	Open	Reserved	
Another...				Rich	oadmin		5/18/07 12:51 PM	Open	Incompl...	
ERP2				Omri	admin	5/8/07	6/1/07 2:38 PM	Open	Reserved	
ERP3	111			Omri	oadmin		6/5/07 1:40 PM	Open	Reserved	
ERP3	111			Omri Kessel	admin	6/5/07	6/6/07 5:48 PM	Open	Incompl...	
ERP4 a...	111			Omri	Tom	6/8/07	6/11/07 11:18 AM	Open	Reserved	
foo				Jeff	admin	5/28/07	6/8/07 10:52 AM	Open	Incompl...	
I need s...	111			Omri kessel	admin	6/4/07	6/5/07 4:36 PM	Open	Reserved	
My Reqe...				Omri	admin	5/1/07	5/21/07 10:53 AM	Open	Reserved	
dsd	111			Omri	admin		6/6/07 11:28 AM	Open	Incompl...	
My Reque...				Jeff	oadmin		5/18/07 1:26 PM	Open	Reserved	
Request...				Jeff	oadmin		8/2/07 12:58 PM	Open	Reserved	

Figure 1: Request View

2. Display the tabs for Reservations and Reservation Violations. This is done by clicking on the icons on the lower left of the page:

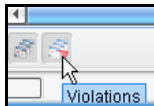


Figure 2: Detail Icons

The view changes then to show the added tabs:

3.1 Create a New Request

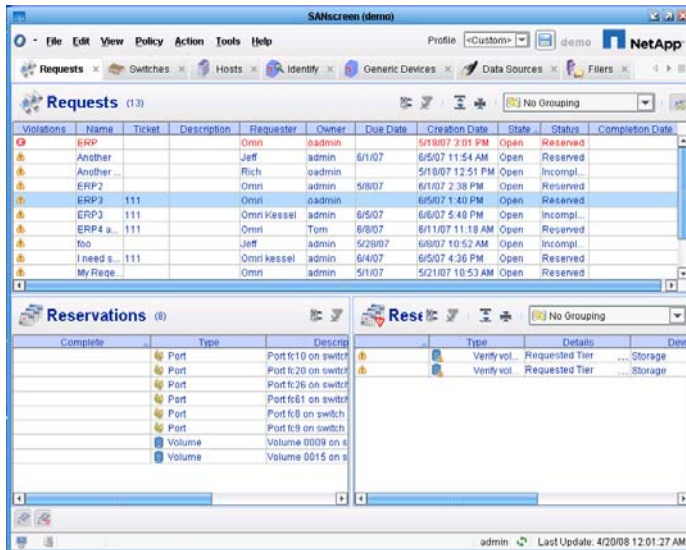


Figure 3: Requests with Detail Icons Maximized

3. Right-click anywhere in the Request view and click on Create Request, or select Create Request from the Action menu to open the following view:

The screenshot shows the 'Edit Request' dialog box. It has a title bar 'Edit Request' and a close button. The main area is titled 'Edit Service Request' and contains a red error message: 'Request name is empty.' Below this are several input fields: Name (with a red 'X' icon), Requester, Data Center (a drop-down menu), Business Unit (a drop-down menu), Creation Date (7/23/07 3:02 PM), Ticket, Owner (admin), E-mail, SAN (a drop-down menu), and Project (a drop-down menu). There are also tabs for Targets, Requirements, Reservations, and Notes. The 'Targets' tab is selected, showing 'Request target type:' with radio buttons for 'Host(s)' and 'Other'. Below this is a text area with instructions: 'Select a request target type. Select Host(s) type if the request is for single or multiple hosts. Select Other if the request is not for specific hosts.' At the bottom, there are summary statistics: 'Total Estimated Cost: 0', 'Total Actual Cost: 0', 'Satisfied Requirements: 0 / 0', and 'Completed Reservations: 0 / 0'. There are 'OK' and 'Cancel' buttons at the bottom right.

Figure 4: Creating a Request

4. Fill in the fields. Be aware that each selection has an impact on the way the request works. For example, if you select a Data Center or SAN from the drop-down box for the Data Center field, the Reservations that are made will only show resources that exist in that data center or that SAN. If you select a Business Unit or a project name in the drop-down box for that field, all the reserved resources

3.1 Create a New Request

will be annotated with the selected values. (See the *SANscreen Service Insight User Guide* for detailed information on Annotations.)

As another example, the Due Date should be as realistic as possible. Otherwise, there will be over-due violations that show up and impact the outcome of the request.

Email notification is sent (only) to the requestor concerning the status of the request.

5. Check that all fields have been entered. If a required field has been left empty, a hint is given at the top of the page. In the above illustration, you are informed that the request Name is empty. All required fields also show a red line to indicate they must be completed. Hover over that line with the mouse to see the issue. Until the fields are filled in, you cannot save the Request. Once the fields are complete, clicking on **OK** will save it.

The following shows a view with the all relevant fields entered:

The screenshot shows a 'View Request' dialog box with the following fields filled in:

- Name: Ferlang15
- Owner: admin
- Requester: ReginaC
- E-mail: admin@onaro.com
- Data Center: San Francisco
- SAN: Building
- Business Unit: PM
- Project: Caritas
- Creation Date: 10/11/07 1:33 PM
- Due Date: (empty)
- Ticket: (empty)

Below the fields are tabs for 'Targets', 'Requirements', 'Reservations', and 'Notes'. The 'Targets' tab is selected, showing a table of request target types. The 'Request target type' is set to 'Host(s)'.

New	Host Name	IP
kc_orawareh		192.1.168.24
Federer		192.1.168.30
csNAS1		192.1.168.13
nj_sap17		192.1.168.22
Becker		192.1.168.27

At the bottom, there are status indicators: Total Estimated Cost: 0, Total Actual Cost: 0, Satisfied Requirements: 0 / 0, and Completed Reservations: 0 / 0. There are 'OK' and 'Cancel' buttons at the bottom right.

Figure 5: Request Fields Completed

3.1.1 Target Tab

This tab allows targets to be defined for existing host(s), new hosts or applications:

1. You need to request a target type from the Target tab shown in Figure 5.
2. Click on the Host(s) or Other radio button.
3. Select Other if the request is for general SAN resources not associated with specific hosts—perhaps to reserve space for a migration planned for a later date but independent of hosts using the data. No other action is necessary.
4. Select Host(s) if any need to be added, using the following steps.

3.1 Create a New Request

- Under the Target tab, you can add a new host or an existing one by using the buttons on the lower right of the Request view, or you can add hosts that are members of a specified application.

3.1.1.1 Add Existing

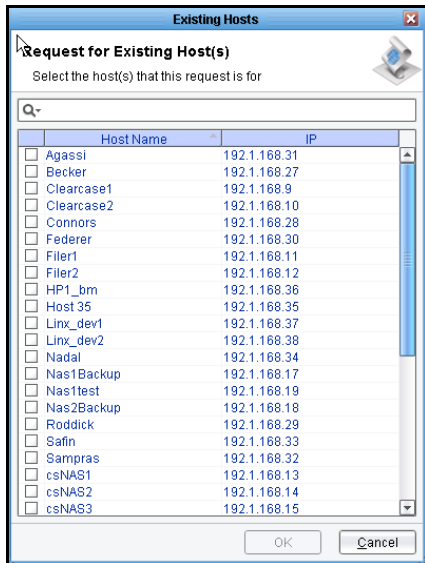


Figure 6: Adding Existing Hosts

3.1.1.2 Add by App...

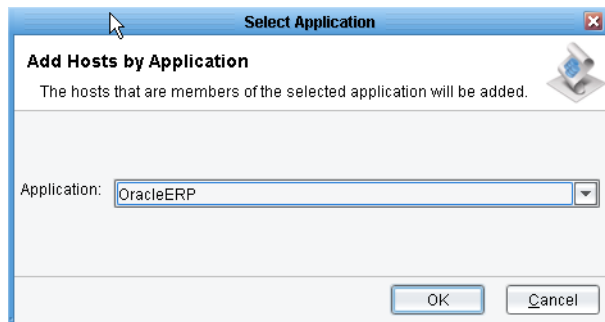
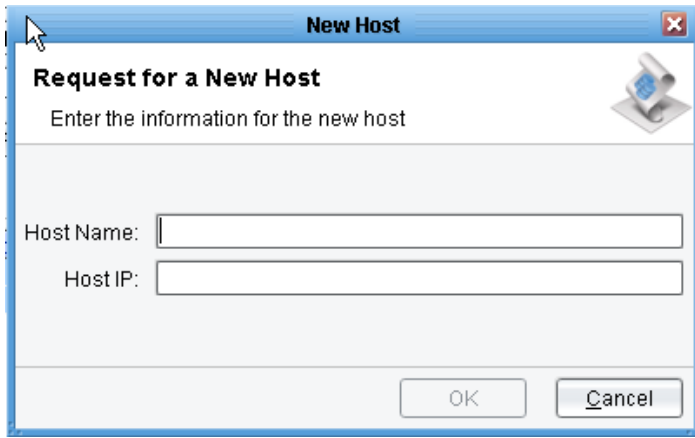


Figure 7: Adding Hosts by Application

All members of that application will be added if this option is selected.

3.1.1.3 Add New...



New Host

Request for a New Host

Enter the information for the new host

Host Name:

Host IP:

OK Cancel

Figure 8: Adding New Hosts

Select this button to add a host that will be brought on-line and which has not yet been detected by SANscreen. Once a new host has been added, the reminder message disappears. In addition, two new buttons are available that allow you to remove one or all hosts that have been added.

TheTarget tab now displays the hosts you have added. In addition, the message at the top left of the Request view reminds you need to add requirements to the request.

3.2 Setting Up Requirements

This process allows you to specify which switch ports are required, the amount, the size of LUNs and the tier level of service.

3.2.1 Requirements Tab

1. Click on the Requirements tab next:

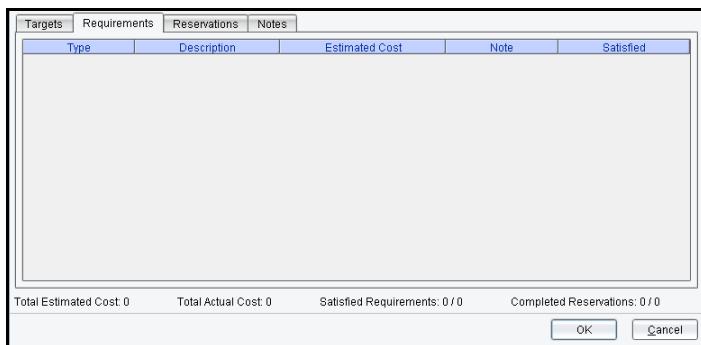


Figure 9: Requirements Tab

2. The two hosts that are to be connected each require 2 ports, 3 LUNs of 60GB at Tier 1, and 2 LUNs of 10GB at Tier 2. Use the buttons on the right of the tab to create those parameters.

3.2.1.1 Add Ports

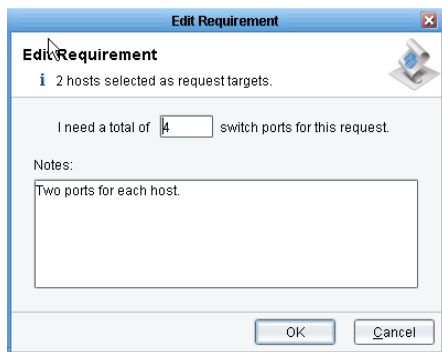


Figure 10: Adding Ports to Requirements

1. Enter the total number of ports, using the “Notes” section to indicate how those ports need to be assigned between the two hosts.
2. The information is then displayed on the Requirements tab, including the “Note” and Estimated Cost:

3.2 Setting Up Requirements

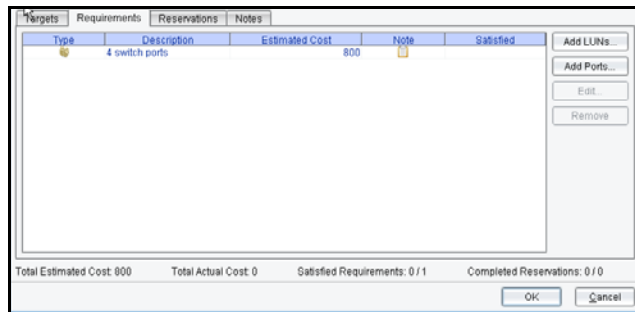


Figure 11: Ports Shown on Requirements Tab

Note

Note: Observe that at the bottom of the view cost information is supplied.

3.3.1.2 Add LUNs

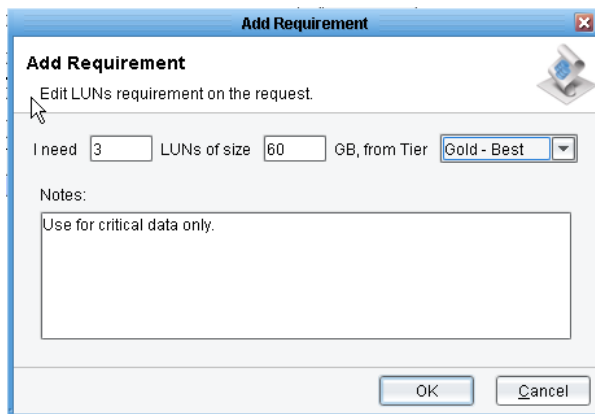
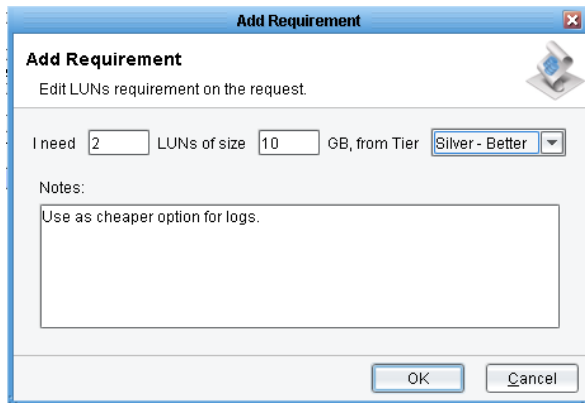


Figure 12: 3 LUNs of 60 GB Added to Requirements

1. The “Notes” section will be visible to the user. Use the “Notes” to set up conditions for these assignments.
2. Repeat the process for the second tier LUNs:

3.2 Setting Up Requirements



Add Requirement

Edit LUNs requirement on the request.

I need LUNs of size GB, from Tier

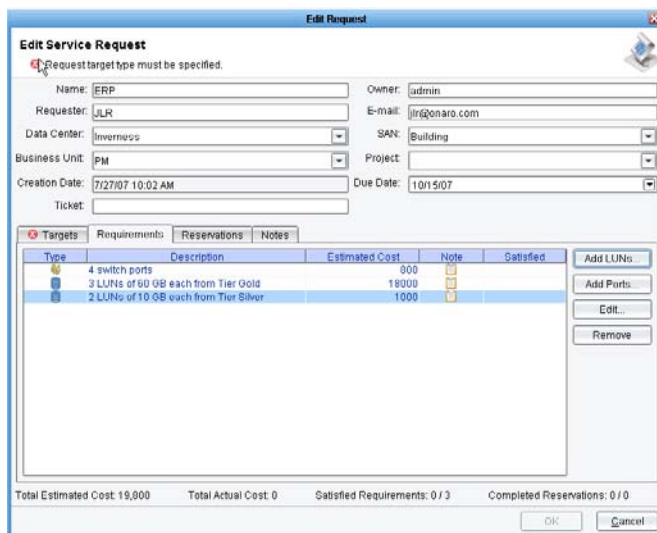
Notes:

Use as cheaper option for logs.

OK Cancel

Figure 13: 2 LUNs of 10GB Added to Requirements

The Requirements tab will now display your assignments:



Edit Service Request

Request target type must be specified.

Name: Owner:

Requester: E-mail:

Data Center: SAN:

Business Unit: Project:

Creation Date: Due Date:

Ticket:

Targets Requirements Reservations Notes

Type	Description	Estimated Cost	Note	Satisfied
4 switch ports		000		
3 LUNs of 60 GB each from Tier Gold		18000		
2 LUNs of 10 GB each from Tier Silver		1000		

Add LUNs...
Add Ports...
Edit...
Remove

Total Estimated Cost: 19,000 Total Actual Cost: 0 Satisfied Requirements: 0 / 3 Completed Reservations: 0 / 0

OK Cancel

Figure 14: LUNs Added to Requirements Tab

ADDENDUM: REQUEST PORTAL

Note

Note: Information for a request could come in to you through the SANscreen Request Portal. You could find the Request fields and the first two tabs already filled in. This can be accepted, revised, or rejected.

Chapter 4 Reserving Optimal Resources

Reservations in Capacity Manager allow you to manage the space allocation of different resources from multiple arrays or fabrics. This prevents unavailable or inaccurate information from impacting the SAN. It also prevents service interruptions and protects resources that have been already reserved.

Capacity Manager assists the SAN administrator in the decision process of matching SAN resources to satisfy the user's request.

The storage administrator starts the process of reserving resources to satisfy the requirements of the request. Based on the original request, the storage administrator needs essential information to do this, such as:

- What volumes are available in what physical location and that meet the LUN requirements?
- Which arrays are least utilized to use for the reservation?
- What connections to the fabric exist?
- What switch ports are available?

This chapter explains the specific procedures required to make a Reservation.

4.1 Make the Reservation

1. The icon message at the top of the view directs you to the Reservations tab. Here there is a tree that displays your current configuration:

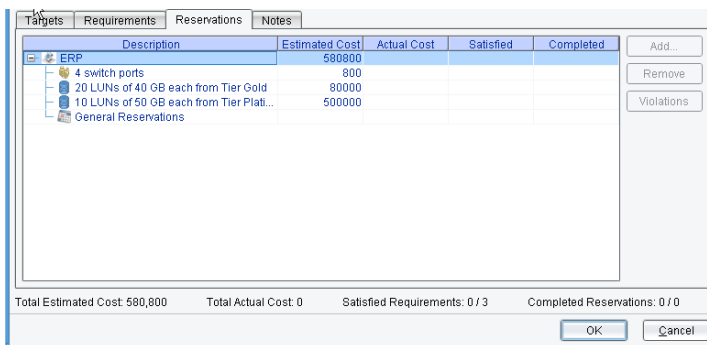


Figure 15: Reservations Tree

4.2 Satisfying a Requirement

2. The first element at the top of the tree shows the overall request and status. The next three levels display the requirements that have been added on for switch ports and LUNs. If you hover over any section of this tree, the “Notes” you added during the configuration appear in popups.

General Reservations:

The General Reservations icon lets you add in additional reservations for volumes, capacity or ports. For example, when you make requirements for the switch ports, you may need to set up an ISL. Under General Reservations, you can use the Add button to allocate new ports as part of the overall reservation description:

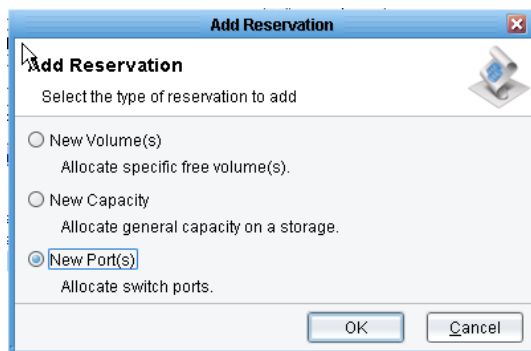


Figure 16: Using General Reservations

4.2 Satisfying a Requirement

Once you have successfully set up the Requirements in the Request view, you need to reserve resources.

1. Click on the Reservations tab and highlight one of the requirements you have created.
2. Click the Add button on the right. You have the option to choose to set up volumes or capacity for the type of reservation:

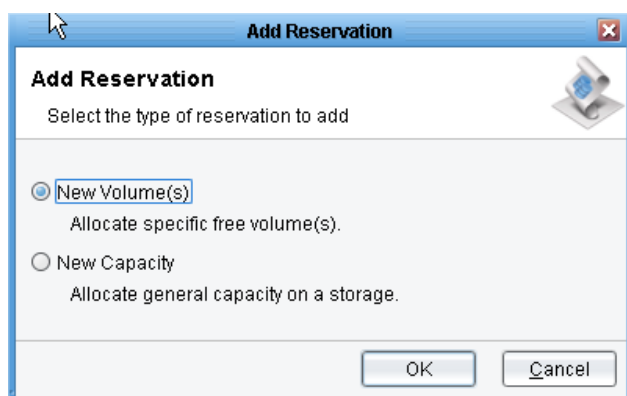


Figure 17: Adding a Reservation

4.2.1 Volume Requirements

- 1. Select New Volumes.
- 2. Note the icon on the upper right of the view.



This represents a filter that toggles the available storage arrays and volumes. Automated filtering helps you select resources that best map to the requirement.

If the filter is available, an icon also appears next to “Storage Array” on the view:



- 3. Click on the filter icon to see the available storage arrays. Highlight an array.
- 4. In the Volume section, notice that the Volume title also shows an icon, indicating that a filter is available. Click on the filter icon in the upper right of that section. The volumes associated with the highlighted storage array will display. Both sections are shown in the following illustration:

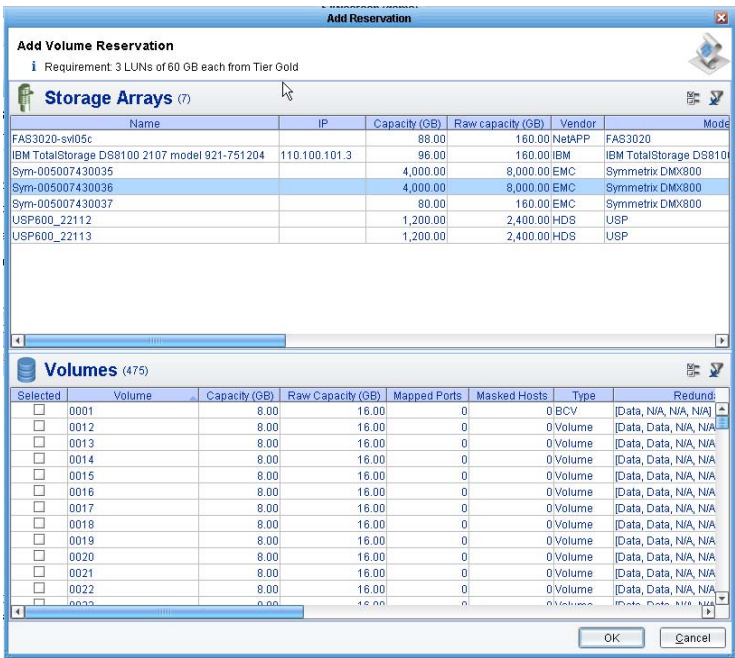


Figure 18: Volume Reservation

- 5. The volumes to be checked are those that meet the LUN requirements. Check the necessary volumes to meet the requirement, and click OK. The Reservation tab displays a green arrow next to the will show up on the Reservations tab to indicate that that requirement has been satisfied.

4.2 Satisfying a Requirement

4.2.2 Port Requirements

1. For ports, the process is very similar.

Highlight the port requirement in the Reservations tab. Click the Add button on the right to get the following view. This view is shown with the filter On:

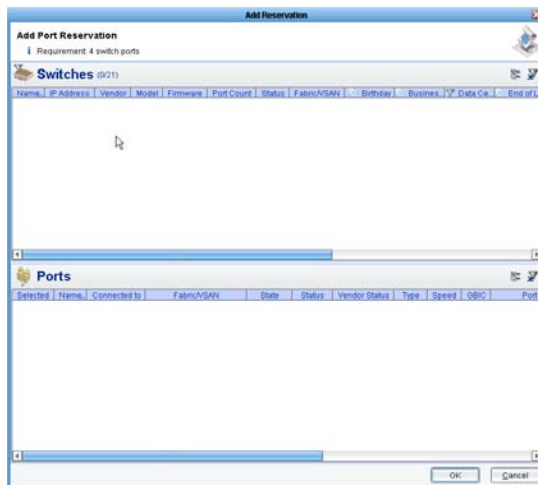


Figure 19: Reservation Switch Filter

2. Click on the filter icon for Switches to get the full list available. Highlight the switch to be used and then click on the filter icon in the Ports section. A full view of available ports is shown next:

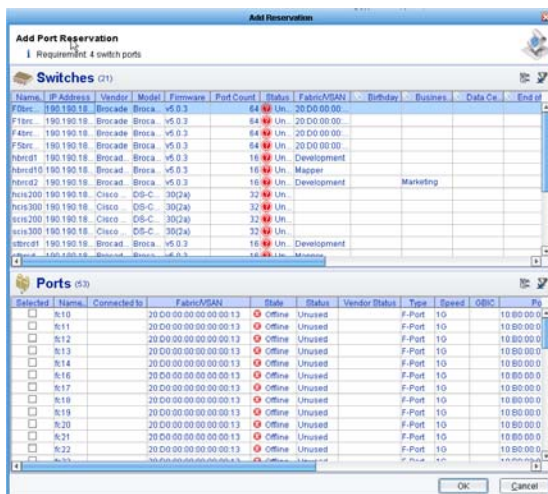


Figure 20: Port Reservation

3. Once the volumes and ports have been reserved successfully, the Reservations tab will show green arrows indicating all is well:

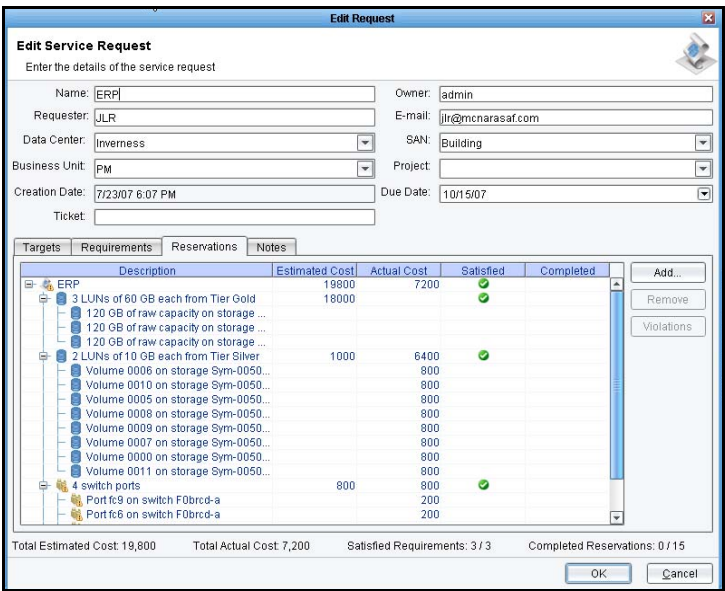


Figure 21: Satisfying Requirements in Reservation

However, returning to the Requests view on SANscreen, the Reservation Violations tab displays all four ports and classifies them as belonging to hot switches:

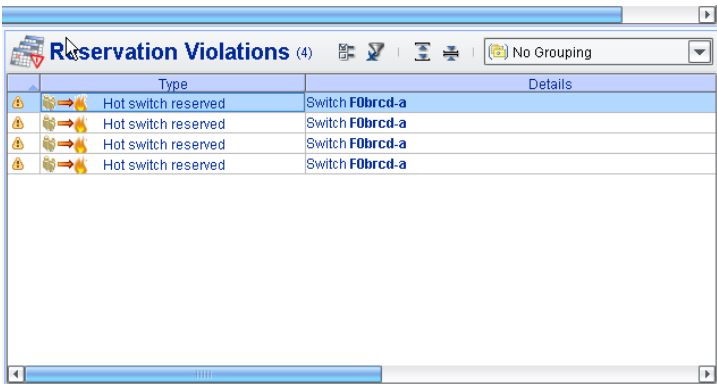


Figure 22: Reservation Violations

Because this means that the switches are too busy, they are not the best choice for the reservation process. In this event, you simply return to the Reservations tab, highlight the ports in question, click the Remove button on the right, and begin the process over again to reserve different ports.

The Requests view on SANscreen always shows you both successful results and reservation violations.

Chapter 5 Cancelling a Request

This chapter describes how to deallocate reservations.

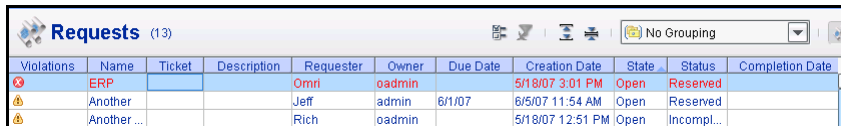
5.1 Perform a Deallocation of a Reservation

If the storage administrator receives a message to cancel a request that has already been allocated because the resources are no longer needed, then the following procedure applies:

Note

This may be important especially in the case where a request was partially or fully implemented or reserved. Canceling in an organized fashion addresses one of the major sources for orphan capacity that is created because of partial cancellations.

1. In the Requests view, highlight the request to be cancelled.



Violations	Name	Ticket	Description	Requester	Owner	Due Date	Creation Date	State	Status	Completion Date
	ERP			Omri	oadmin		5/18/07 3:01 PM	Open	Reserved	
	Another			Jeff	admin	6/1/07	6/5/07 11:54 AM	Open	Reserved	
	Another...			Rich	oadmin		5/18/07 12:51 PM	Open	Incompt...	

Figure 23: An Allocated Request

2. Right-click on the request and select Set Request State/Cancelled. The Requests view will display Cancelled in the State column.
3. The Reservations Violation view will indicate any allocated resources on the cancelled request. (If no reservations have been allocated, you can consider the request successfully cancelled.)

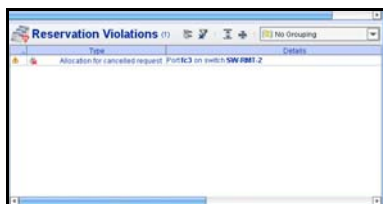


Figure 24: Confirmation of Cancelled Request

Note

NOTE: View the filter on the State column:



This list lets you display only the requests that are New and Open. However, you can access the cancelled request at any time and re-open it if desired by selecting Cancelled in the filter list. (Only if you remove the request does it completely disappear from the system.)

5.2 Remove a Request

If it becomes necessary to remove a request, perform the following procedure. The request must first show up as Cancelled in the State column.

For normal operation, requests stay in the system. They are needed for reports, so even when a request has been completed, it will not be removed. Thus, cancelling a request is not advised unless it is absolutely necessary, since it is a step that cannot be undone.

1. Go to the Requests view. Highlight the cancelled request that is to be removed.
2. Right-click the request. The following options appear:

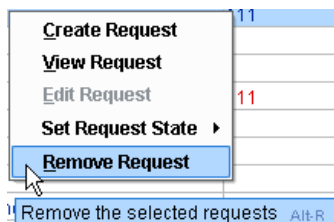


Figure 25: Remove Request Option

3. If you confirm that you do want the request removed by selecting that option, another popup appears asking for further confirmation:

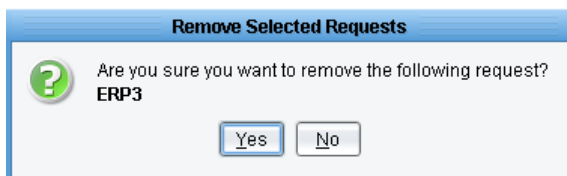


Figure 26: Removing a Request

4. Click on Yes to remove the request.

CAUTION: Remember that if the request has been removed, so all reserved resources have also been removed and are now available for others to use.

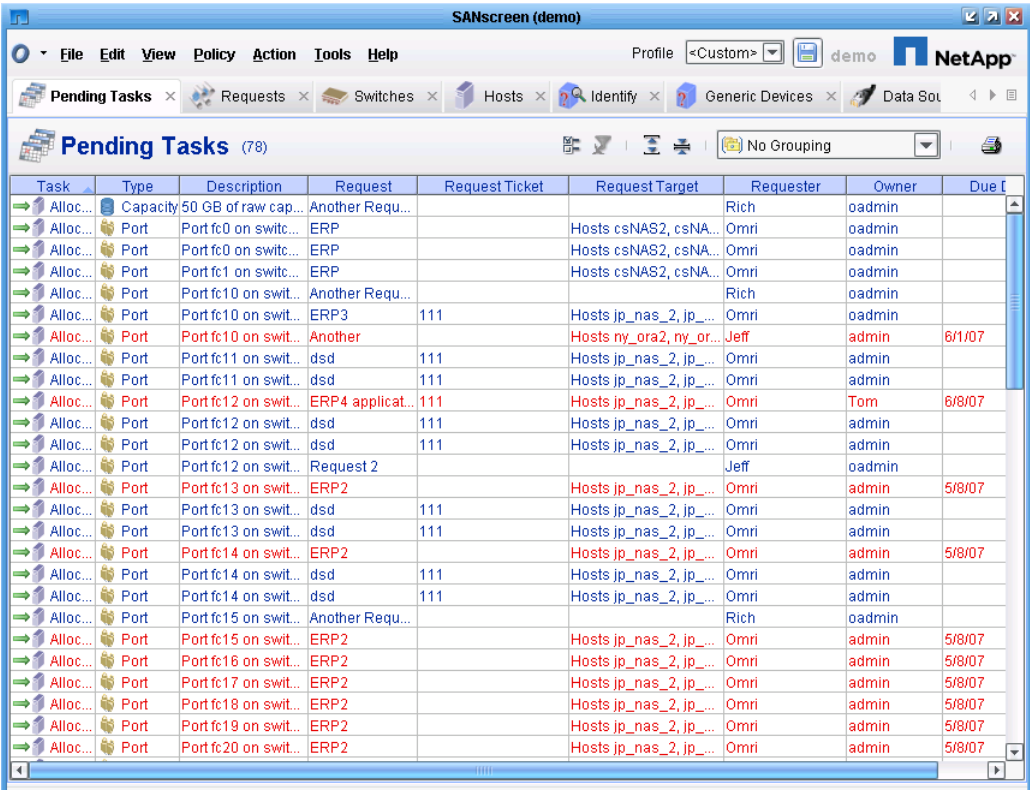
Chapter 6 Communicating the Status of Requests

Although the Requirements for the ERP extension have been satisfied, none of the storage resources have been allocated. Once allocation occurs, the Reservation process is Complete.

6.1 View Pending Tasks

To view tasks:

1. Click on the Pending Tasks icon in the SANscreen launch menu. The page that opens will show all Reservations that are pending:



Task	Type	Description	Request	Request Ticket	Request Target	Requester	Owner	Due Date
Alloc...	Capacity	50 GB of raw cap...	Another Requ...			Rich	oadmin	
Alloc...	Port	Port fc0 on switc...	ERP		Hosts csNAS2, csNA...	Omri	oadmin	
Alloc...	Port	Port fc0 on switc...	ERP		Hosts csNAS2, csNA...	Omri	oadmin	
Alloc...	Port	Port fc1 on switc...	ERP		Hosts csNAS2, csNA...	Omri	oadmin	
Alloc...	Port	Port fc10 on swit...	Another Requ...			Rich	oadmin	
Alloc...	Port	Port fc10 on swit...	ERP3	111	Hosts jp_nas_2, jp_...	Omri	oadmin	
Alloc...	Port	Port fc10 on swit...	Another		Hosts ny_or_2, ny_or...	Jeff	admin	6/1/07
Alloc...	Port	Port fc11 on swit...	dsd	111	Hosts jp_nas_2, jp_...	Omri	admin	
Alloc...	Port	Port fc11 on swit...	dsd	111	Hosts jp_nas_2, jp_...	Omri	admin	
Alloc...	Port	Port fc12 on swit...	ERP4 applicat...	111	Hosts jp_nas_2, jp_...	Omri	Tom	6/8/07
Alloc...	Port	Port fc12 on swit...	dsd	111	Hosts jp_nas_2, jp_...	Omri	admin	
Alloc...	Port	Port fc12 on swit...	Request 2			Jeff	oadmin	
Alloc...	Port	Port fc13 on swit...	ERP2		Hosts jp_nas_2, jp_...	Omri	admin	5/8/07
Alloc...	Port	Port fc13 on swit...	dsd	111	Hosts jp_nas_2, jp_...	Omri	admin	
Alloc...	Port	Port fc13 on swit...	dsd	111	Hosts jp_nas_2, jp_...	Omri	admin	
Alloc...	Port	Port fc14 on swit...	ERP2		Hosts jp_nas_2, jp_...	Omri	admin	5/8/07
Alloc...	Port	Port fc14 on swit...	dsd	111	Hosts jp_nas_2, jp_...	Omri	admin	
Alloc...	Port	Port fc14 on swit...	dsd	111	Hosts jp_nas_2, jp_...	Omri	admin	
Alloc...	Port	Port fc15 on swit...	Another Requ...			Rich	oadmin	
Alloc...	Port	Port fc15 on swit...	ERP2		Hosts jp_nas_2, jp_...	Omri	admin	5/8/07
Alloc...	Port	Port fc16 on swit...	ERP2		Hosts jp_nas_2, jp_...	Omri	admin	5/8/07
Alloc...	Port	Port fc17 on swit...	ERP2		Hosts jp_nas_2, jp_...	Omri	admin	5/8/07
Alloc...	Port	Port fc18 on swit...	ERP2		Hosts jp_nas_2, jp_...	Omri	admin	5/8/07
Alloc...	Port	Port fc19 on swit...	ERP2		Hosts jp_nas_2, jp_...	Omri	admin	5/8/07
Alloc...	Port	Port fc20 on swit...	ERP2		Hosts jp_nas_2, jp_...	Omri	admin	5/8/07

Figure 27: Pending Tasks View

2. The Reservations are not completed until there is a corresponding physical setup of the equipment at the data center—in this case, Inverness. Once that is done, the Reservations disappear from the Pending Tasks view.

In the Requests view, the Status field automatically changes from Reserved to Allocated. (Once allocated, the request can be marked as complete by right-clicking on the request, selecting Set Request State from the popup, and choosing *Complete*.)

Chapter 7 Reports

IT organizations function better—more efficiently--when they are run like a business and charge for the services they provide. In Capacity Manager, three reports allow insight into the data that describes the allocation and deallocation of tier resources and the status of the SAN infrastructure.

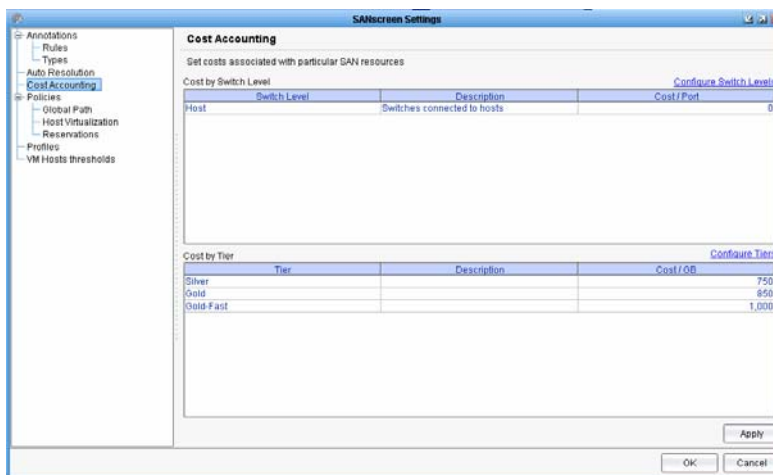
The reports include **Storage Utilization**, **Chargeback** and a historical view of the SAN in an audit report—the **Allocation Log**.

Using these reports, the SAN administrator has a way to apply cost to resources associated with Business Units in the organization. Excess storage can be returned to the SAN and reclaimed for use in another area that may be under-utilized, or reserved for future planning needs

7.1 Cost Accounting

Before creating the Storage Utilization report, you need to assign cost values to the tiers and switch levels (core, host and storage) you have defined under Annotation Types (see the *Service Insight User Guide*). This is done in Cost Accounting, where SANscreen automates the process of tracking, recording and analyzing costs associated with the tiers.

1. Select the **Tools/SANscreen Settings** in the top menu:
2. Select **Cost Accounting**:



3. Double-click on the cost values that appear in the far right columns to convert the field to an **Edit** pane.
4. Make the desired changes to the values.

Once you have categorized your equipment into tiers, SANscreen Annotations lets you manage and partition this information and collect it into a **Storage Utilization** report and a **Capacity Management** report, both accessed through the SANscreen portal and found under SAN Utilization and Capacity Management categories, respectively.

7.2 Storage Utilization Report

The Storage Utilization section of the report lets you maintain an optimum level of resource usage over time by providing tier utilization information for planning and purchasing decisions:

The Storage Utilization report is composed of two sections:

1. **Summary section** for each data center, providing a high-level view of storage utilization categorized by tier level (protection level).
2. **Detailed section** for each data center, providing a list of storage arrays and their utilization status categorized by tier level (protection level).

The Storage Utilization report takes into account the physical location of devices based on data center.

The report uses a color scheme of green, yellow, and red to visualize status:

7.2 Storage Utilization Report

Storage Utilization Report										
	Austin	Boston	San Francisco	Other Data Center						
Tier Bronze Storage GB Available :	0	928	0	0						
Tier Gold Storage GB Available :	208	0	0	0						
Tier Platinum Storage GB Available :	1888	0	0	0						
Tier Silver Storage GB Available :	0	14704	0	144						
Uncategorized Tier Storage GB Available :	1984	0	0	0						
Uncategorized Storage GB Available :	1984	0	0	0						
Total Capacity:	5120	16000	48	176						
Allocated Capacity:	1008	560	48	32						
Status:	Green	Green	Red	Green						
Austin										
Frame	Vendor	Model	Tier	# of volumes	Total Storage	Storage Allocated	Storage Reserved	Storage Available	Utilization Allocated	Status
FAS3020-sw05c	NetAPP	FAS3020			160	0	0	160	0.00%	Green
			Uncategorized Tier	11	0	0	0	0	0.00%	Green
IBM TotalStorage DS8100 2107 model 921-751204	IBM	IBM TotalStorage DS8100 2107 model 921			160	80	96	80	50.00%	Yellow
USP600_22112	HDS	USP	Tier Gold	12	192	80	96	208	41.67%	Yellow
					2400	512	0	1888	21.33%	Green
USP600_22113	HDS	USP	Tier Platinum	150	2400	512	0	1888	21.33%	Green
					2400	416	0	1984	17.33%	Green
			Uncategorized Tier	150	2400	416	0	1984	17.33%	Green
Boston										
Frame	Vendor	Model	Tier	# of volumes	Total Storage	Storage Allocated	Storage Reserved	Storage Available	Utilization Allocated	Status
Sym-005007430035	EMC	Symmetrix DMX800			8000	384	144	7616	4.80%	Green
			Tier Bronze	45	720	0	32	752	0.00%	Green
			Tier Silver	455	7280	384	112	7008	5.27%	Green
Sym-005007430036	EMC	Symmetrix DMX800			8000	176	48	7824	2.20%	Green
			Tier Bronze	13	208	32	0	176	15.38%	Green
			Tier Silver	487	7792	144	48	7696	1.85%	Green
San Francisco										
Frame	Vendor	Model	Tier	# of volumes	Total Storage	Storage Allocated	Storage Reserved	Storage Available	Utilization Allocated	Status
Sym-005007430039	EMC	Symmetrix DMX800			16	16	0	0	100.00%	Red
			Tier Bronze	1	16	16	0	0	100.00%	Red
Sym-0050074300310	EMC	Symmetrix DMX800			16	16	0	0	100.00%	Red
			Tier Silver	1	16	16	0	0	100.00%	Red
Sym-0050074300311	EMC	Symmetrix DMX800			16	16	0	0	100.00%	Red
			Tier Silver	1	16	16	0	0	100.00%	Red
Other Data Center										
Frame	Vendor	Model	Tier	# of volumes	Total Storage	Storage Allocated	Storage Reserved	Storage Available	Utilization Allocated	Status
Sym-005007430037	EMC	Symmetrix DMX800			160	16	0	144	10.00%	Green
			Tier Silver	10	160	16	0	144	10.00%	Green
Sym-005007430038	EMC	Symmetrix DMX800			16	16	0	0	100.00%	Red
			Tier Silver	1	16	16	0	0	100.00%	Red

10.6.2.1 Report Details

Each Business Unit in the sample report shown on the previous page is detailed according to the following parameters:

Frame:	Array name as discovered by SANscreen
Vendor:	EMC, HP, etc.
Model:	Firmware, microcode
Tier:	Tier level (Gold, Silver, Bronze)
# of volumes:	How many volumes at that tier level
Total Storage:	Total raw capacity used for that level
Storage Allocated:	Raw capacity in use by this tier level
Storage Reserved:	Raw capacity reserved on the storage
Storage Available:	Total storage/Storage allocated
Utilization Allocated:	Storage allocated/Total storage percentage
Status:	Red, Yellow or Green according to the configuration file threshold levels

The Summary and Development sections of the Capacity Management report show the chargeback information.

Summary

Tier	Cost (1GB)
Gold	100
Bronze	25
Silver	50

Business Unit	Gold		Bronze		Silver		Total	
	Capacity	Cost	Capacity	Cost	Capacity	Cost	Capacity	Cost
Dev	24	2400	0	0	8	400	32	2800
Total	24	2400	0	0	8	400	32	2800

Tier	Cost (1GB)
Gold	100
Bronze	25
Silver	50

Date	Gold				Bronze			
	Capacity	Cost	Capacity Diff	Cost Diff	Capacity	Cost	Capacity Diff	Cost Diff
2006-09-01	0	0	0	0	0	0	0	0
2006-10-01	0	0	0	0	0	0	0	0
2006-11-01	0	0	0	0	0	0	0	0
2006-12-01	0	0	0	0	0	0	0	0
2007-01-01	0	0	0	0	0	0	0	0
2007-02-01	0	0	0	0	0	0	0	0
2007-03-01	0	0	0	0	0	0	0	0
2007-04-01	24	2400	24	2400	0	0	0	0
2007-05-01	24	2400	0	0	0	0	0	0
2007-06-01	24	2400	0	0	0	0	0	0
2007-07-01	24	2400	0	0	0	0	0	0
2007-08-01	24	2400	0	0	0	0	0	0
Total		12000	24	2400		0	0	

Silver				Total			
Capacity	Curt	Capacity Diff	Curt Diff	Capacity	Curt	Capacity Diff	Curt Diff
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	24	2400	24	24000
\$	400	\$	400	32	2800	\$	4000
\$	400	0	0	32	2800	0	0
\$	400	0	0	32	2800	0	0
\$	400	0	0	32	2800	0	0
				152	13600	32	2800

7.4 Allocation Audit Report

Once the tiers have been set up and their cost associated and applied to SAN resources, all that data is placed into reports, automatically. Using the data from the **Allocation Audit Report**, the storage administrator not only sees which volumes have been placed where, but why it happened and for whom it had been assigned. If any allocation has been made without going through the Request system, red lines in the report signify that the allocations were not properly controlled.

The Allocation Log section of the report shows what has been allocated throughout the SAN, giving a current and historical view. Red indicates incorrect allocations that do not belong to any Reservations.

Allocations between 2006-07-30 16:01:18 and 2007-08-06 16:01:18

Allocation Audit Log

Ticket Number	Requester	Date	Business Unit	Application	Host	Storage	Volume	Tier	Capacity (GB)	Raw Capacity (GB)	Request
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22113	00:24		0	16	
2007-03-14 09:31:25			OracleERP	ny_oral	USP600	22113	00:0C		0	16	
2007-03-14 09:31:25			WebSite	ny_ptrTest1	USP600	22113	00:11		0	16	
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22112	00:24		0	16	
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22113	00:65		0	16	
2007-03-14 09:31:25			OracleERP	ny_oral2	USP600	22112	00:00		0	16	
2007-04-03 14:31:37			Mapping Service	Federex	Sym-005007430037		0001	Silver	0	16	
2007-03-14 09:31:25			Exchange	ny_exch2	USP600	22112	00:15		0	16	
2007-03-14 09:31:25			OracleERP	ny_oral2	USP600	22113	00:00		0	16	
2007-03-14 09:31:25			WebSite	ny_ptrTest1	USP600	22112	00:1B		0	16	
2007-03-14 09:31:25			Exchange	ny_exch2	USP600	22113	00:00		0	16	
2007-04-03 14:31:30				n3_exch002	Sym-005007430035		0014	Silver	0	16	
2007-03-14 09:31:25			Exchange	ny_exch2	USP600	22113	00:15		0	16	
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22112	00:19		0	16	
2007-04-03 14:31:30				kc_oraldev7	Sym-005007430036		0230	Silver	0	16	
2007-03-14 09:31:25			WebSite	ny_ptrTest1	USP600	22112	00:25		0	16	
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22113	00:23		0	16	
2007-04-03 14:31:30				n3_oral_hld	Sym-005007430035		0365	Silver	0	16	
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22113	00:64		0	16	
2007-04-03 14:31:37			Mapping Service	Agassi	Sym-005007430037		0001	Silver	0	16	
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22113	00:1A		0	16	
2007-03-14 09:31:25			Exchange	ny_exch1	USP600	22113	00:1E		0	16	
2007-04-03 14:31:30				kc_oraldev7	Sym-005007430036		0200	Silver	0	16	
2007-04-03 14:31:37			Mapping Service	Connors	Sym-005007430037		0001	Silver	0	16	
2007-03-14 09:31:25			Japan	jp_nas_1	USP600	22112	00:64		0	16	
2007-04-03 14:31:30				n3_map17	Sym-005007430035		0345	Silver	0	16	
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22112	00:1A		0	16	
2007-04-03 14:31:37			Mapping Service	Safin	Sym-005007430037		0001	Silver	0	16	
2007-04-03 14:31:30				n3_map17	Sym-005007430035		0000	Silver	0	16	
2007-03-14 09:31:25			Japan	jp_nas_1	USP600	22113	00:65		0	16	
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22113	00:65		0	16	
2007-03-14 09:31:25			OracleERP	ny_oral2	USP600	22113	00:17		0	16	
2007-04-03 14:31:30				Line dev1	Sym-030074300310		0000	Silver	0	16	
2007-03-14 09:31:25			Japan	jp_nas_2	USP600	22112	00:0F		0	16	
2007-03-14 09:31:25			OracleERP	ny_oral	USP600	22112	00:20		0	16	
2007-03-14 09:31:25			Exchange	ny_exch1	USP600	22112	00:0A		0	16	
2007-03-14 09:31:25			DevTrack	cnNAS2	FAS3020-sv105c		/VOL/APP/LUN3		0	0	
2007-03-14 09:31:25			WebSite	ny_ptrTest1	USP600	22113	00:64		0	16	
2007-04-03 14:31:30				n3_map17	Sym-005007430035		0343	Silver	0	16	
2007-03-14 09:31:25			WebSite	ny_ptrTest1	USP600	22112	00:07		0	16	
2007-03-14 09:31:25			OracleERP	ny_oral	USP600	22113	00:00		0	16	

The following categories in this report apply to Capacity Management:

- <Date (when record was created)
- <Business Unit
- <Application
- <Host
- <Storage
- <Volume
- <Tier
- <Capacity (GB)
- <Raw Capacity (GB)

External Application Integration

Service Connect API enables NetApp customers and Independent Software Vendors (ISVs) to develop the Annotations application on top of the SANscreen change and service repository and engine. This application can then use the API to integrate SANscreen into existing management systems and processes. The API provides read-only access into the information that resides in the SANscreen repository.

To use the API, you must have a current version of SANscreen, a SANscreen license, and a separate API license. The API is based on SOAP and Web services. This provides the flexibility to use the API from almost any programming environment, including Java and Perl.

This chapter discusses the fields available to support this integration in your own SAN.

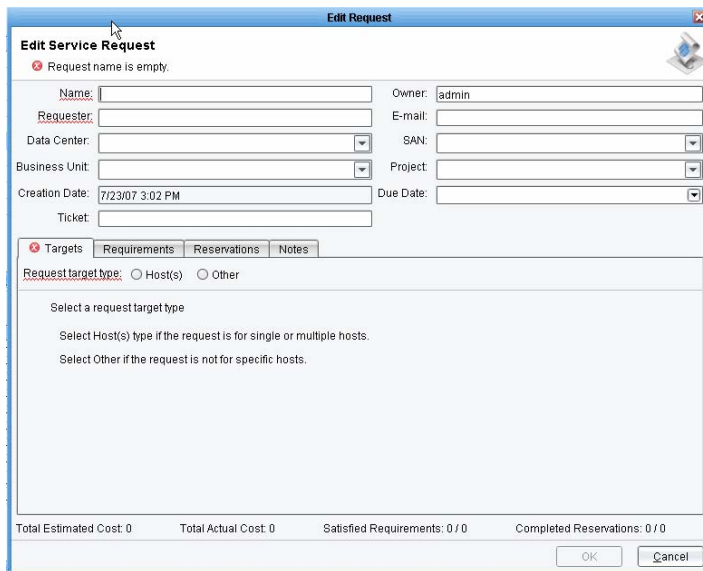
External System Integration – Capacity Manager

Overview

The following objects are important for integration of Capacity Manager to external information systems. Each section describes the object and its role.

Request

A request (seen in the API as ReservationRequest) is an object representing a user request. A request holds the general request information:



The request must have at least one requirement. Each requirement has a state:

- New
- Open
- Completed
- Cancelled
- Rejected

Querying the state periodically may be important in order to track the progress of the request and update external information systems.

Target

A target is a host for which the new capacity or resources are required. The relationship between a request and a target is 1:N. The target can be an existing host or a new host that will be added in the future.

Requirement

A requirement is a definition of what should be provided. Requirements can be of type “LUN” or “PORT”. For Luns, the number of Luns, size of Luns and tier can be indicated (LUN_COUNT, LUN_SIZE and TIER_ID). For ports, the number of required ports can be indicated (PORT_COUNT). The relationship between a request and a requirement is 1:N.

In order to start implementation, all requirements should be in a satisfied state. Querying the state of each of the requirements can provide information about which ones are still pending, and require SAN team intervention (reserving the required resources).

Use Scenarios

The use scenarios described here focus on the usage requests, requirements and targets.

Creating a new request

Description: Customer wants to create a new request in the system as a result of an event that happened in an external system.

Examples:

1. The customer is using a system that allows server side IT automation. The customer wants a server decommission request to create a request within the Capacity Manager to reclaim all resources related to the decommissioned server by writing a script that reads the recent action with the server side IT automation and creates a new request in the Capacity Manager.
2. The customer uses using a ticketing system to open IT-related requests. The customer wants to write a script that takes the request from the ticketing system and opens a similar request in the Capacity Manager.

Tracking a past request

Description: The customer wants to update periodically an external system with the status of a request in the Capacity Manager.

Examples:

1. The customer is using a ticketing system and would like to write code that would periodically read the requests in the Capacity Manager, take the request status and update it in the corresponding ticket in the ticketing system. The customer is interested in status changes and completion of request.

Code Samples

The samples provide a quick way to start to code application and business unit manipulations. Although the sample is in Java, the underlying communication layer is Web Services SOAP. As a result, you can use the API form and language that supports SOAP (e.g. Perl, .net, VB, C/C++ etc.).

Disclaimer

These samples are provided as is. Care should be taken while using the API because information in SANSscreen database will be changed. **Be advised that changes cannot be undone.** Make sure to back up the SANSscreen database before running any API-related script.

Building

The sample code below requires the following:

1. License to use the API. Verify that by surfing to the SANSscreen server, click 'license manager' and verify that the license is "ok" and that the API box is checked (see table below):

License Manager

Licensed Modules:

Module	Quantity	API	Expires	Serial Number
Application Insight: Array Performance	1000 TB	✓	Oct 8, 2008	123456
Application Insight: FC Switch	1000 Ports	✓	Oct 8, 2008	123456
Capacity Manager	1000 TB	✓	Oct 8, 2008	123456
Service Assurance	1000 Ports	✓	Oct 8, 2008	123456
Service Insight: FC Port	1000 Ports	✓	Oct 8, 2008	123456
Service Insight: NAS TB	1000 TB	✓	Oct 8, 2008	123456
VM Insight	1000 TB	✓	Oct 8, 2008	123456

Add...

Import...

Remove

2. JDK1.5 installed.
3. Download the code samples from the SANSscreen portal → SANSscreen Connect API → Examples
4. Open the examples under c:\temp
5. When compiling and running the following jar files (java libraries) should be in the classpath:
 - a. C:\TEMP\sansscreenapi-examples\java\lib\activation.jar
 - b. C:\TEMP\sansscreenapi-examples\java\lib\FastInfoset.jar
 - c. C:\TEMP\sansscreenapi-examples\java\lib\java-sdk.jar
 - d. C:\TEMP\sansscreenapi-examples\java\lib\jaxrpc-impl.jar
 - e. C:\TEMP\sansscreenapi-examples\java\lib\jaxrpc-spi.jar
 - f. C:\TEMP\sansscreenapi-examples\java\lib\jbossall-client.jar
 - g. C:\TEMP\sansscreenapi-examples\java\lib\jsr173_api.jar

- h. C:\TEMP\sansscreenapi-examples\java\lib\mail.jar
- i. C:\TEMP\sansscreenapi-examples\java\lib\saaj-impl.jar
- j. C:\TEMP\sansscreenapi-examples\java\lib\xercesImpl.jar

Verification

In order to verify the effect and correctness of the sample code, the developer can have a SANSscreen client open and directed at the same SANSscreen server as the API code / script. Any changes in the client should be reflected immediately in the API read results and any API change actions should be presented immediately in the SANSscreen client UI.

Example 1 – Application and Business Unit

Before running that sample, make sure to backup the SANSscreen database.

Overview

The following sample does the following things:

1. Request is created.
2. Requirements are added to the request.
3. Request is updated with new set of data.
4. Requirements of the request are updated with new data.
5. All requirements of the request are removed.

Code Sample

File: ApplicationAndBusinessUnit.java

```
import com.onaro.sansscreen.wsapi.*;

import com.onaro.sansscreen.wsapi.*;
import com.sun.xml.rpc.client.StubBase;
import transport.HttpClientTransportFactory;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.xml.rpc.Stub;
import java.util.HashMap;
import java.util.Map;

/**
 * This example illustrates how to manage requests/requirements as part of provisioning
 * manager
 * product. The results of the operations are printed out.
 *
 * 1. Request is created
 * 2. Requirements are added to the request
 * 3. Request is updated with new set of data
 * 4. Requirements of the request are updated with new data
 */
```

```

* 5. All requirements of the request are removed
*/
public class CapacityRequestExample {

    private SANscreenAPIEndPoint endPoint;

    public CapacityRequestExample() throws Exception {
        endPoint = init();
    }

    private SANscreenAPIEndPoint init() throws Exception {
        //needed for ssl
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null,
            new TrustManager[] { new AcceptAllTrustManager() }, null);

        String serviceUrl = "http://localhost/sanscreenapi";
        System.setProperty("javax.xml.soap.MessageFactory",
            "com.sun.xml.messaging.saaj.soap.ver1_1.SOAPMessageFactory1_1Impl");

        SANscreenAPIEndPoint endPoint = new SANscreenAPIService_Impl()
            .getSANscreenAPIEndPointPort();
        ((Stub) endPoint)._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY,
            serviceUrl);
        ((Stub) endPoint)._setProperty(Stub.USERNAME_PROPERTY, "admin");
        ((Stub) endPoint)._setProperty(Stub.PASSWORD_PROPERTY, "onaro");
        ((StubBase) endPoint)._setTransportFactory(new
            HttpClientTransportFactory(sslContext.getSocketFactory()));
        return endPoint;
    }

    private ReservationRequest[] createRequests() throws Exception {
        ReservationRequest [] requests;

        APISessionContext context = endPoint.openSession("1.0", 10000);

        // request is targeted at two new hosts, we will put reservation requirements for
        // ports and luns for those hosts later
        ReservationRequestTarget [] newTargets = new ReservationRequestTarget[2];
        Attribute [] attributes = new Attribute[3];
        attributes[0] = new Attribute("IP_ADDRESS", "127.3.4.6");
        attributes[1] = new Attribute("HOST_NAME", "host 3");
        attributes[2] = new Attribute("IS_NEW", "true");
        newTargets[0] = new ReservationRequestTarget();
        newTargets[0].setType("HOST");
        newTargets[0].setAttributes(attributes);

        attributes[0] = new Attribute("IP_ADDRESS", "192.1.168.4");
        attributes[1] = new Attribute("HOST_NAME", "host 1");
        attributes[2] = new Attribute("IS_NEW", "true");
        newTargets[1] = new ReservationRequestTarget();
        newTargets[1].setType("HOST");

```

```

newTargets[1].setAttributes(attributes);

long dueDate = System.currentTimeMillis() + 10000;

ReservationRequest newRequest = new ReservationRequest();
newRequest.setId("-1");
newRequest.setDisplayName("Request 1");
newRequest.setDescription("Request Description 1");
newRequest.setOwner("Owner 1");
newRequest.setRequester("Requester 1");
newRequest.setRequesterEmail("Email@requester.com");
newRequest.setJustification("Justification 1");
newRequest.setComments("Comments 1");
newRequest.setTargetType("HOST");
newRequest.setState("NEW");
newRequest.setRequestTargets(newTargets);
newRequest.setProjectId(1);
newRequest.setSanId(2);
    newRequest.setDataCenterId(1);
newRequest.setTicket("Ticket 1");
newRequest.setDueDate(dueDate);

endPoint.addReservationRequests(context, new ReservationRequest [] {newRequest});
requests = endPoint.getReservationRequests(context);

    return requests;
}

private ReservationRequest[] updateRequests(ReservationRequest request) throws
Exception {
    ReservationRequest [] requests;
    APISessionContext context = endPoint.openSession("1.0", 10000);
    long dueDate = System.currentTimeMillis() + 10000;

    request.setId(request.getId());
    request.setDisplayName("Request 2");
    request.setDescription("Request Description 2");
    request.setOwner("Owner 2");
    request.setRequester("Requester 2");
    request.setRequesterEmail("Email@requester.com2");
    request.setJustification("Justification 2");
    request.setComments("Comments 2");
    request.setTargetType("HOST");
    request.setState("NEW");
    request.setRequestTargets(null);
    request.setProjectId(2);
    request.setSanId(3);
    request.setDataCenterId(2);
    request.setTicket("my ticket 2");
    request.setDueDate(dueDate + 1);

    endPoint.updateReservationRequests(context, new ReservationRequest [] {request});
    requests = endPoint.getReservationRequests(context);
}

```

```

        return requests;
    }

    private ReservationRequirement[] createRequirements(ReservationRequest request)
throws Exception {
        ReservationRequirement[] requirements;

        APISessionContext context = endPoint.openSession("1.0", 10000);

        // requirement for luns - 2 luns 30GB each of tier (identified by id 3) - use annotations API
        // to obtain tier id
        ReservationRequirement req1 = new ReservationRequirement();
        req1.setType("LUN");
        req1.setNote("Note 1");
        Attribute [] attributes = new Attribute[3];
        attributes[0] = new Attribute();
        attributes[0].setName("LUN_COUNT");
        attributes[0].setValue("2");
        attributes[1] = new Attribute();
        attributes[1].setName("LUN_SIZE");
        attributes[1].setValue("30");
        attributes[2] = new Attribute();
        attributes[2].setName("TIER_ID");
        attributes[2].setValue("3");
        req1.setAttributes(attributes);

        // requirement for ports - 2 ports are required for new hosts
        ReservationRequirement req2 = new ReservationRequirement();
        req2.setType("PORT");
        req2.setNote("Note 1");
        Attribute [] portAttributes = new Attribute[1];
        portAttributes[0] = new Attribute();
        portAttributes[0].setName("PORT_COUNT");
        portAttributes[0].setValue("2");
        req2.setAttributes(portAttributes);

        endPoint.addReservationRequirements(context, request.getId(), new
ReservationRequirement [] {req1, req2});
        requirements = endPoint.getReservationRequirements(context, request.getId());

        return requirements;
    }

    private ReservationRequirement[] updateRequirements(ReservationRequest request)
throws Exception {
        ReservationRequirement[] requirements;
        APISessionContext context = endPoint.openSession("1.0", 10000);

        requirements = endPoint.getReservationRequirements(context, request.getId());

        for (ReservationRequirement req:requirements) {
            req.setNote("Note 2");

```

```

        Attribute [] attributes = req.getAttributes();

        Map<String, String> attributesMap = new HashMap<String, String>();

        for (Attribute attr : attributes) {
            attributesMap.put(attr.getName(), attr.getValue());
        }

        // we are updating requirement to be 3 luns 40 GB each of a tier (id 5)
        if (attributesMap.get("LUN_COUNT") != null) {
            attributesMap.put("LUN_COUNT", "3");
            attributesMap.put("LUN_SIZE", "40");
            attributesMap.put("TIER_ID", "5");
        } else if (attributesMap.get("PORT_COUNT") != null) {
            // 3 ports are required for new hosts
            attributesMap.put("PORT_COUNT", "3");
        }

        for (Attribute attr : attributes) {
            String attributeVal = attributesMap.get(attr.getName());
            attr.setValue(attributeVal);
        }

        req.setAttributes(attributes);

        endPoint.updateReservationRequirements(context, request.getId(), new
ReservationRequirement [] {req});
    }
    requirements = endPoint.getReservationRequirements(context, request.getId());

    return requirements;
}

private ReservationRequirement[] removeRequirements(ReservationRequest request)
throws Exception {
    APISessionContext context = endPoint.openSession("1.0", 10000);
    ReservationRequirement[] requirements =
endPoint.getReservationRequirements(context, request.getId());
    String[] reqIds = new String[requirements.length];
    for (int i = 0; i < reqIds.length; i++) {
        reqIds[i] = requirements[i].getId();
    }
    endPoint.removeReservationRequirements(context, request.getId(), reqIds);

    requirements = endPoint.getReservationRequirements(context, request.getId());

    return requirements;
}

private void printReservationRequest(ReservationRequest request) {
    log("The request Id is: " + request.getId());
    log("The request Name is: " + request.getDisplayName());
    log("The request Description is: " + request.getDescription());
}

```

```

log("The request Owner is: " + request.getOwner());
log("The request Requester is: " + request.getRequester());
log("The request E-mail is: " + request.getRequesterEmail());
log("The request Justification is: " + request.getJustification());
log("The request Comments is: " + request.getComments());
log("The request Target Type is: " + request.getTargetType());
log("The request State is: " + request.getState());
log("The request Project Id is: " + request.getProjectId());
log("The request SAN Id is: " + request.getSanId());
log("The request Data Center Id is: " + request.getDataCenterId());
log("The request Ticket is: " + request.getTicket());
log("The request Error Count is: " + request.getErrorCount());
log("The request Warning Count is: " + request.getWarningCount());
log("The request Due Date is: " + request.getDueDate());

ReservationRequestTarget [] targets = request.getRequestTargets();

for (ReservationRequestTarget target: targets) {
    log("The target Type is: " + target.getType());

    Attribute [] attributes = target.getAttributes();
    log("The request Due Date is: " + attributes.length);

    Map<String, String> attributesMap = new HashMap<String, String>();

    for (Attribute attr : attributes) {
        attributesMap.put(attr.getName(), attr.getValue());
    }

    log("The target IP_ADDRESS is: " + attributesMap.get("IP_ADDRESS"));
    log("The target HOST_NAME is: " + attributesMap.get("HOST_NAME"));
    log("The target IS_NEW flag is: " + attributesMap.get("IS_NEW"));
}
}

private void printReservationRequirements (ReservationRequirement[] requirements) {

    for (ReservationRequirement req: requirements) {
        log("The requirement Type is: " + req.getType());
        log("The requirement Note is: " + req.getNote());
        Attribute [] testAttributes = req.getAttributes();

        Map<String, String> attributesMap = new HashMap<String, String>();
        for (Attribute attr : testAttributes) {
            attributesMap.put(attr.getName(), attr.getValue());
        }
        if (attributesMap.get("LUN_COUNT") != null) {
            log("The requirement LUN_COUNT is: " +
attributesMap.get("LUN_COUNT"));
            log("The requirement LUN_SIZE is: " +
attributesMap.get("LUN_SIZE"));
            log("The requirement TIER_ID is: " +
attributesMap.get("TIER_ID"));

```

```

        } else if (attributesMap.get("PORT_COUNT") != null) {
            log("The requirement PORT_COUNT is: " +
attributesMap.get("PORT_COUNT"));
        }
    }

    private void log(String msg) {
        System.out.println(msg);
    }

    public static void main(String[] args) throws Exception {
        CapacityRequestExample example = new CapacityRequestExample();

        try {
            ReservationRequest[] requests = example.createRequests();
            example.log("Create Requests is done....");
            example.printReservationRequest(requests[0]);

            ReservationRequirement[] requirements = example.createRequirements(requests[0]);
            example.log("Create Requirements is done....");
            example.printReservationRequirements(requirements);

            requests = example.updateRequests(requests[0]);
            example.log("Update Requests is done....");
            example.printReservationRequest(requests[0]);

            requirements = example.updateRequirements(requests[0]);
            example.log("Update Requirements is done....");
            example.printReservationRequirements(requirements);

            requirements = example.removeRequirements(requests[0]);
            example.log("Remove Requirements is done....");
            example.printReservationRequirements(requirements);
        } catch (APIBadRequestException ex) {
            System.out.println(ex.getMessage());
        }
    }
}

```

End Of Doc

Integration—Annotations, Applications, Business Units

Overview

The following samples provide a quick way to start to code Annotations, applications and Business Unit manipulations. Although the sample is in Java, the underlying communication layer is Web Services SOAP. As a result, you can use the API from and language that supports SOAP (e.g., Perl, .net, VB, C/C++ etc.).

Disclaimer

These samples are provided as is—NetApp is not responsible or liable and does not support this code. Care should be taken while using the API because information in the SANscreen database will be changed. ***Be advised that changes can not be undone.*** Make sure to back up the SANscreen database before running any API-related script.

Building

The sample code in this chapter requires the following:

A license to use the API. Verify that by surfing to the SANscreen server, clicking “license manager.” Verify that the license is On and that the API box is checked (see the License Manager on the next page).

Licensed Modules:					
Module	Quantity	API	Expires	Serial Number	
Application Insight: Array Performance	1000 TB	✓	Oct 8, 2008	123456	Add...
Application Insight: FC Switch	1000 Ports	✓	Oct 8, 2008	123456	Import...
Capacity Manager	1000 TB	✓	Oct 8, 2008	123456	Remove
Service Assurance	1000 Ports	✓	Oct 8, 2008	123456	
Service Insight: FC Port	1000 Ports	✓	Oct 8, 2008	123456	
Service Insight: NAS TB	1000 TB	✓	Oct 8, 2008	123456	
VM Insight	1000 TB	✓	Oct 8, 2008	123456	

1. Install JDK1.5.
2. Download the code samples from the SANscreen Portal → SANscreen Connect API → Examples.
3. Open the examples under temp.
4. When compiling and running the following jar files (java libraries) you should be in the classpath:
 - a. C:\TEMP\sanscreenapi-examples\java\lib\activation.jar
 - b. C:\TEMP\sanscreenapi-examples\java\lib\FastInfoset.jar
 - c. C:\TEMP\sanscreenapi-examples\java\lib\java-sdk.jar
 - d. C:\TEMP\sanscreenapi-examples\java\lib\jaxrpc-impl.jar
 - e. C:\TEMP\sanscreenapi-examples\java\lib\jaxrpc-spi.jar
 - f. C:\TEMP\sanscreenapi-examples\java\lib\jbossall-client.jar
 - g. C:\TEMP\sanscreenapi-examples\java\lib\jsr173_api.jar
 - h. C:\TEMP\sanscreenapi-examples\java\lib\mail.jar
 - i. C:\TEMP\sanscreenapi-examples\java\lib\saaj-impl.jar
 - j. C:\TEMP\sanscreenapi-examples\java\lib\xercesImpl.jar

Verification

In order to verify the effect and correctness of the sample code, the developer can have a SANscreen Client open and directed at the same SANscreen server as the API code/script. Any changes in the client should be reflected immediately in the API “get” results. Any API “set” actions should be presented immediately in the SANscreen Client UI.

Example 1 – Application and Business Unit

CAUTION: Before running the sample, make sure to back up the SANscreen database!

Overview

The code sample does the following things:

1. Finds a host by name.
2. Presents what application instances are configured in SANscreen as related to that host, along with which Business Units these applications belong to.
3. Defines a new application and a new Business Unit.
4. Associates the host with the new application.
5. Presents the new and revised relationship of applications and Business Units to the host.

Code Sample

File: ApplicationAndBusinessUnit.java

```
import com.onaro.sanscreen.wsapi.*;
import com.sun.xml.rpc.client.*;

import javax.net.ssl.*;
import java.rmi.RemoteException;

/**
 * This sample shows setting and getting Business Unit and Application
 */
public class ApplicationAndBusinessUnit {
    static String hostIP;
    static String hostID;
    static String hostName;
    static String applicationName;
    static String applicationId;

    public static void main(String[] args) throws Exception {
        SANscreenAPIEndPoint endpoint = createEndPoint();
        APISessionContext context = endpoint.openSession("1.0", 10000);

        String hostNameToSearch = "HT-ACC-1";

        // Get all info about this host (id and ip)
        getHostInfo(endpoint, context, hostNameToSearch); // sets variables with found data

        // Show applications that are configured on that host
        showApplicationsAndBusinessUnit(endpoint, context, hostID, hostName);

        // Define a new application
        String newApplicationName = "Application 1";
        String theBusinessUnit = "Marketing";
        String priority = "High";
```

```

        boolean ignoreSharing = false;
        createANewApplication(endpoint, context, newApplicationName, priority, theBusinessUnit,
ignoreSharing);

        // Retrieve the application information (in order to find the application id)
        getApplicationInfo(endpoint, context, newApplicationName); // sets variables with found
data

        // Add our host to the new application
        endpoint.addHostToApplication(context,hostIP, hostName, applicationId );

        // Show applications that are configured on that host (with the new added application)
        showApplicationsAndBusinessUnit(endpoint, context, hostID, hostName);

    }

    private static void createANewApplication(SANscreenAPIEndPoint endpoint, APISessionContext
context, String newApplicationName, String priority, String theBusinessUnit, boolean
ignoreSharing) throws APIException, RemoteException {
        endpoint.setApplication(context, newApplicationName, priority, theBusinessUnit,
ignoreSharing );
    }

    private static void getHostInfo(SANscreenAPIEndPoint endpoint, APISessionContext context,
String hostname) throws APIException, RemoteException {
        DeviceResponse deviceResponse = endpoint.findObjectsByName(context, null, hostname);
        hostID = deviceResponse.getDevices()[0].getId();
        hostIP = deviceResponse.getDevices()[0].getIp();
        hostName = hostname;
    }

    private static void getApplicationInfo(SANscreenAPIEndPoint endpoint, APISessionContext
context, String appname) throws APIException, RemoteException {
        ApplicationResponse application = endpoint.getApplication(context, null, appname);
        applicationName = appname;
        applicationId = application.getApplications()[0].getId();
    }

    private static void showApplicationsAndBusinessUnit(SANscreenAPIEndPoint endpoint,
APISessionContext context, String hostId, String hostname) throws APIException,
RemoteException {
        ApplicationResponse applicationsOfHost = endpoint.getApplicationsOfHost(context, null,
hostId);
        Application[] applications = applicationsOfHost.getApplications();
        System.out.println("Host "+hostname+" is configured with the following applications:");
        for (int appIdx = 0; appIdx < applications.length; appIdx++) {
            Application application = applications[appIdx];
            System.out.println("Application name: "+application.getName()+" Business
Unit: "+application.getBusinessUnit()+"");
        }
    }
}

```

```

protected static SANscreenAPIEndPoint createEndPoint() {
    TrustManager[] trustAllCerts = new TrustManager[]{
        new X509TrustManager() {
            public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                return null;
            }

            public void checkClientTrusted(
                java.security.cert.X509Certificate[] certs, String authType) {}

            public void checkServerTrusted(
                java.security.cert.X509Certificate[] certs, String authType) {}
        }
    };

    // Install the all-trusting trust manager
    try {
        SSLContext sc = SSLContext.getInstance("SSL");
        sc.init(null, trustAllCerts, new java.security.SecureRandom());
        HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
    } catch (Exception e) {}

    //this will prevent the server from throwing the following exception:
    //"Couldn't create SOAP message factory due to exception: javax.xml.soap.SOAPException:
    Failed to create MessageFactory: org.jboss.axis.soap.MessageFactoryImpl"
    System.setProperty("javax.xml.soap.MessageFactory",
        "com.sun.xml.messaging.saaj.soap.ver1_1.SOAPMessageFactory1_1Impl");

    SANscreenAPIEndPoint port = new
    SANscreenAPIService_Impl().getSANscreenAPIEndPointPort();
    ((StubBase) port)._setProperty(StubPropertyConstants.ENDPOINT_ADDRESS_PROPERTY,
        "http://localhost:80/sanscreenapi");
    ((StubBase) port)._setProperty(StubPropertyConstants.USERNAME_PROPERTY, "admin");
    ((StubBase) port)._setProperty(StubPropertyConstants.PASSWORD_PROPERTY, "NetApp");
    return port;
}
}

```

Example 2 – Annotation

CAUTION: Before running the sample, make sure to back up the SANscreen database and change the host name in the sample from “csNAS1” to a host name that exists in your environment. Change the newDataCenter to a value that exists in the SANscreen system (use the SANscreen Client for verification).

Overview

This sample defines two simple get and set methods for Annotations (getAnnotationForDevice, setAnnotationForDevice). For simplicity’s sake, these methods return string error values. The developer is encouraged to enhance this implementation to throw exceptions for the different error conditions (although the sample can be used as is). Note that these methods direct the developer not to use the Business Unit annotation for hosts.

The code sample does the following things:

Gets the Data Center Annotation of a certain host.

6. Gets the Note Annotation for that host.
7. Gets a value that is empty from that host—Annotation Sunset.
8. Gets an Annotation that does not exist and produces an error.
9. Gets a host name that does not exist and produces an error.
10. Sets a new Data Center for the host, and presents the old value.
11. Sets a new note with a date stamp for the host, and presents the old value.
12. Attempts to set a Business Unit Annotation that leads to an error.

After running the sample code once the result will be similar to:

1 DataCenter for csNAS1: 'Boston'

2 Note for csNAS1: 'Original note'

3 (should produce no value) Sunset for csNAS1: 'NoValue'

*4 (should produce an error - the annotation is not found) Notex for csNAS1:
'Error:AnnotationNotFound'*

5 (should produce an error - the host is not found) Note for csNAS1_foo: 'Error:DeviceNotFound'

6 set DataCenter for csNAS1. Old value: 'Boston'

7 set Note for csNAS1. Old value: 'Original note'

8 The new note for csNAS1. Old value: 'This is a new note! Assigned on Fri Dec 28 15:04:21 EST 2007'

9 (should produce an error - incorrect association) set BU for csNAS1: 'Error:IncorrectAssociation'

Done

After running the sample the second time, the result will be similar to:

1 *DataCenter* for csNAS1: 'NY'

2 Note for csNAS1: 'This is a new note! Assigned on Fri Dec 28 15:04:21 EST 2007'

3 (should produce no value) *Sunset* for csNAS1: 'NoValue'

4 (should produce an error - the annotation is not found) *Notex* for csNAS1: 'Error:AnnotationNotFound'

5 (should produce an error - the host is not found) Note for csNAS1_foo: 'Error:DeviceNotFound'

6 set *DataCenter* for csNAS1. Old value: 'NY'

7 set Note for csNAS1. Old value: 'This is a new note! Assigned on Fri Dec 28 15:04:21 EST 2007'

8 The new note for csNAS1. Old value: 'This is a new note! Assigned on Fri Dec 28 15:05:36 EST 2007'

9 (should produce an error - incorrect association) set BU for csNAS1: 'Error:IncorrectAssociation'

Done

Note the differences in results for steps 1, 2, 6, 7 and 8 (underlined). The second run reflects changes that were applied in the first run.

Code Sample

File: AnnotationGetSet.java

```
import com.NetApp.sansscreen.wsapi.*;
import com.sun.xml.rpc.client.*;

import javax.net.ssl.*;
import java.rmi.RemoteException;
import java.util.Date;

/**
 * This sample shows setting and getting annotations
 */
public class AnnotationGetSet {
    public static String NO_VALUE="NoValue";
    public static String DEVICE_NOT_FOUND="Error:DeviceNotFound";
    public static String ANNOTATION_NOT_FOUND="Error:AnnotationNotFound";
    public static String ENUMERATION_NOT_FOUND="Error:EnumerationNotFound";
    public static String INCORRECT_ASSOCIATION="Error:IncorrectAssociation";
    public static String SET_FAILED="Error:SetFailed";
    public static String errors[] = {NO_VALUE, DEVICE_NOT_FOUND,
    ANNOTATION_NOT_FOUND};

    public static void main(String args[]) throws Exception {
        SANSscreenAPIEndPoint endpoint = createEndPoint();
        APISessionContext context = endpoint.openSession("1.0", 10000);
```

```

String myHostName = "csNAS1";
String newDataCenter = "NY";

String nonExistantHost = myHostName+"_foo";

int step = 1;

//----- GET -----
System.out.println((step++) + " DataCenter for "+myHostName+": " +
getAnnotationForDevice(context, endpoint, myHostName, "Data_Center")+""");
System.out.println((step++) + " Note for "+myHostName+": " +
getAnnotationForDevice(context, endpoint, myHostName, "Note")+""");

System.out.println((step++) + " (should produce no value) Sunset for "+myHostName+":
" + getAnnotationForDevice(context, endpoint, myHostName, "Sunset")+""");
System.out.println((step++) + " (should produce an error - the annotation is not found)
Notex for "+myHostName+": " + getAnnotationForDevice(context, endpoint, myHostName,
"Notex")+""");
System.out.println((step++) + " (should produce an error - the host is not found) Note for
"+nonExistantHost+": " + getAnnotationForDevice(context, endpoint, nonExistantHost,
"Note")+""");

//----- SET -----

System.out.println((step++) + " set DataCenter for "+myHostName+". Old value: " +
setAnnotationForDevice(context, endpoint, myHostName, "Data_Center", newDataCenter)+""");

System.out.println((step++) + " set Note for "+myHostName+". Old value: " +
setAnnotationForDevice(context, endpoint, myHostName, "Note", "This is a new note! Assigned
on "+(new Date()).toString()+""");
System.out.println((step++) + " The new note for "+myHostName+". Old value: " +
getAnnotationForDevice(context, endpoint, myHostName, "Note")+""");

System.out.println((step++) + " (should produce an error - incorrect association) set BU for
"+myHostName+": " + setAnnotationForDevice(context, endpoint, "csNAS2", "Business_Unit",
"Dev")+""");

//----- DONE -----

endpoint.closeSession(context);
System.out.println("Done");
}

/**
 * Gets an annotation value by device name
 * @param context Session context
 * @param endpoint end point for function activation
 * @param deviceName The name of device to find
 * @param annotationName The string name of annotation (by definition, not the display
name)

```

```

    * @return The string value of the annotated field. If there is no value for that annotation,
    "NoValue" is returned. If the device is not found, the value "DeviceNotFound" is returned. If the
    annotation is not found, "AnnotationNotFound" is returned.
    * @throws APIException
    * @throws RemoteException
    */
    private static String getAnnotationForDevice(APISessionContext context,
    SANscreenAPIEndPoint endpoint, String deviceName, String annotationName) throws
    APIException, RemoteException {
        String value=NO_VALUE;

        DeviceResponse deviceResponse = endpoint.findObjectsByName(context, null,
        deviceName);

        if( deviceResponse.getDevices().length < 1 ) return DEVICE_NOT_FOUND;
        if( deviceResponse.getDevices()[0].getObjectType().equals("Host") &&
        annotationName.equals("Business_Unit") ) return INCORRECT_ASSOCIATION;

        AnnotationType annotationType =
        endpoint.getAnnotationTypeByName(context,annotationName);
        if( annotationType == null ) return ANNOTATION_NOT_FOUND;

        ObjectReference deviceObjectReference = new
        ObjectReference(deviceResponse.getDevices()[0].getId(),deviceResponse.getDevices()[0].getObj
        ectType());
        AnnotationValue[] annotationValues = new
        com.NetApp.sanscreen.wsapi.AnnotationValue[0];
        try {
            annotationValues = endpoint.getAnnotationValues(
                context,
                new ObjectReference[] {deviceObjectReference},
                new String[] {annotationType.getId()});
        } catch (APIException e) {
            return NO_VALUE;
        } catch (RemoteException e) {
            return NO_VALUE;
        }
        if( annotationValues.length < 1 ) return NO_VALUE;

        if( annotationType.getDataType().equals("FLEXIBLE_ENUM") ) {
            AnnotationEnumValue[] enumerationValues = annotationType.getEnumerationValues();
            for (int i = 0; i < enumerationValues.length; i++) {
                AnnotationEnumValue enumerationValue = enumerationValues[i];
                if( enumerationValue.getId().equals(annotationValues[0].getEnumId()) ) {
                    value = enumerationValue.getName();
                    break;
                }
            }
        } else {
            value = annotationValues[0].getValue();
        }
    }

```

```

        return value;
    }

    /**
     * Sets an annotation value for a device
     * @param context Session context
     * @param endpoint end point for function activation
     * @param deviceName The name of device to find
     * @param annotationName The string name of annotation (by definition, not the display
    name)
     * @return The old string value of the annotated field if the operation was successfull. If there
    is no value for that annotation, "NoValue" is returned. If the device is not found, the value
    "DeviceNotFound" is returned. If the annotation is not found, "AnnotationNotFound" is returned.
     * @throws APIException
     * @throws RemoteException
     */
    private static String setAnnotationForDevice(APISessionContext context,
    SANscreenAPIEndPoint endpoint, String deviceName, String annotationName, String newValue )
    throws APIException, RemoteException {
        String oldValue=getAnnotationForDevice(context,endpoint, deviceName, annotationName);
        if( oldValue.equals(DEVICE_NOT_FOUND)) return oldValue;
        if( oldValue.equals(ANNOTATION_NOT_FOUND)) return oldValue;
        if( oldValue.equals(INCORRECT_ASSOCIATION)) return oldValue;

        DeviceResponse deviceResponse = endpoint.findObjectsByName(context, null,
        deviceName);

        AnnotationType annotationType =
        endpoint.getAnnotationTypeByName(context,annotationName);

        ObjectReference deviceObjectReference = new
        ObjectReference(deviceResponse.getDevices()[0].getId(),deviceResponse.getDevices()[0].getObj
        ectType());

        AnnotationValue annotationValue = null;

        if( annotationType.getDataType().equals("FLEXIBLE_ENUM") ) {
            AnnotationEnumValue[] enumerationValues = annotationType.getEnumerationValues();
            for (int i = 0; i < enumerationValues.length && annotationValue == null; i++) {
                AnnotationEnumValue enumerationValue = enumerationValues[i];
                if( enumerationValue.getName().equals(newValue) ) {
                    annotationValue = new AnnotationValue(null,enumerationValue.getId(),
                    deviceObjectReference, System.currentTimeMillis(), annotationType.getId(),
                    enumerationValue.getName());
                }
            }
            if( annotationValue == null ) return ENUMERATION_NOT_FOUND;
        } else {
            annotationValue = new AnnotationValue(null,"-1", deviceObjectReference,
            System.currentTimeMillis(), annotationType.getId(), newValue);
        }
    }

```

```

        boolean res = endpoint.setAnnotationValues(context,new AnnotationValue[]{
annotationValue } );
        if(!res) return SET_FAILED;

        return oldValue;
    }

protected static SANscreenAPIEndPoint createEndPoint() {
    TrustManager[] trustAllCerts = new TrustManager[]{
        new X509TrustManager() {
            public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                return null;
            }

            public void checkClientTrusted(
                java.security.cert.X509Certificate[] certs, String authType) {
            }

            public void checkServerTrusted(
                java.security.cert.X509Certificate[] certs, String authType) {
            }
        }
    };

    // Install the all-trusting trust manager
    try {
        SSLContext sc = SSLContext.getInstance("SSL");
        sc.init(null, trustAllCerts, new java.security.SecureRandom());
        HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
    } catch (Exception e) {
    }

    //this will prevent the server from throwing the following exception:
    //"Couldn't create SOAP message factory due to exception: javax.xml.soap.SOAPException:
Failed to create MessageFactory: org.jboss.axis.soap.MessageFactoryImpl"
    System.setProperty("javax.xml.soap.MessageFactory",
        "com.sun.xml.messaging.saaj.soap.ver1_1.SOAPMessageFactory1_1Impl");

    SANscreenAPIEndPoint port = new
SANscreenAPIService_Impl().getSANscreenAPIEndPointPort();
    ((StubBase) port)._setProperty(StubPropertyConstants.ENDPOINT_ADDRESS_PROPERTY,
"http://localhost:80/sanscreenapi");
    ((StubBase) port)._setProperty(StubPropertyConstants.USERNAME_PROPERTY, "admin");
    ((StubBase) port)._setProperty(StubPropertyConstants.PASSWORD_PROPERTY, "NetApp");
    return port;
}
}

```

End Of Doc

Index

A

- about SANscreen
 - quick start instructions 10
- Add LUNs 33
- Add Ports 32
- administrative portal 10
- Allocation Audit Report 53
- Annotation Rules
 - configure 19
 - Storage Tier Quality 20
 - vendor 24
 - vendors 21
 - Volume Tier Quality 20
- Annotation tier values
 - automatically assigning 20
- Annotation Types 21, 23

C

- Cancelling a Request 42
- Chargeback Report 51
- configure annotation rules 19
- Create a New Request 27

D

- Deallocation of a Reservation 42
- Disk Size 26
- Disk Speed 25
- Disk Type 25
- documentation set 9

F

- Filter by Model 24
- Filtering Volumes 24
- Filtering volumes
 - Disk Size 26

- Disk Speed 25
- Disk Type 25
- redundancy 25
- Volume Type 24

G

- General Reservations 37

O

- overview of
 - quick start instructions 10

P

- Placing a Request 27
- Port Requirements 39

Q

- quick start instructions 10

R

- Redundancy 25
- Remove a Request 43
- reports 47
- Request Portal 35
- Reserving resources 36
- Rules 19, 20, 22

S

- SANscreen Settings 19
- Satisfying a Requirement 37
- service quality 19
- Setting Up Requirements 32

Index

State Transitions of a Request 18
Storage Tier Quality 20, 22
 classification 20
Storage Utilization Report 48
Storage Utilization report 47

T

Target Tab 29
tier values
 setting 21
Tiering
 level of service 20

U

using Rules 19

V

vendor See Annotation Rules
View Pending Tasks 45
Volume Requirements 38
Volume Rule 22
Volume Tier Quality 20, 22, 23
Volume Type 24